Alma Mater Studiorum - Università di Bologna

DIPARTIMENTO DI FISICA

Dottorato di Ricerca in Fisica - Ciclo XXIII

A Novel Map-Matching Procedure for Low-Sampling GPS Data with Applications to Traffic Flow Analysis

Luca Giovannini

Ph.D. Thesis

Coordinatore del Dottorato:
Chiar.mo Prof.
Fabio Ortolani

Relatore:
Chiar.mo Prof.
Sandro Rambaldi

Bologna, 2011

Settore Scientifico Disciplinare: FIS/07

Abstract

An extensive sample (2%) of private vehicles in Italy are equipped with a GPS device that periodically measures their position and dynamical state for insurance purposes. Having access to this type of data allows to develop theoretical and practical applications of great interest: the real-time reconstruction of traffic state in a certain region, the development of accurate models of vehicle dynamics, the study of the cognitive dynamics of drivers. In order for these applications to be possible, we first need to develop the ability to reconstruct the paths taken by vehicles on the road network from the raw GPS data. In fact, these data are affected by positioning errors and they are often very distanced from each other ($\sim$ 2 Km). For these reasons, the task of path identification is not straightforward.

This thesis describes the approach we followed to reliably identify vehicle paths from this kind of low-sampling data. The problem of matching data with roads is solved with a bayesian approach of maximum likelihood. While the identification of the path taken between two consecutive GPS measures is performed with a specifically developed optimal routing algorithm, based on A* algorithm. The procedure was applied on an off-line urban data sample and proved to be robust and accurate. Future developments will extend the procedure to real-time execution and nation-wide coverage.

Sommario

Un ampio campione (2%) di veicoli italiani dispone di un dispositivo GPS che ne misura periodicamente la posizione e lo stato dinamico per fini assicurativi. L'accesso a questo tipo di dati permette lo sviluppo di applicazioni di grande interesse sia teorico che pratico: la ricostruzione in tempo reale dello stato del traffico in una determinata area, il supporto allo sviluppo di modelli accurati di dinamica veicolare, lo studio della dinamica cognitiva degli automobilisti. Propedeutico a tutte queste ricerche é peró la ricostruzione dei tragitti percorsi dai veicoli sulla rete stradale a partire dai dati GPS. Questi dati sono affetti da errori di posizionamento e sono spesso molto spaziati tra di loro (circa 2km), caratteristiche che rendono non ovvia la ricostruzione dell'effettivo percorso sulla rete.

Questa tesi descrive l'approccio seguito per ricostruire in maniera affidabile i percorsi su rete a partire da queste condizioni di bassa campionatura. Il problema del posizionamento dei dati sulle strade viene affrontato secondo un approccio bayesiano di massima verosimiglianza. Mentre per la determinazione del percorso seguito tra dati successivi é stato sviluppato uno specifico algoritmo di optimal routing basato sull'algoritmo A*. La procedura é stata applicata ad un campione storico di dati su scala urbana e si é dimostrata robusta ed accurata. Gli sviluppi futuri prevedono l'applicazione in tempo reale su scala nazionale.

Contents

Introduction **1**

Motivation of the Research 1

Outline of this Thesis 4

1 The Map-Matching Problem **5**

1.1 Map-Matching Problem Statement 5

1.2 Review of Map-Matching Approaches 7

1.3 Overview of the Procedure 10

2 Input Data Specifications **11**

2.1 Vehicle GPS data 11

2.2 Road network data 15

2.3 Florence test dataset 18

3 Trajectory Aggregation **21**

3.1 Trajectory Aggregation Procedure 22

3.1.1 Parameters and definitions 22

3.1.2 Reordering of engine state information 24

3.1.3 Removal of known data consistency anomalies 25

3.1.4 Definition of the spatio-temporal boundaries 27

3.1.5 Identification of data reliability intervals 28

3.1.6 Consistency specifications of the aggregated data 28

3.2 Results and Applications 29

3.2.1 Trajectory aggregation for Florence dataset 29

3.2.2 Statistical laws in urban mobility 33

4 Data Matching by Affinity 39

4.1 Data Matching Procedure 40

4.1.1 Categories of data accuracy 40

4.1.2 Definition of positioning probability 41

4.1.3 Definition of prior road probability 44

4.1.4 Matching data to roads by affinity 45

4.2 Results and Applications 47

4.2.1 Data matching for Florence dataset 47

4.2.2 Fundamental diagram of traffic flow 50

5 Global Path Matching 55

5.1 Path Matching Procedure 56

5.1.1 A* algorithm for path finding 56

5.1.2 Path finding between records 58

5.1.3 Global best path matching 61

5.1.4 Path consistency check 62

5.1.5 Road transits computation 64

5.2 Results and Applications 65

5.2.1 Path matching for Florence dataset 65

5.2.2 Towards real-time traffic monitoring 67

Conclusions 71

References 75

Introduction

Motivation of the Research

In recent years the use of positioning devices has become ever more common in everyday life. The case of GPS (Global Positioning System) instruments is exemplar in this respect. GPS devices started out as highly professional tools, used only in specific application fields (surveying, topography, military) and ended up as everyday gadgets, present in many of the current multi-functional mobile phones that we all use.

Looking specifically at the automotive market, GPS positioning devices are now very common features on board of vehicles. The most common applications range from showing the current vehicle position on a street map to planning the best route to a given destination and guiding the driver towards it.

But the applications of on-board GPS positioning are not limited to this. Even if less known, many vehicles actually mount GPS receivers on board also for insurance-related applications. For example, an insurance company can be interested in keeping a record of the vehicle speed and movements, so that, in case of an accident, its dynamics can be reconstructed more accurately, responsibilities ascertained and claims settled with less risks of fraud. Another useful insurance-related application of an on-board GPS device is the localization of the current position of a stolen vehicle.

Applications also exist in the field of traffic monitoring and management. In many cities around the world the position of public transportation vehicles is monitored by a central authority and this information is used to improve the quality of the service to the public. For example, users waiting at the bus stop are updated on the expected time of arrival of the next buses. In a similar way, allowing traffic authorities to monitor positions and movements of private vehicles can simplify the management of toll-road payment as well as the monitoring of traffic in and out of limited-access areas of cities. A very interesting application of vehicle positioning is also real-time monitoring of global traffic conditions and the short term prediction of its evolution, the so-called traffic now-casting.

Recently, the Physics of the City Laboratory of the University of Bologna has been given access to a huge database of vehicle GPS data by Octo Telematics SpA, an Italian company specializing in the provision of telematics services and systems for the insurance and automotive market. Octo Telematics SpA handles the data collected by GPS devices mounted on board of private vehicles in Italy, mainly for insurance-related applications. All the data collected by these GPS devices are periodically sent back to the storing facility of the company and put into the database in order to be elaborated as required. As for January 2011, the number of vehicles equipped with such a device amounts to about 1.2 millions, which represents roughly 2% of the total private vehicles registered in Italy [1].

The amount of empirical data available (see Fig. 1) is incomparably more extensive in size and more homogeneous in distribution than what could be collected in years of specific field measures or by conventional interviews to drivers. Obviously, the data available to us do not contain any personal information about the identity of the drivers, so to guarantee the respect of their privacy rights.

From a scientific standpoint, this database represents a unique opportunity to study the properties of vehicular traffic as a complex dynamical system, from the microscopic to the macroscopic scale. In fact, the data can be used to study the global statistical properties of the system, as well as to study the impact of the human cognitive component on traffic [2], [3].

Moreover, the availability of such a wealth of vehicle data is also inter-

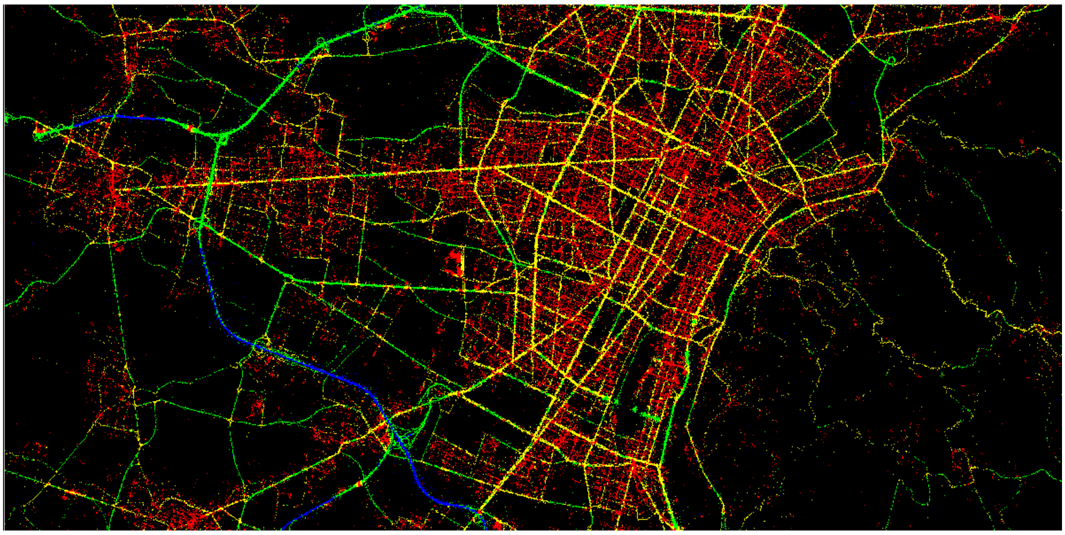

Figure 1: GPS data plot in the metropolitan area of Turin, Italy, September 2007. Approximately 7 million data points are plotted in this image with a color code describing speed.

esting from the perspective of application development. The extent and the distributed nature of the data make them exploitable as probe-data to determine the traffic conditions on a given area of the road network. Further, if this knowledge of traffic conditions is integrated with a short-term predictive model for vehicle dynamics, then a real-time traffic now-casting infrastructure could be put in place. This is undoubtedly not a simple application to develop but by no means is the only one possible. For example, accurate models for microscopic traffic dynamics could be developed thanks to a continuous testing against real-word data. Also, an update service for road network maps could be put in place using vehicle data to identify newly built roads.

In order to actually move forward with many of the research projects that were just described, it is first of all necessary to develop the ability to reconstruct the path followed by each vehicle on the road network from the raw GPS data that we have access to. Due to GPS positioning errors and low spatial sampling of data, this task is not straightforward at all. These GPS measurements are in fact affected by errors and this uncertainty often prevents an unambiguous direct matching to the underlying road network map. Moreover, due to the need of containing the costs for data delivery and storage, the majority of these GPS measurements are saved to memory with a low spatial sampling of roughly 2 km, which means that we do not have any direct information on most part of the vehicle path. Thus, it is necessary to develop a procedure to reconstruct vehicle paths that properly takes into account both those sources of uncertainty. This situation falls into the broader category of problems commonly referred to as map-matching.

This thesis describes the details of this specific map-matching problem and presents the procedure that was developed to solve it, along with validation results and examples of applications to the study of traffic as a dynamical system.

Outline of this Thesis

The rest of this thesis is organized in five chapters and a conclusion.

Chapter 1 defines the terms of the low-sampling map-matching problem, sets it in the context of the current variety of approaches and presents the approach to the solution.

Chapter 2 describes the details of the input data.

Chapter 3, 4 and 5 describe the three main phases of the map-matching algorithm. Chapter 3 presents the trajectory aggregation phase, necessary to organize and aggregate the data for an efficient elaboration during the following steps. Chapter 4 describes how GPS data are matched to road segments. Chapter 5 discusses the details of the path finding algorithm used to reconstruct the vehicle paths on map. Each chapter ends with a section dedicated to validation results and applications.

The conclusion presents a summary of the results and describes the plans for future developments.

Chapter 1

The Map-Matching Problem

1.1 Map-Matching Problem Statement

Before giving a proper statement of the map-matching problem discussed in this thesis, it is necessary to give some preliminary definitions:

GPS Datum: A GPS datum is the ensemble of all direct and indirect measurements on the state of a vehicle taken by the on-board GPS device at a certain instant in time and saved into memory. A detailed description of the components of a GPS datum is given in Tab. 2.1.

GPS Trajectory: A GPS trajectory is an ordered sequence of GPS data belonging to the same vehicle. Each GPS trajectory comprises all and only the GPS data available for a single vehicle journey (from engine start to engine stop).

Road Network: A road network is a directed graph representing the shape and properties of the road system of a certain geographical area. The vertexes describe road intersections and dead ends, while the edges describe road shapes and attributes.

Road Element: A road element is a directed edge of a road network graph. A detailed description of the components of a road element is given in Tab. 2.2.

Road Transit: A road transit is a quantity that describes the transit of a vehicle on a road element or a portion of it. A detailed description of the components of a road transit is provided in Tab. 1.1.

Vehicle ID	Vehicle identification number
Road ID	Road identification number
Transit Duration	Duration of the transit (total or partial)
Timestamp	Time at the end of the transit (total or partial)
Average Speed	Average speed of the vehicle during the transit (total or partial)

Table 1.1: The components of a Road Transit

Reconstructed Path: A reconstructed path is an ordered sequence of road transits along connected road elements.

The map-matching problem can now be defined as follows: *Given a GPS trajectory T and a road network N, find the path P that matches T with its most realistic reconstruction on N.*

Specifically, the problem we confront in this thesis is a low-sampling map-matching problem as the GPS data available to us are mostly sampled at a low-frequency rate of one measure every 2 Km. Fig. 1.1 shows a typical GPS trajectory from our dataset, along with its reconstructed path. It is apparent how the low spacial sampling of the GPS data increases the difficulty of the map-matching problem.

The procedure developed in this work to solve the map-matching problem has been designed as an off-line application and has therefore been tested on datasets bounded to a given geographical area and a given interval of time. The test dataset chosen here to present the results of our map-matching approach covers the city Florence during the month of March 2008.

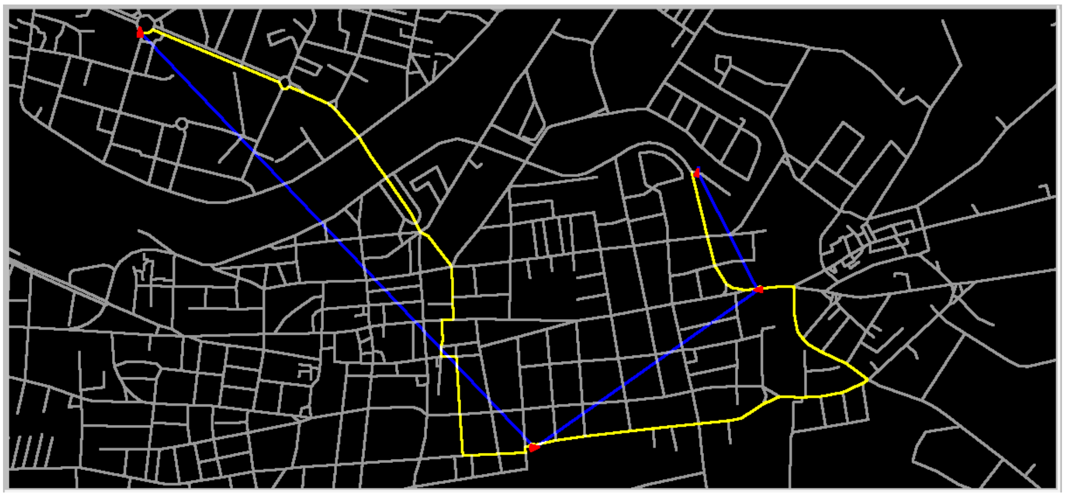

Figure 1.1: A typical map-matching case. The red triangles identify single GPS data records. The blue line, connecting the GPS data into a sequence, represents a GPS trajectory. The yellow line describes a possible reconstruction of the path followed by the vehicle.

1.2 Review of Map-Matching Approaches

Generally speaking, a map-matching algorithm is an automatized procedure that combines measures from one or more positioning devices with data from a road network map to provide an enhanced positioning output. This task is usually not straightforward because of the combined effect of measurement errors in positioning data and accuracy errors in road network data. The predominant positioning technology employed for map-matching applications is by far GPS, because of its global coverage and the relatively low cost of the measuring devices. However, map-matching solutions designed for the automotive market may rely also on Deduced Reckoning (DR) sensors as secondary sources of positioning data to bridge any possible gap in GPS coverage [4]. A typical DR sensor consist of an odometer and a gyroscope that keep track of vehicle speed and steering.

The first research works on map-matching date back to the late eighties of the last century, but the attention to this problem really begun to rise in the nineties, in parallel with the diffusion of Personal Navigation Assistants

(PNAs). In fact, PNAs were soon equipped with GPS sensors and applications were developed to show to the user his current position on a map and to guide him or her to the chosen destination. Obviously, in order for this kind of application to work, reliable map-matching algorithms needed to be developed [5].

Up to the present day, the majority of works related to map-matching is still focused on solving, with ever more accuracy and reliability, the same problem of keeping track in real-time of the correct position of the user on a map. Thus, the positioning data used for this kind of task has usually a high sampling frequency ($\sim$ 1 sample per second) and great attention is put in the trade-off between accuracy and computational cost. Many different techniques have been developed to solve this kind of map-matching problem, ranging from simple geometrical considerations to more advanced inference methods, but they are commonly categorized for simplicity into four groups: geometric, topological, probabilistic and advanced [6].

Geometric map-matching techniques match position data to road segments relying on simple geometrical constraints, for example minimizing the distance between vehicle and road and the difference between vehicle heading and road direction [5]. Topological map-matching techniques not only rely on geometrical constraints to find the best matches, but they also take into account the topological constraints set by the road network structure on the matching path, for example by making sure that consecutive data are associated to adjacent road segments [7], [8], [9]. Probabilistic map-matching techniques take into account the errors associated with positioning measures and road network data to get a more accurate determination of the suitable road segment candidates for matching [10], [11]. Advanced map-matching techniques are a collection of very different approaches that rely on advanced inference techniques, such as Bayesian inference [13], [14], Kalman filters [4], Fuzzy logic [15], Dempster-Shafer's belief theory [16].

More recently, the attempts to develop efficient means for intelligent traffic management have brought forward a new set of problems and solutions related to map-matching. Among them, the problem of traffic monitoring via Floating Car Data (FCD) is one of the most studied [19]. The idea behind FCD is that it is not necessary to equip every single vehicle on the road with sensors for position and speed in order to be informed on real-time traffic conditions. Realizing the latter would require a huge investment in sensors and a huge cost for communications between every vehicle and the central monitoring unit. Instead, within the FCD framework, traffic conditions can be determined by a small, distributed sub-sample of the vehicles moving on the road network. Those probe-vehicles are constrained to flow according to the overall traffic fluxes they are part of and are thus representative of them.

For this kind of application, data requirements and elaboration techniques are in many respect different from the scenario of real-time user positioning on map. The precise positioning of each datum along the matched road segment is no longer very relevant, but the correct identification of the path followed by the vehicle along the road network is now much more important. Moreover, applications do not necessarily need real-time data elaboration. However, it is relevant to note that the data collected by the probe vehicles, due to the need of containing the costs for data delivery and storage, have usually a low sampling frequency ($\sim$ 1 sample every 2 minutes or more), bringing forth uncertainties in path reconstruction and traffic conditions update [17].

It is then apparent that great attention has to be put in the routing task, which is the problem of identifying the correct path taken by the vehicle on the road network between two consecutive positioning measurements. The most common approach is obviously to look for the shortest path [17]. More elaborated approaches look for the path which is closer in length to the distance between the two positioning measurements [19] or for the path that has the average travel time which is closer to the actual time elapsed between the two measurements [18].

The field of traffic study and monitoring is where the contribution of this thesis falls. As mentioned above, we have access to a very rich database of GPS tracking data for vehicles distributed throughout Italy and these data are mostly sampled at a low-frequency rate of one measure every 2 Km. In order to reconstruct the path taken by vehicles on the road network from such positioning data, we developed a procedure for low-sampling map-matching.

1.3 Overview of the Procedure

The aim of a map-matching algorithm is to reconstruct from a GPS trajectory the path driven by a vehicle on the road network. The main difficulties in this task derive from the errors of GPS positioning measurements and from the uncertainty introduced by the sampling of the data [12].

The map-matching procedure developed in this thesis handles the positioning uncertainties adopting a bayesian approach of maximum likelihood. The data are projected on the road segments that have the higher probabilities of having generated them.

The identification of the possible paths taken by vehicles is performed by a modified version of the A* algorithm for optimal routing [20]. The routing algorithm matches the vehicle path to the one requiring the shortest travel-time, while taking into account the constraints determined by space and time intervals between measures.

The overall procedure can be divided in different phases. Before the actual map-matching of GPS trajectories takes place, some initialization operation are performed to speed up the following elaborations: road network data for the study area are loaded in memory and a *road proximity map* is created. This proximity map allows for a fast identification of the road arcs that are close to every given spacial position inside the study area.

Once the initialization step is completed, the map-matching can start. First of all, the data from each vehicle goes through a trajectory aggregation stage, that serves the purpose of removing useless data and aggregating useful GPS data into trajectories. Then, GPS trajectories are processed in sequence through the two last steps of the procedure: the projection of GPS data onto the surrounding road elements and the identification of the optimal path between projected data.

The chapters that follow will describe in details the three phases of trajectory aggregation, data matching and global path matching. The results of the application of each of these processing steps to the Florence test dataset will also be discussed.

Chapter 2

Input Data Specifications

2.1 Vehicle GPS data

The data on vehicle mobility that we use in this work derive from measurements taken by GPS devices installed on a sample of roughly 2% of the private vehicles registered in Italy. The data gathered by these devices, installed mainly for insurance-related applications, are collected via GSM/GPRS network by Octo Telematics SpA [1], who granted us access to part of its database for research purposes.

The GPS device identifies the location of the vehicle on the Earth surface and associates to this position reading an accurate measurement of time. From these fundamental measurements of position and time, the device can derive estimates about vehicle speed, heading and distance from the previous measurement.

During each reading, the device records a few other quantities, such as the state of the engine and the quality of the signal received from GPS satellites. From the GPS signal quality depends the accuracy of the measurements of position, time and all the other derived quantities. Tab. 2.1 describes the specific format of a GPS datum.

While the vehicle is moving, the GPS device takes its measures with a typical frequency of about one measure per second. However, only a subset

Vehicle ID	id	Vehicle identification number
Date	date	Date of the measurement (dd/mm/yy)
Time	time	Time of the measurement (hh:mm:ss)
Latitude	lat	Vehicle latitude in WGS84 reference system (10e-6 arc degrees)
Longitude	lon	Vehicle longitude in WGS84 reference system (10e-6 arc degrees)
Speed	vel	Vehicle speed (Km/h)
Heading	ang	Direction of movement (degrees, clockwise with respect to North)
GPS signal quality	gps_q	1 = no signal, 2 = poor signal, 3 = good signal
Engine state	eng_s	0 = engine on, 1 = cruise, 2 = engine off
Distance from previous datum	ds	Distance traveled from the last saved GPS datum (meters)

Table 2.1: The components of a GPS Datum

of these readings are stored in the built-in memory. When a predefined quota of memory is filled, the device proceeds to send the saved data to a central storing facility of Octo Telematics via the GSM/GPRS network.

Different settings for data memorization are possible for the GPS devices. The two most common settings are:

Standard Mode: The device memorizes a new datum as soon as the distance traveled by the vehicle from the last saved datum reaches 2 Km and the GPS signal quality is good. In this mode, the device sends the data every time 50 records are present in memory.

Traffic-Info Mode: The device memorizes a new datum every 30 seconds and sends the data as soon as 24 records are present in memory. This mode is typically activated when the vehicle travels on a highway and is thus less commonly used than the Standard Mode.

Irrespective of the current memorization mode, the device always takes a reading when the vehicle's engine is started and saves it to memory. Similarly, a reading is always taken and saved to memory when the engine is turned off.

The uncertainty on the measured quantities is strictly related to the quality of GPS signal reception during the measurements. The error on the measure of time due to GPS signal quality is of the order of 1 second and it is thus almost irrelevant for our map-matching purposes. Instead, the error on the positioning measurement is much more sensitive to the quality of the GPS signal. When signal reception is good, the typical positioning uncertainty is usually of the order of 10 meters, but in adverse circumstances the errors can increase up to 30 meters or more. Moreover, if we refer for positioning to a digital road network map, as in our case, then also digitization errors and centerline inaccuracies in the map have an impact on the overall positioning error [6]. Fig. 2.1 is a GPS data plot in the area of a busy highway with access ramps. The highway segment clearly shows how distributed the GPS data can be across the road they presumably belong to, while, looking at the ramps, it is also apparent how systematic discrepancies in positioning might occur between data and digitized roads. Obviously, the uncertainties on the readings for speed, heading and distance from the previous measurement depend directly from the uncertainties on the measurements of position and time from which

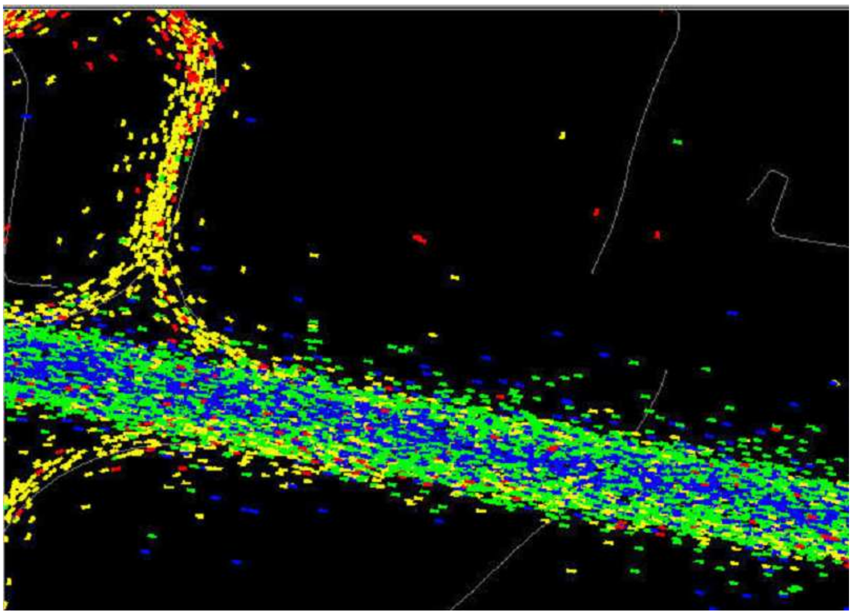

Figure 2.1: A close up view of a GPS data plot in the area of a busy highway with access ramps. GPS data are represented by rectangles with a color code describing speed: red has $vel \leq 30$ km/h, yellow has 30 km/h $< vel \leq 50$ km/h, green has 50 km/h $< vel \leq 90$ km/h and blue has 90 km/h $< vel$.

they derive. It is relevant to note here that speed and heading are rounded up to the closest even integer value.

Measurement uncertainty due to poor or bad GPS signal quality is a major cause of difficulties when attempting to reconstruct vehicle movements on the road network from a low-sampled sequence of positioning data. In particular, a case where this problem is very often present is the starting of the vehicle. In fact, even if the GPS device attempts to remain in contact with the satellite constellation even when the vehicle is not in use, the reading taken when the engine is started is very often taken in the condition of no signal lock. However, in these cases it is often possible to recover a reliable positioning just by referring to the previous saved datum, taken when the engine was turned off.

To estimate the nature of positioning uncertainties in our data we isolated a sub-sample of roughly 1.7 million records, chosen among those taken under good signal conditions ($gps\_q = 3$) and unambiguously matchable to long straight road segments. The distribution in Fig. 2.2 describes the minimum distance between datum and road segment and has a standard deviation of about 8.5 meters. The distribution in Fig. 2.3, restricting only to data records with $vel > 10km/h$, describes the difference between vehicle heading (ang) and segment direction and has a standard deviation of about 1.2 degrees. The two distributions have a similar shape, even if the latter is visibly more noisy. This noise is mainly due to the round-off on ang, but other factors may impact, such as the derived nature of the reading for vehicle heading, the dependence of heading accuracy on vehicle speed, the greater sensibility of angles to inaccurate road digitization.

2.2 Road network data

For this work we used a commercial off-the-shelf road network database. Road information is stored in the database in Esri Shapefile format, one of the most common standards for this type of geo-referenced vector data. In this format, road shape is described by a piecewise linear curve, called a *polyline*, that represents the road-centerline. Moreover, to each road is associated a list of

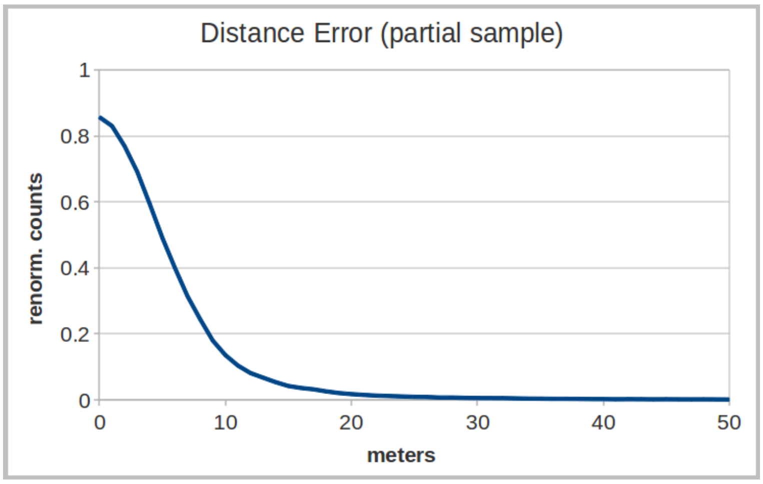

Figure 2.2: Error distribution for distance, $\sigma \simeq 8.5$ meters. The distribution has been computed on a limited sample of data unambiguously matchable to long straight road segments.

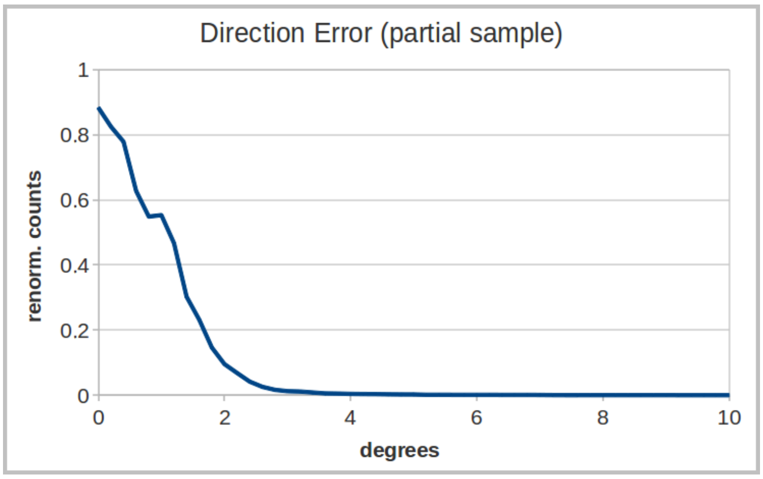

Figure 2.3: Error distribution for direction, $\sigma \simeq 1.2$ degrees. The distribution has been computed on a limited sample of data unambiguously matchable to long straight road segments, with $vel > 10km/h$.

attributes that further describe it, such as road name, road type, speed limit, one-way status, etc.

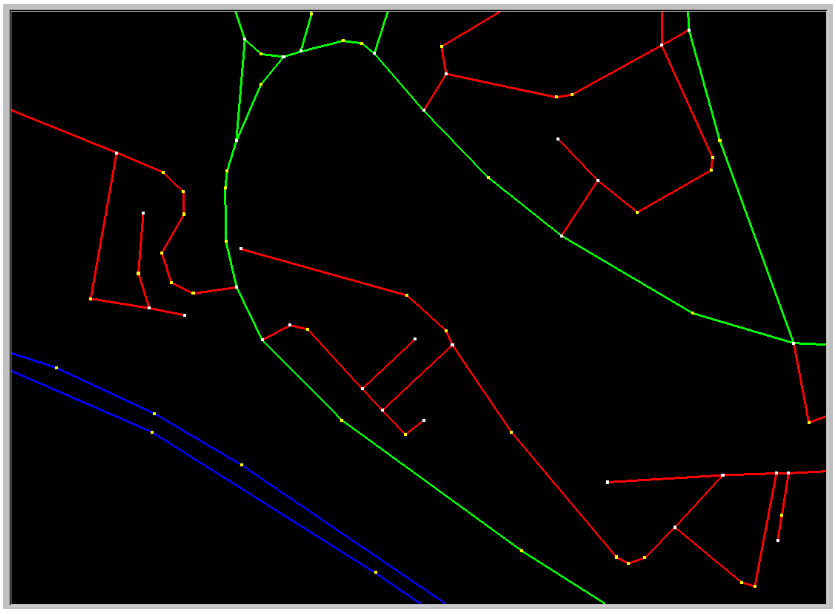

Figure 2.4: Portion of road network data from the city of Florence, Italy. Each *road element* is colored according to the speed limit reported in its attributes. The white square dots represent *road nodes* and the yellow square dots represent *shape points*.

However, before using the road data for our map-matching task, the original database format is reorganized so to be more efficient for our purposes. Firstly, we reconstruct the topological structure of the road network and save all the relevant data on a specific file. This information is vital for the map-matching process but it is not explicitly available in the original shapefile format. Then, polyline shape data is checked for vertex spatial sampling irregularities and then saved into a second file. Finally, road attributes relevant to the map-matching process are filtered from the original attribute list and saved into a third file. Obviously, the format rearrangement that we apply is fully reversible, so that it is possible to export our results on the road network back into Shapefile format for an easier data sharing. Tab. 2.2 describes the final format for a road element and Fig. 2.4 shows a graphic representation of road network data.

Road ID	Road identification number.
Road Shape	Road shape is described by *polylines*. The position of each polyline is defined by an ordered list of vertexes, whose geographical coordinates are given in latitude and longitude in the WGS84 geodetic reference system. The segments that make up the polyline are referred to as *arcs*, the first and the last vertexes of the polyline are referred to as *nodes*, while the other vertexes are called *shape points*.
Road Access	Describes if the road is two-way accessible (value = 0), one-way accessible from the front node (value = 1), one-way accessible from the back node (value = 2) or two-way restricted (value = 3). Two-ways restricted road are usually located in the central historical areas of cities, where access is limited to residents and public vehicles.
Road Type	Describes the level of importance of the road, as for nominal capacity and intersection type. It ranges from 0 to 5, where 0 is associated to highways and other long-distance roads and 5 represents neighborhood roads.
Road Length	Describes the length of the polyline and is expressed in meters.
Speed Limit	Describes the nominal driving speed for the road under optimal traffic conditions and is expressed in Km/h.

Table 2.2: The components of a Road Element

2.3 Florence test dataset

The test dataset chosen to present the results of our map-matching procedure covers a rectangular area centered on the province of Florence and refers to the whole month of March 2008. The spacial region of the dataset (Fig. 2.5) is bounded by the following coordinates in the WGS84 geodetic reference system,

South-West corner (lat, lon): 43.450, 10.710 (arc degrees)

North-East corner (lat, lon): 44.240, 11.755 (arc degrees)

and has an approximate size of 84 Km in width and 88 Km in height.

The raw dataset is made by 17.3 million GPS data records, belonging to 35'273 different vehicles. It is important to note that this dataset collects the data from all the vehicles equipped with a GPS device from Octo Telematics that drove inside the defined area during the month of March 2008. This means that it refers not only to the movements of the vehicles owned by residents in the area, but also to the vehicles coming from outside or in transit through the region.

Not all the data in the dataset will actually be used in the map-matching. The algorithm will be used to reconstruct only the vehicle paths lying inside the metropolitan area of the city of Florence. More precisely, as shown in Fig. 2.5, this is a circular area with a radius of 10 Km centered on the city landmark of the Fortezza da Basso (WGS84 coordinates: 43.779497, 11.248106). Restricting the use of data to a subset of the whole dataset is important in order to identify correctly the extent and the nature of GPS trajectories, distinguishing between trajectories representing a transit trough the study area, an inward/outward trip or an internal journey. More details on this regard will be given in Chapter 3.

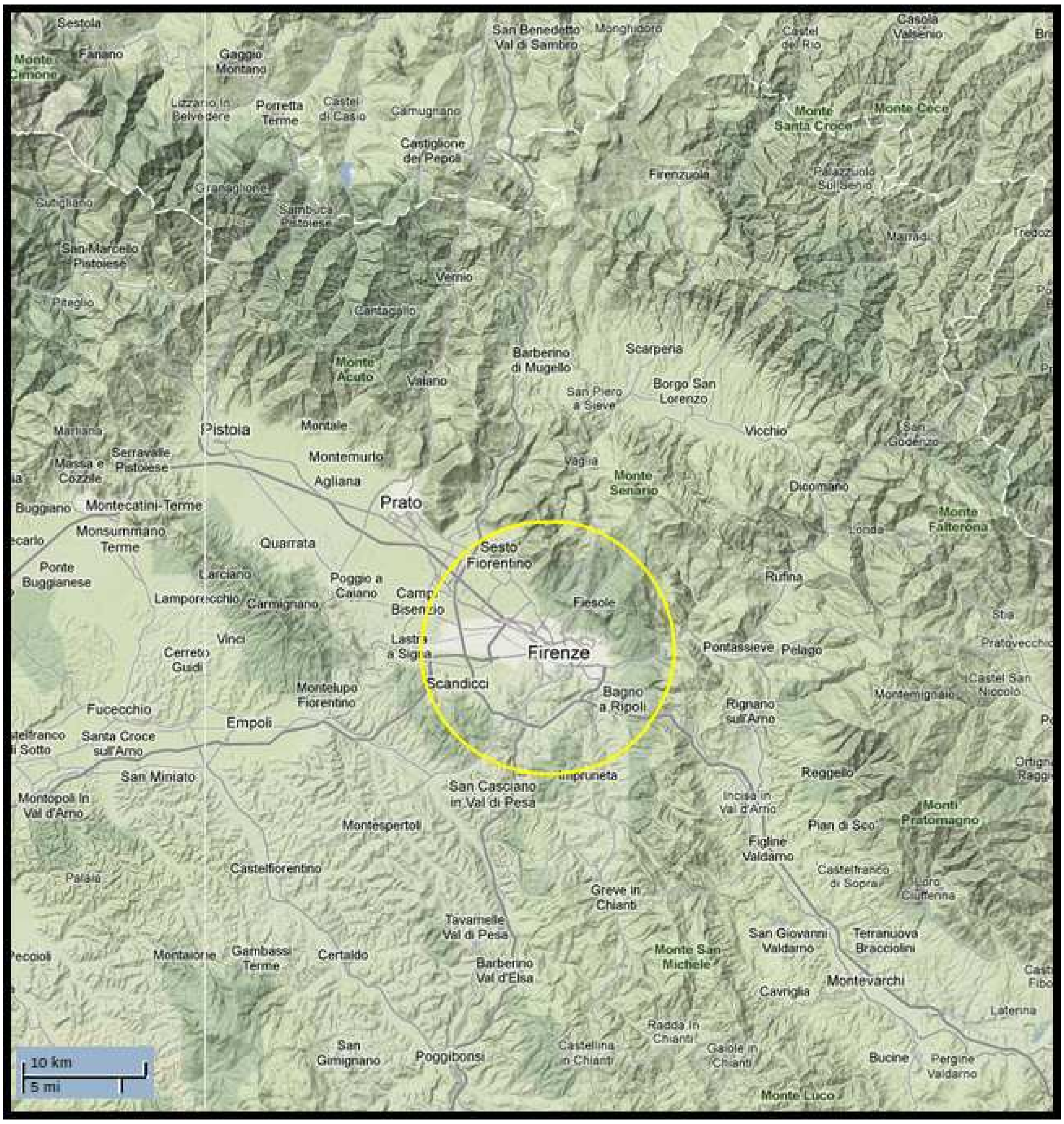

Figure 2.5: Spacial extension of the raw test dataset. The area covers the whole province of Florence, Italy, and part of the territories of the surrounding provinces. The yellow circle defines the area of the city of Florence where vehicle paths are reconstructed. (c)2011 Google, map data (c)2011 Tele Atlas.

Chapter 3

Trajectory Aggregation

The aim of the trajectory aggregation phase is to process the raw ensemble of GPS data into an ordered sequence of well defined GPS trajectories, ready to be map-matched. Raw data is grouped according to vehicle ID and then every group of data corresponding to a vehicle is elaborated separately. Trajectory aggregation consists in a sequence of four difference steps, each resulting in possible data modification or removal.

The first step reconstructs the temporal order of the vehicle dataset and consolidates the continuity of engine state information. All data records bearing no useful information are immediately removed, such as multiple copies of the same record or records storing measures taken during the vehicle cruise in conditions of no GPS signal lock. Then, the vehicle dataset is rearranged with respect to the temporal order and to the proper engine state sequence of cruise → engine stop → engine start → cruise.

The second step removes known anomalies in the consistency of the data sequence. This means, for example, that vehicle stops that are too short to be relevant or data records that are spatially much closer than expected are removed. Specific attention is also put on data records corresponding to engine-on events. This kind of data stores the measurements taken exactly when the vehicle - and the GPS device with it - is turned on and is thus very often taken in conditions of no GPS lock, resulting in inaccurate readings.

The third step defines the spatio-temporal interval of interest within the dataset. All data records falling outside the chosen interval are removed and the information on entering into and exiting from the interval is integrated into the data via a new ad-hoc parameter (see section 3.1.1 below for more details). This is a necessary step for an off-line map-matching application based on a localized raw dataset because it allows for a reliable identification of all spatial interruptions along the sequence of GPS data.

The fourth step identifies the intervals of reliability and relevancy of GPS data sequences and performs a final consistency check. For example, if two subsequent data records belonging to the same GPS trajectory are found to be too much distant in space or time then what expected in a regular sampling under good GPS signal conditions, then the trajectory is split in two parts and the information on the probable signal loss in-between is integrated into the relevant data records via a new ad-hoc parameter (see section 3.1.1 below for more details).

At the end of the whole process, the aggregated sequence of GPS data is checked to verify its overall compliance to a set of consistency properties (see section 3.1.6).

3.1 Trajectory Aggregation Procedure

The sequence of trajectory aggregation steps that were previously delineated will be here described in details. References to the single elements of a GPS datum will be done according to the definitions provided in Tab. 2.1. The procedures described below assume that the data has already been grouped by vehicle ID and that each vehicle dataset is processed separately.

3.1.1 Parameters and definitions

In the process of trajectory aggregation, several parameters are needed to define how to reorganize, modify or remove data. Tab. 3.1 reports those parameters, along with their default value and a short description of their purpose.

MAX_CRUISE_SPEED	250 Km/h	max allowed speed for a cruise record.
MAX_DELAY_TIME	90 sec	max allowed distance in time between an engine-off record and the one preceding it to upgrade *gps_q* for the former.
MAX_STOP_SPACE	100 m	max allowed distance in space between engine-off and engine-on records to upgrade *gps_q* for the latter.
TINY_STOP_TIME	5 sec	min allowed duration for a vehicle stop (temporary)
SMALL_STOP_TIME	30 sec	min allowed duration for a vehicle stop
TINY_STEP_SPACE	30 m	min allowed distance in space between subsequent records in a trajectory
HUGE_STEP_TIME	3'600 sec	max allowed distance in time between subsequent records in a trajectory
HUGE_STEP_SPACE	3'000 m	max allowed distance in space between subsequent records in a trajectory
MIN_TRAJ_TIME	20 sec	min duration for a valid trajectory
MIN_CAR_RECORDS	5 records	min number of data records necessary for a valid vehicle dataset

Table 3.1: Definition of the parameters used during trajectory aggregation

During trajectory aggregation, data records are arranged in trajectories. A regular trajectory begins with an engine-on record and ends with an engine-off record, but the trajectories are not all regular. A trajectory is irregular if one or both its ends are interrupted, whether because the vehicle crossed the spatio-temporal boundaries of the dataset or because the on-board device lost the GPS signal (or failed for whatever reason) in the middle of the trip. A new parameter, called trajectory state (trj_s), is defined to organize the sequence of data into trajectories and to describe the different types of trajectory interruptions. The value of trj_s is initialized to equal eng_s for each data record and is changed, when appropriate, during the different steps of trajectory aggregation. In the following sections, the context of definition of each of the new values will be described in details. The list of valid values for trj_s is described by Tab. 3.2 below.

Value of trj_s	**Corresponding meaning**
0, 1, 2	engine on, cruise, engine off
3, 4	entering into/exiting from dataset temporal bounds
5, 6	entering into/exiting from dataset spatial bounds
7, 8	GPS signal recovered, GPS signal lost

Table 3.2: Acceptable values for the trajectory state

Whenever it is simply required to know if a data record is at the beginning, in the middle or at the end of a trajectory, we will refer to a simplified description of the trajectory state. Labeling this quantity ts, its relation with trj_s is defined as follows:

$ts = 0$	for $trj\_s = 0, 3, 5, 7$, referring to the beginning of a traj.
$ts = 1$	for $trj\_s = 1$, referring to data internal to the traj.
$ts = 2$	for $trj\_s = 2, 4, 6, 8$, referring to the end of a traj.

3.1.2 Reordering of engine state information

First of all, the raw vehicle dataset is reorganized in increasing temporal order. If two records have the same timestamp but different values for eng_s, the one with the smaller *eng_s* is put before the other.

Then, data records that carry no useful information on vehicle movements are removed:

- Multiple copies of the same record are all removed but one.
- Multiple copies of the same record that differ only in *ds* are all removed but one, which is assigned the sum of all *ds* values.
- Records with $eng\_s = 1$ and $gps\_q = 1$ are removed.
- Records with $eng\_s = 1$ and $ds < TINY\_SPACE$ are removed.

Finally, the sequence of *eng_s* values is considered. To evaluate its consistency, for each couple of subsequent records we check the quantity
$seq = 10 \cdot eng\_s_1 + eng\_s_2$:
Consistent values are: 02, 01, 11, 12, 20.
Inconsistent values are: 00, 10, 21, 22.

When inconsistent records are found, the following actions are taken:

seq = 00	the first record is removed
seq = 10	trj_s_1 is set to 8 This signals that we lack information on some of the vehicle's movements before the second record.
seq = 21	trj_s_2 is set to 7 This signals that we lack information on some of the vehicle's movements after the first record.
seq = 22	the second record is removed

3.1.3 Removal of known data consistency anomalies

Engine-on data ($eng\_s = 0$) often has inaccurate readings, because very rarely the GPS device locks to the signal as soon as the vehicle is turned on. In order to regularize them, the following actions are taken on each engine-on record:

- The values of *vel*, *ang* and *ds* are all forcibly initialized to zero.
- If the previous record is an engine-off ($eng\_s = 2$) record and its euclidean distance to the current record is less then MAX_STOP_SPACE, than *gps_q*

of the engine-on record is upgraded to the value of the engine-off record.

Then, for each couple of subsequent records of the same trajectory, the euclidean distance (dse) is computed and compared to the value of ds stored in the second record of each couple. Whenever $ds < dse$, then ds is reset to equal dse.

At this point, the duration of all the proper vehicle stops (an engine-off record followed by an engine-on one) are evaluated. If the stop duration is shorter than $TINY\_STOP\_TIME$ than the stop is removed from the sequence of data:

- Both engine-off and engine-on records are removed.
- The ds of the record following the engine-on datum is incremented with the ds of the engine-off datum.

Then, the value of ds is checked for all data records that are internal or at the end of a trajectory ($ts = 1$ and $ts = 2$ respectively, see 3.1.1). If it is found that $ds < TINY\_STEP\_SPACE$, then different actions are taken depending on the trajectory state of the current record (ts_2) and the one preceding it (ts_1). Defining $seq = 10 \cdot ts_1 + ts_2$, the following table presents the different cases:

seq	**previous record**	**current record**	**following record**
01, 11		removed	ds++
02	removed	removed	
12	removed	ds++, gps_q++	

Where the following symbols have been used:

- ds++ means that ds is incremented with ds of the preceding record.
- gps_q++ means that gps_q is upgraded to the value of the preceding record if the distance in time between the two records is less than MAX_DELAY_TIME.

At the end of this check, the duration of all the proper vehicle stops is checked again, this time against a tighter threshold. If the stop duration is shorter than $SMALL\_STOP\_TIME$ than the stop is removed from the sequence of data, following the same scheme described above.

Finally, vel is checked for every record.
Whenever $vel > MAX\_CRUISE\_SPEED$ Km/h, then vel is reset to equal the limit.

3.1.4 Definition of the spatio-temporal boundaries

First of all, we identify the trajectory interruptions due to the crossing of the temporal boundaries of the dataset:

- If the first record of the vehicle dataset has $eng\_s = 1$, then its trj_s is set to 3.
- If the last record of the vehicle dataset has $eng\_s = 1$, then its trj_s is set to 4.

Then, we identify the trajectory interruptions due to the crossing of the spatial boundaries of the dataset.

- Each record is checked to identify if it falls inside or outside of the area of interest.
- All records falling outside of the chosen area are removed.
- Records falling inside the chosen area are treated according to the following table.

previous and following records are inside the area	record is kept
previous and following records are outside the area	record is removed

previous record is inside, following record is outside	if $ts = 2$, record is kept
	if $ts = 0$, record is removed
	if $ts = 1$, trj_s is set to 6

previous record is outside, following record is inside	if $ts = 2$, record is removed
	if $ts = 0$, record is kept
	if $ts = 1$, trj_s is set to 5

3.1.5 Identification of data reliability intervals

In case of GPS signal loss or degradation during the vehicle's movement, the on-board GPS device delays the storage of data until the signal is recovered. We treat this kind of situation as a trajectory interruption. For each couple of subsequent records in a trajectory we check the distance in time (dt) and the distance in space (ds of the second record). If it is found that $ds > HUGE\_STEP\_SPACE$ or $dt > HUGE\_STEP\_TIME$, then different actions are taken depending on the trajectory state of the two records. Defining $seq = 10 \cdot ts_1 + ts_2$, the following table presents the different cases:

seq	**first record**	**second record**
01	record is removed	trj_s is set to 7
02	record is removed	record is removed
11	trj_s is set to 8	trj_s is set to 7
12	trj_s is set to 8	record is removed

Then, we check the overall travel time of each trajectory (dT).
If $dT < MIN\_TRAJ\_TIME$, the trajectory is too short to be relevant and is thus removed from the dataset.

Finally, we count the number of valid records in the vehicle dataset (N). If $N < MIN\_CAR\_RECORDS$, then the whole dataset of the vehicle is discarded.

3.1.6 Consistency specifications of the aggregated data

As a consequence of the procedures described in the previous sections, vehicle datasets are now self-consistent and provided with a common set of properties.

The sequence of values of trj_s describes, in a continuous and coherent manner, the evolution in time of four important quantities:

- The vehicle's driving state ($trj\_s = 0, 1, 2$). Identifies when the vehicle is traveling and when is parked.
- The vehicle's presence during the time interval of interest ($trj\_s = 3, 4$).
- The vehicle's presence inside the area of interest ($trj\_s = 5, 6$). Identifies the spatial type of a trajectory: internal, inbound, outbound or in transit.

- The availability of reliable data for the vehicle ($trj\_s = 7, 8$). Identifies the intervals of availability and unavailability of information on vehicle state and position.

Moreover, the following properties hold:

- All vehicle datasets are composed by at least 5 records.
- All trajectories are composed by at least two data records and their overall duration is longer than 20 seconds.
- The duration of vehicle stops is no shorter than 30 seconds.
- The distance in time between subsequent data records in a trajectory is no longer than 1 hour.
- The distance in space between subsequent data records in a trajectory is comprised between 30 m and 3 Km.
- All records have a speed no greater than 250 Km/h.
- There exist no cruise records ($eng\_s = 1$) taken in conditions of no GPS lock ($gps\_q = 0$).
- All engine-on records ($eng\_s = 1$) have $vel = 0$, $ang = 0$, $ds = 0$ and the value of gps_q now appropriately describes the uncertainty on lat and lon.
- With respect to the raw original dataset, records have been removed or modified, but no extra record has been added.
- The sequence of trajectory aggregation steps is self-consistent in the sense that if it is applied to a dataset that has already been aggregated it does not produce any change in it.

3.2 Results and Applications

3.2.1 Trajectory aggregation for Florence dataset

In this section we present the results of the trajectory aggregation procedure on the Florence test dataset. First we report the statistics on data removal, trajectory type and quality, then we show some examples of trajectory aggregation.

The table below reports the number of data records removed and modified during the four steps of the procedure:

	removed	**modified**
Reordering of engine state information	364’639	6’740
Removal of known data consistency anomalies	911’068	2’859’610
Definition of the spatio-temporal boundaries	11’959’899	291’753
Identification of data reliability intervals	17’443	19’801
TOTAL	13’253’049	3’177’904

The overall impact of the trajectory aggregation procedure on the number of records is thus:

Records in the raw dataset	17’295’057
Records removed during aggregation	13’253’049
Records in the final dataset	4’042’008

We note that, from the initial sample of 17.3 million records:

- The majority of removed records is due to the definition of the spatio-temporal boundaries and do not represent errors (90% of total removed records).
- The number of records removed because of errors and anomalies is much smaller in comparison (10% of total removed records).
- The number of modified records amounts to 3.2 million records, 18% of the original dataset.
- The records in the final dataset are 4.0 millions, 76% of the raw data already within the specified spatio-temporal boundaries.

We observe that the number of drivers in the dataset has decreased from 35’273 drivers present in the raw dataset to 25’048 drivers present in the aggregated one.

For what concerns trajectory definition, we find that, from a total of 640’797 trajectories:

- 66% are regular (initial $eng\_s = 0$ and final $eng\_s = 2$).
- 31% are interrupted by the spatio-temporal boundaries (initial $eng\_s = 3, 5$ or final $eng\_s = 4, 6$).

- 3% are interrupted because of signal loss or device failure (initial $eng\_s = 7$ or final $eng\_s = 8$).

Examining the quality of regular trajectories, we have that for 57% of them both the first and the last record have a very accurate positioning information ($gps\_q = 3$).

Comparing Fig. 3.1 and 3.2, it is possible to verify qualitatively the impact of this phase on the quality of data. For each couple of subsequent records in a trajectory we read the distance in time (dt) and the distance in space (ds of the second record). Then we check the value of $seq = 10 \cdot ts_1 + ts_2$, that can have only four legal values: 01, 02, 11, 12.

For each of the possible values of seq, we compute the distribution of ds values, weighting each count with dt. In this way we obtain an integrated measure of the distribution of the spatial and temporal intervals between data.

Fig. 3.1 shows the distributions for the four seq values before trajectory aggregation, while Fig. 3.2 shows the same distributions after the procedures described in this chapter have been applied to the data. We remark that for Fig. 3.1 we used only the raw data lying inside the same spatial boundaries as the aggregated data.

We observe that Fig. 3.2 is smoother and cleaner than Fig. 3.1 as a consequence of trajectory aggregation. In particular, it is apparent that after this procedure all subsequent data of a trajectory are separated by no more than 3 Km. Moreover, we observe that, after aggregation, the distributions for $seq = 02$ and $seq = 12$ show an increase of signal around the 2 Km mark. This is due to the step described in section 3.1.3. If the last two data records of a trajectory are spatially too close to each other, than the first is erased. As a consequence, the last record will then have a much longer spatial separation from the record preceding it, typically of the order of 2 Km.

Some interesting information on vehicle movements can be derived from the elaborated data just as a consequence of trajectory aggregation. For example, Fig. 3.3 traces the movements of a certain vehicle during a week. The figure shows how straightforward it is to identify how far and for how long does the vehicle drive, as well as when it exits from or enters back into the test area.

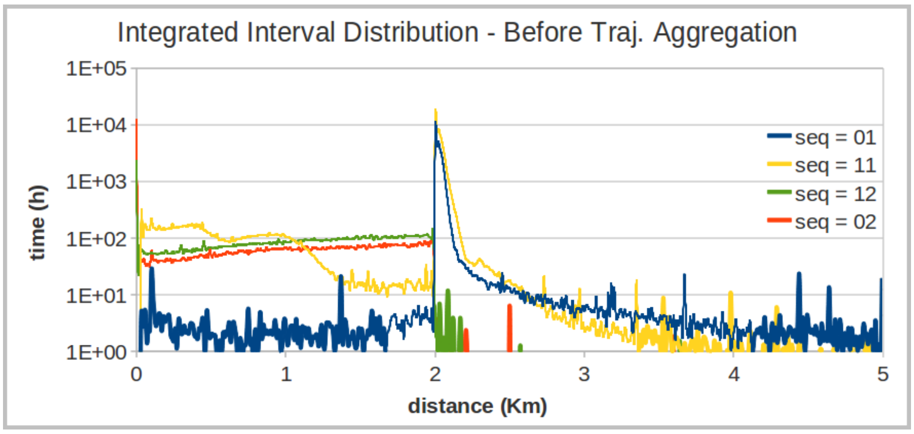

Figure 3.1: Integrated interval distribution before trajectory aggregation.

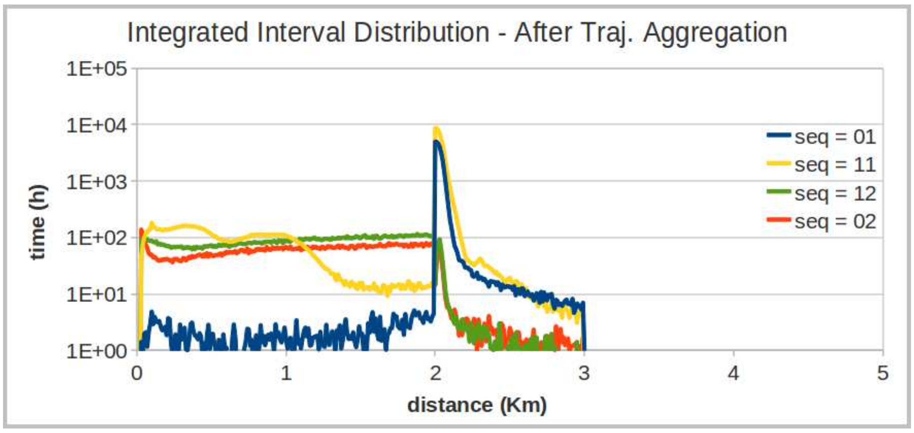

Figure 3.2: Integrated interval distribution after trajectory aggregation.

Briefly, the vehicle drives for 20 Km, stops for nearly 6 hours and then restarts and leaves the test area. The vehicle comes back almost a day and a half later, entering the boundary from roughly 15 Km afar from where it exited, and finally stops after a final 20 Km drive. Then, a similar behavior follows.

Fig. 3.4, instead, describes in more details the different layers of information carried by the sequence of trj_s values. The figure shows the evolution during a week of two quantities: the driving state and the presence inside the test area. It is clear from the image that the vehicle follows a very regular pattern during workdays: it leaves home between 6.00am and 7.00am, performs some occasional trips in the middle of the morning and comes back home at around 8.00pm and earlier on Friday. On Saturday, instead, it makes a small trip in the morning and later leaves the test area, only to come back the day after, on Sunday.

3.2.2 Statistical laws in urban mobility

The trajectory aggregation procedure groups vehicle data into trajectories. Having access to big samples of data it is thus possible to study the statistical properties of vehicle's trips and stops as described by the sequence of trajectories.

The study of these properties may help to identify global laws in the dynamic system of vehicular traffic, and also to shed some light on the cognitive processes behind drivers' decisions. What follows is part of a detailed analysis and modeling undertaken by the Physics of the City Laboratory on the aggregated data from the Florence test dataset [2], [3]. The sample is restricted to resident vehicles having a continuous sequence of regular trajectories.

Fig. 3.5 shows the distribution of the total daily trip length (L) for the vehicles in the set. We observe that the distribution follows an exponential law

$$P(L) \propto e^{-L/L_0},$$

where L_0 represents a length of 24.9 Km.

The fact that this distribution follows an exponential profile can be interpreted by the principle of maximum entropy, assuming the independent

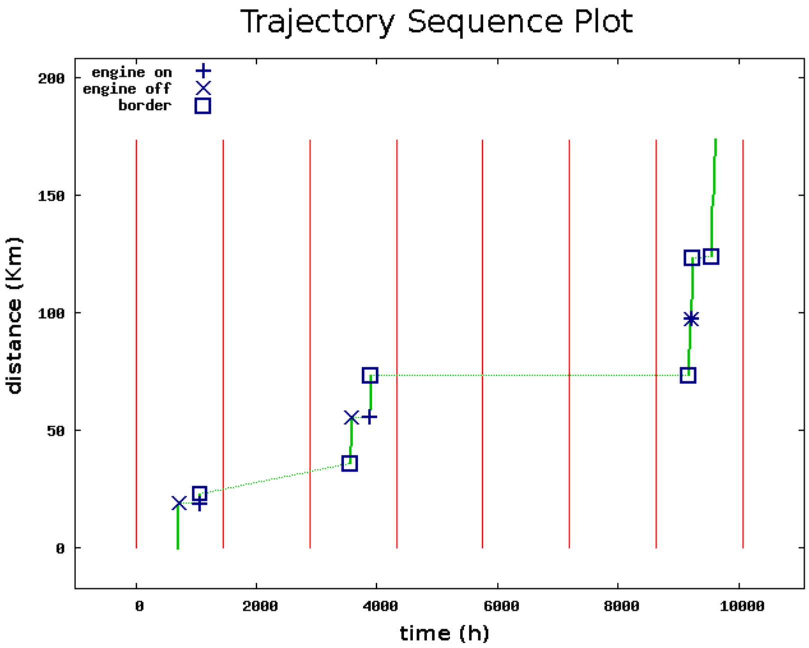

Figure 3.3: Trajectory sequence plot. The sequence of vehicle movements starts from the origin and is represented by a green line, + represents engine-on data, × represents engine-off data and □ represents spatial boundary crossing. The sequence extends for a week, vertical lines divide the timespan in separate days.

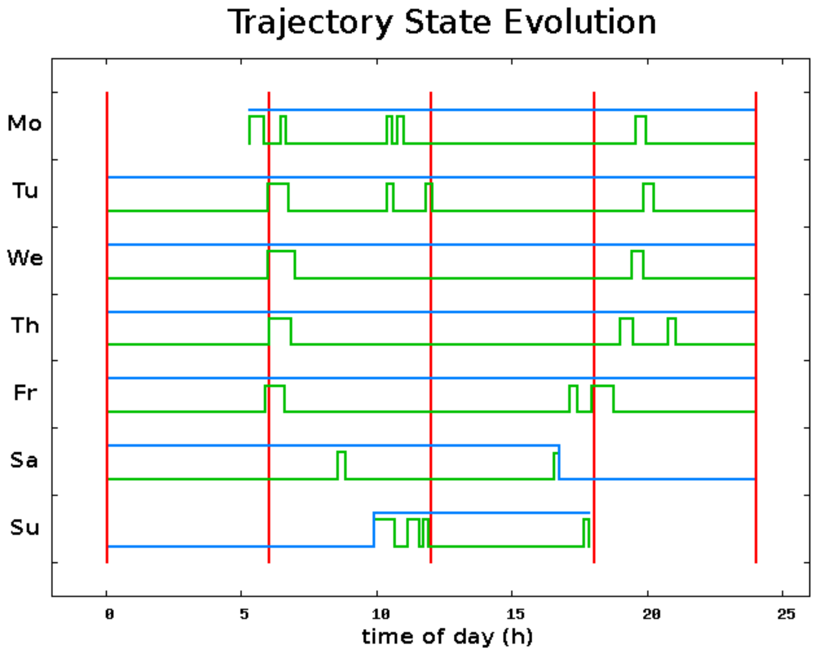

Figure 3.4: Trajectory state evolution. The blue line describes vehicle presence inside the test area: high means present, low means absent. The green line describes the driving state: high means driving, low means parked.

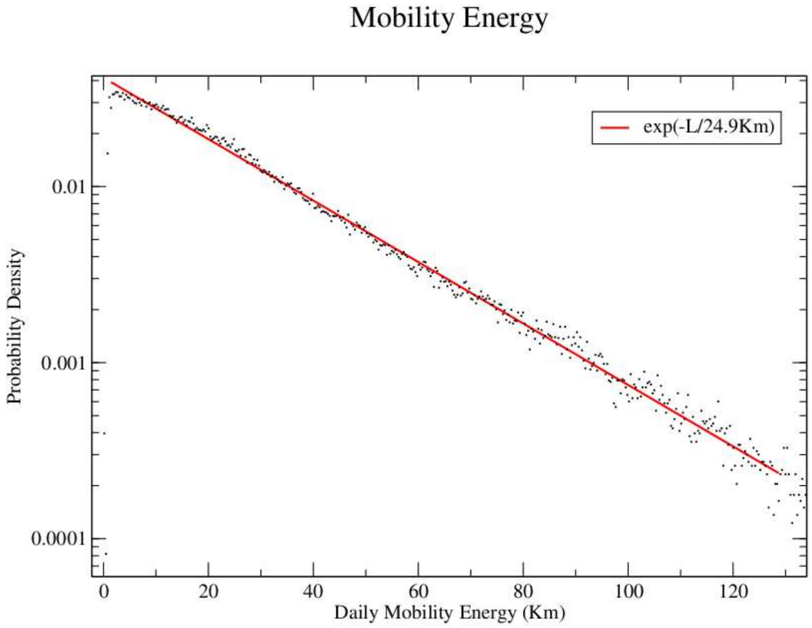

Figure 3.5: Mobility energy distribution, from [2].

behavior of each vehicle and the existence of an average daily trip length for drivers. Following this interpretation, the length L can be described as a *mobility energy* available on average each day to be spent driving. This mobility energy can be thought as the maximum effort the driver is willing to undertake daily. This interpretation is coherent with findings from other studies on the subject [24].

Fig. 3.6, instead, shows the distribution of vehicle downtimes, that is the duration of vehicle stops. Obviously, vehicle downtimes correspond to the time spent by the driver in activities other than driving. The distribution has a base trend that follows a power law

$$P(T) \propto T^{-0.95},$$

where the fit has been computed on the more relevant part of the curve, comprising 65% of total counts.

The power-law fit can be interpreted with a simple model based on the observation that the total time allotted daily to our activities is limited by our

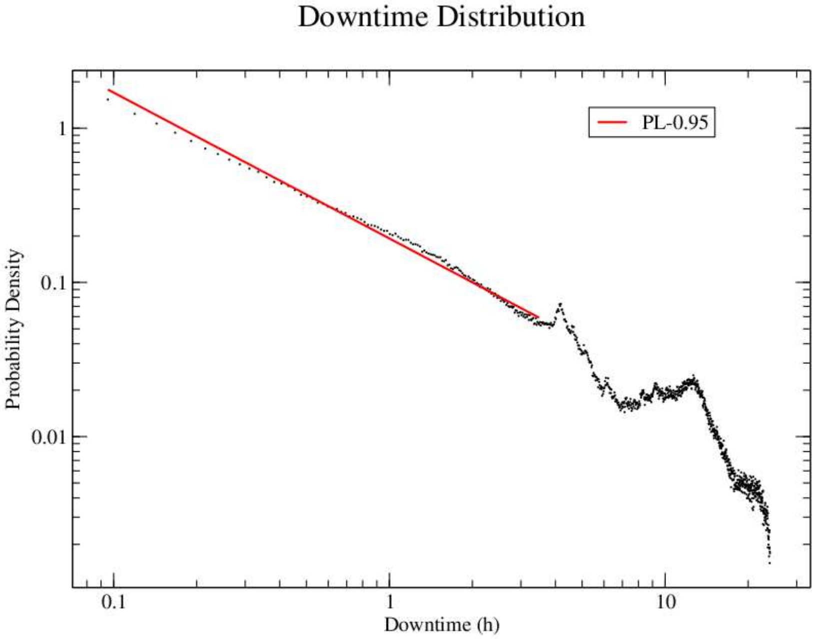

Figure 3.6: Vehicle downtime distribution, from [2].

circadian rhythm. If we assume to have 24 hours at disposal and sequentially we decide, with uniform probability, how much time to spend on each new activity basing just on the time that remains, it can be shown that we obtain a downtime distribution $\propto T^{-1}$, coherently with what we observe [2].

We observe that the constraint that we assume to explain the downtime distribution is much stronger than the constraint we assume to explain the total trip length distribution. While for downtime we assume a fixed total daily time, for the total trip length we assume a constraint on maximum length that has to be respected only on average.

However, both models suggest that on a macroscopic level what we observe is mainly a statistical behavior, with very few traces of deterministic human processes. For example, downtime distribution does show several peaks superimposed to this trend, and these peaks seem characteristic of human activities with a defined duration.

Chapter 4

Data Matching by Affinity

The aim of this phase is to match data records to the road network. As we have seen in section 2, this matching is not obvious because of the uncertainties in positioning measurements and in road digitization.

We chose a bayesian approach to the problem. Following the rule of conditional probability,

$$P(R|D) \propto P(D|R) \cdot P(R), \tag{4.1}$$

we compute the likelihood $P(R|D)$ that each datum D belongs to its surrounding roads R from assumptions on the prior probability $P(R)$ for a vehicle to be driving on road R and assumptions on the positioning probability $P(D|R)$ to measure datum D while driving on road R.

The prior probability $P(R)$ is simply a quantity proportional to the average road flux and its importance is crucial for the correct determination of likelihoods. The positioning probability $P(D|R)$ takes into account both uncertainties in vehicle position (lat, lon) and vehicle heading (ang) that are modeled with simple assumptions and calibrated according to data. In the following sections these quantities will be described in more details.

Thus, the likelihood $P(R|D)$ allows to identify the most reasonable candidates for matching in a meaningful quantitative way. For this reason we call this quantity *affinity* between road R and datum D. To each data record we associate all the matches with high affinity and not just the single most affine one. This is important for the last phase of map-matching, the global path reconstruction of GPS trajectories.

4.1 Data Matching Procedure

4.1.1 Categories of data accuracy

When determining the affinity between road and datum, vehicle speed (vel) and signal quality (gps_q) are very important variables. In fact, the measure on vehicle heading (ang) is derived from two subsequent GPS readings on position, so that its accuracy is obviously dependent on speed and signal quality. Moreover, for engine-on and engine-off data, that correspond to parked vehicles, there is not a strict correlation between vehicle heading and road direction. Indeed, vehicles are not necessarily parked along a road and parallel to it, they could be parked at a different angle or they could be in a garage or in a parking lot, far from any road at all.

We take these considerations into account by defining three simple speed categories:

Parked: all records with $eng\_s = 0$ or $eng\_s = 2$.
Low-speed: all records with $gps\_q < 3$ or $vel < 5$ Km/h that are not parked.
High-speed: all remaining records.

For parked data records we define a second characterization. When two regular trajectories (see section 3.1.1) follow each other, we check how close the engine-off record of the first is to the engine-on record of the second. If their euclidean distance is less then MAX_STOP_SPACE (see Tab. 3.1), we call the parked records *joint*. For all the other engine-on and engine-off data that do not meet these conditions we apply the term *disjoint*.

As we will see in the following sections, data from each category will be processed differently according to its characteristics.

4.1.2 Definition of positioning probability

In order to define $P(D|R)$, the positioning probability for road R, we first need to define $P(D|A)$, the positioning probability for road arc A. In this definition we want to take into account the relative distance (d) between datum and arc, the length of the arc (L) and the difference (θ) between vehicle heading and arc direction, as shown in Fig. 4.1.

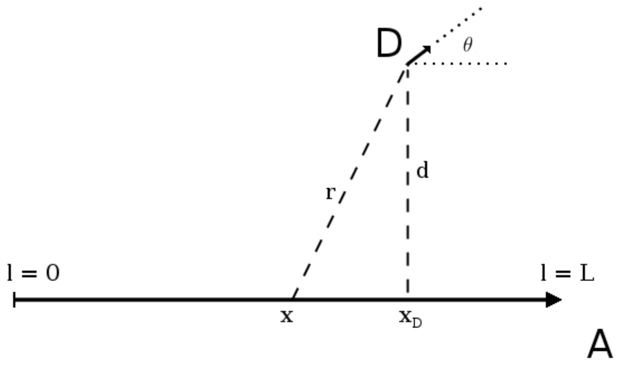

Figure 4.1: Schematic configuration of road arc A and datum D for the definition of positioning probability.

We begin by modeling the probability $P(r)$ of positioning the vehicle at a distance r from its actual location:

$$P(r) \propto \frac{1}{1 + (\frac{r}{\sigma_r})^2},$$

where σ_r is a parameter that represents the scale of error dispersion. According to the estimate on distance error distribution (Fig. 2.2), we assigned $\sigma_r = 9$ meters.

Regarding the shape of this error distribution, other authors make different choices and typically chose a Gaussian profile to model this error [11], [14]. However, our choice to model this probability with a Cauchy distribution is motivated by the observation of the error distribution (Fig. 2.2) estimated in section 2, that shows a decay in the tails that is much slower than that of a

Gaussian. Clearly, using a Cauchy profile we are just approximating the real distribution, in particular close to the peak, but we are also allowing for an high sensibility on tails, that represent the data that are the less obvious to match.

We can then define the probability $P_D(x)$ of measuring datum D when the vehicle is located at position x along the arc:

$$P_D(x) = \frac{C}{{\sigma_r}^2 + d^2 + (x - x_D)^2},$$

where C is a normalization factor, d is the distance between datum D and the line that extends arc A and x_D is the projection of datum D on A (see Fig. 4.1).

So, the integrated probability $P_D([0, L])$ of measuring datum D when the vehicle is driving somewhere along the arc A is:

$$\begin{aligned} P_D([0,L]) &= \int_0^L P_D(x)dx = C\int_0^L \frac{dx}{{\sigma_r}^2 + d^2 + (x - x_D)^2} \\ &= \frac{C}{d'}\int_{\frac{-x_D}{d'}}^{\frac{L-x_D}{d'}} \frac{dt}{1+t^2} \\ &= \frac{C}{d'}\left[arctan\left(\frac{L-x_D}{d'}\right) + arctan\left(\frac{x_D}{d'}\right)\right], \end{aligned} \tag{4.2}$$

where $d'^2 = {\sigma_r}^2 + d^2$.

Having defined $P_D(x)$ and $P_D([0, L])$, we can now compute the weighted projection $\langle x \rangle_A$ of datum D over arc A:

$$\langle x \rangle_A = \frac{\int_0^L x \cdot P_D(x)dx}{P_D([0,L])} = x_D + \frac{d'}{2} \cdot \frac{ln\left[1 + \left(\frac{L-x_D}{d'}\right)^2\right] - ln\left[1 + \left(\frac{x_D}{d'}\right)^2\right]}{\left[arctan\left(\frac{L-x_D}{d'}\right) + arctan\left(\frac{x_D}{d'}\right)\right]}.$$

The computation $\langle x \rangle_A$ can be time consuming. So, in cases when saving computation time is crucial, $\langle x \rangle_A$ can be approximated by $x_{closest}$, the closest point of the arc A to datum D. Obviously, $x_{closest} = x_D$ if $0 \geq x_D \geq L$, while otherwise $x_{closest}$ coincides with the closest extreme of arc A. Both $\langle x \rangle_A$ and $x_{closest}$ obviously fall by definition inside arc A, but using $x_{closest}$ introduces a bias by giving extra weight to the extremes of arc A.

We then consider the angle θ between vehicle heading and arc direction and model the probability $P_D(\theta)$ of such a deviation with:

$$P_D(\theta) = \frac{1}{1 + (\frac{\theta}{\sigma_\theta})^2}, \tag{4.3}$$

where σ_θ is a parameter that represents the scale of error dispersion. The error distribution for direction (Fig. 2.3) estimated in section 2 was determined restricting to long straight road segments and fast vehicles and thus fixes just a lower limit for σ_θ. In order to take into account shorter segments, road bending and lower velocities we over-estimate it to $\sigma_\theta = 5$ degrees. Again, we modeled this distribution with a Cauchy profile to insure that the overall positioning probability has an high sensibility on tails where it is most needed to distinguish between difficult matches.

Given the unreliability of vehicle heading information for low-speed and parked data (see section 4.1.1) $P_D(\theta)$ is computed as stated only for high-speed data. For low-speed and parked data a value of $P_D(\theta) = 0.5$ is used, that corresponds to assuming and average value of $\theta = \sigma_\theta$.

From equations 4.2 and 4.3 we can now define $P(D|A)$ the positioning probability for arc A:

$$\begin{aligned} P(D|A) &= P_D([0, L]) \cdot P_D(\theta) \qquad (4.4) \\ &= \frac{\sigma_r}{\pi \cdot d'} \left[\frac{1}{1 + (\frac{\theta}{\sigma_\theta})^2} \right] \left[arctan \left(\frac{L - x_D}{d'} \right) + arctan \left(\frac{x_D}{d'} \right) \right], \end{aligned}$$

where C has been explicitly set to $C = \sigma_r/\pi$.

Finally, the definition of $P(D|R)$, the positioning probability on road R, is straightforward:

$$P(D|R) = \sum_{A_i \in R} P(D|A_i). \tag{4.5}$$

The weighted projection $\langle x \rangle_D$ of datum D over road R is:

$$\langle x \rangle_D = \sum_{A_i \in R} \frac{(\langle x \rangle_{A_i} + L_i) \cdot P(D|A_i)}{P(D|R)},$$

where $L_i = \sum_{j<i} L_{A_j}$ and $L_0 = 0$.

4.1.3 Definition of prior road probability

The prior road probability $P(R)$ estimates the probability that a vehicle is driving on road R, independently from the outcome of any positioning measurement. It is, in other words, a measure of the average flux of road R.

In the context of a real-time application of this map-matching procedure, the average road flux is exactly the quantity that will be used for $P(R)$. However, for the current off-line application, we need to estimate it in a different way. A rough but fast estimate could be derived from the road type information (Tab. 2.2). $P(R)$ could be set to 6 scaling values according to the road type of R. However, the pre-determined value of road type could be often far from representing the real intensity of use of a road, and setting the scaling appropriately is not obvious at all.

Instead, we decided to determine $P(R)$ directly from our data, making again use of the Bayes equation 4.1. We want to estimate the amount N_R of data records that can be associated to each road R, and then define:

$$P(R) = N_R/L_R, \tag{4.6}$$

where L_R is the length of road R. This way of estimating $P(R)$ is reasonable as long as we can assume that there are no sharp spatial discontinuities in the density of data along roads. Our datasets meet these conditions in all but a very few known cases.

The value of N_R for each road is determined recursively by using equation 4.1 on the whole subset of high-speed data. $P(R)$ is initially set to a uniform positive value for all roads. For one-way roads, $P(R)$ is forcibly set to zero for the no-way direction. At every iteration, for each high-speed record, affinity $P(R|D)$ is evaluated and used to increment N_R for all roads R with non-zero affinity with D:

$$N_R + = P(R|D),$$

where the sum of $P(R|D)$ over all the relevant roads is normalized to 1. Obviously, this means that the count for each record is divided on all affine roads in a way that is proportional to the affinity itself. Before the next iteration starts, the $P(R)$ is updated to the new estimate by 4.6.

The recursive process is run for 20 iterations and the final estimate for $P(R)$ is kept as the best estimate. The convergence of this procedure is evaluated by the evolution of ΔN_i, the normalized sum of residuals for the $i - th$ iteration with respect to the previous iteration:

$$\Delta N = \frac{1}{\mathcal{N}} \sum_R |N_{R,i} - N_{R,i-1}|,$$

where $\mathcal{N}$ is the total number of data records used, $N_{R,i}$ is the value of N_R after the $i - th$ iteration and $i > 1$. Fig. 4.2 presents a plot of ΔN_i ($\mathcal{N} = 2.7$ million high-speed data, Florence test dataset), showing a power-law trend $\propto i^{-2.4}$.

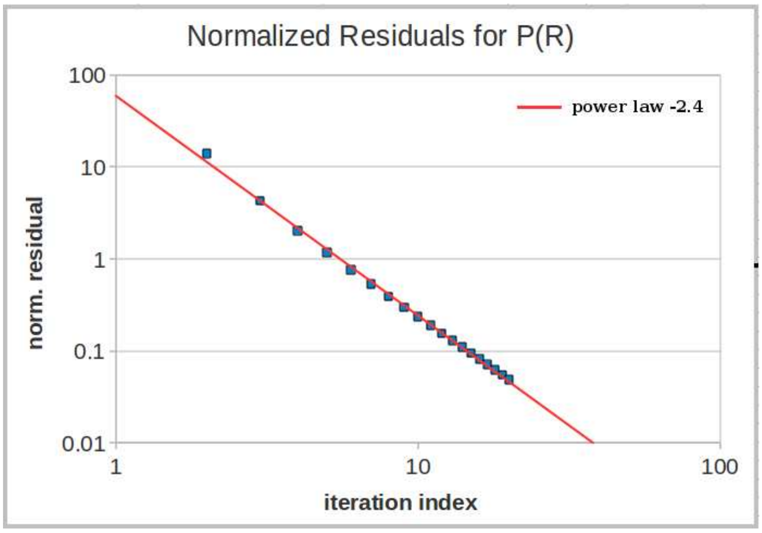

Figure 4.2: Convergence of the $i - th$ normalized sum of residuals for $P(R)$.

For consistency reasons, for roads with $N_{R,best} = 0$, far from any high-speed data, we override 4.6 and set $P(R)$ to the smallest non-null value of $N_{R,best}/L_R$. This is not true for the no-way direction of one-way roads that remains fixed at $P(R) = 0$.

4.1.4 Matching data to roads by affinity

As outlined at the beginning of this chapter, each data record is matched to the surrounding roads after having evaluated their mutual affinity. A rigorous application of this approach would require, for each datum D, to evaluate the affinity $P(R|D)$ for all the roads in the network. This strict approach is very computationally intensive and obviously unnecessary. Therefore, the computation of the affinity is limited to the road arcs closer to the datum.

To speed up the identification of the relevant arcs, a specific proximity-map is created before this phase of map-matching begins. Every cell of this proximity-map represents a square portion of the test area with a side size of 60 meters. For each arc of each polyline the minimum set of cells is found that contains the arc itself and the area surrounding it at a distance $d \leq 60$ meters. Each cell of the set will then keep memory of the arc proximity. So, during map-matching is sufficient to find the cell of the proximity-map where the datum lies and the list of the arcs that are close to that cell will be immediately available.

For each of the arcs in the list, $P(D|A)$ and then $P(D|R)$ is computed according to 4.5, approximating $P(D|A) = 0$ for all the arcs not appearing in the list. Finally, with 4.1 we compute the affinity $P(R|D)$ for all the roads with non-null $P(D|R)$. Obviously, as previously noted, this procedure implies assuming a null affinity for all the roads that are not in the immediate proximity of datum D, while, in general, those affinities are really non-zero even if negligible. For consistency reasons, then, the set of non-null affinities is renormalized to 1.

For parked data records, we do not use $P(R)$ as computed in 4.6 but we put $P(R) = 1$ to cancel any dependence from the prior road probability. This decision derives from the observation that the choice to park along a certain road is not directly correlated with the average road flux. In fact, for vehicles parked outside of roads, assuming such a correlation is a downright error. Moreover, it is also important to stress that, for all data records, $P(R)$ is always zero for the no-way direction of one-way roads.

The aim of this phase of map-matching is to identify the most probable road candidates for matching. Having more than just one match increases the effectiveness of the next phase of map-matching, but having too many candidates adds unnecessary computational costs. So, the benefit of this approach is that it allows to evaluate quantitatively which are the best match candidates. Taking advantage of this, for each datum D we keep only the road match candidates R_i that have:

$$\frac{P(R_i|D)}{P(R_{best}|D)} \geq 0.2,$$

where R_{best} is the most affine road to datum D. Applying this criterion, the existence of a strong candidate eliminates the much weaker ones, while, among candidates of similar strength, no one is eliminated.

For joint parked engine-on records (see section 4.1.1) the list of match candidates is simply copied from the corresponding engine-off record. This is necessary to be consistent with the assumption that joint engine-on and engine-off records represent the same physical position.

Data records that lie in areas with no arcs in the proximity are not matched to any road and, in the following phase of map-matching, they are simply ignored. Obviously, different approaches are possible. For example, candidate arcs for matching could be found by checking the neighboring cells of the proximity-map. However, we decided to apply this simple approach for two reasons. The first and foremost reason is the observation that this problem occurs on a very limited number of cases. Then, we also observed that the majority of these records are either low-speed data, thus probably just very inaccurately positioned, or parked data located in parking lots away from any road. So, even if a road match was found for these data, there would be no obvious way of assessing its reliability. The results presented in the following section will give quantitative support to this argument.

4.2 Results and Applications

4.2.1 Data matching for Florence dataset

The data matching procedure described in this chapter has been applied on the Florence test dataset. Here we present an overview of the results.

The following table reports how many data records belong to each of the three categories of data accuracy:

High-speed data		2'725'989
Low-speed data		303'137
Parked data		1'012'882
joint parked data	885'910	
disjoint parked data	126'972	
TOTAL data		4'042'008

We observe that the majority of data records (67%) are high-speed. Moreover, for what concerns parked data, it is apparent that joint parked records are the majority (87% of total parked records).

The matching of a data record is successful when at least a road match is found. The following table reports the numbers of successes and failures in matching for the overall dataset of 4.0 million records:

Matched data		4'023'393
Un-matched data		18'615
high-speed	5'514	
low-speed	2'784	
parked	10'317	
TOTAL data		4'042'008

We observe that the procedure succeeds on 99.5% of cases and fails only on the remaining 0.5%, corresponding to data records lying in areas with no arcs in the proximity. Considering just matching failures, we observe that 70% of them is associated with low-speed and parked data.

In section 2 we computed two error distributions for distance (Fig. 2.2) and direction (Fig. 2.3) under very strict conditions on GPS measure quality and road geometry. As described in this chapter, we based on the properties of those distributions to define $P(D|R)$ and, eventually, to compute the affinity. We have now repeated the computation of such error distributions using all successfully matched high-speed records, assuming the best match to be the correct one. Fig. 4.3 and 4.4 show the newly computed error distributions for distance and direction, respectively.

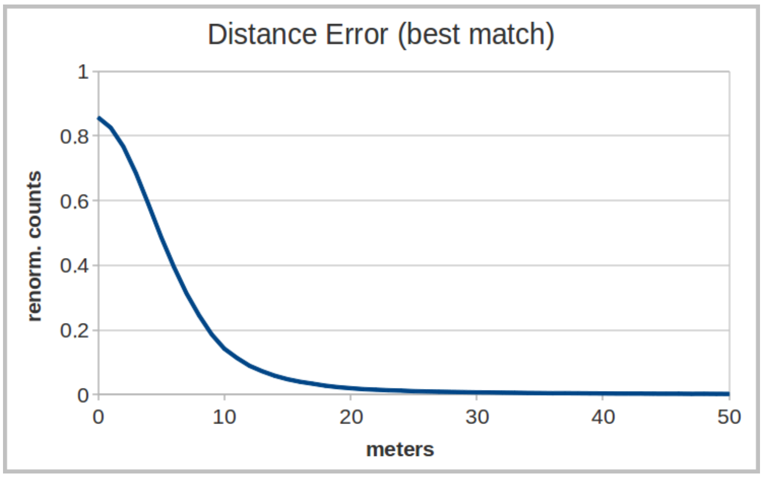

Figure 4.3: Error distribution for distance, $\sigma \simeq 9.5$ meters. The distribution has been computed using all successfully matched high-speed records, assuming the best match to be the correct one.

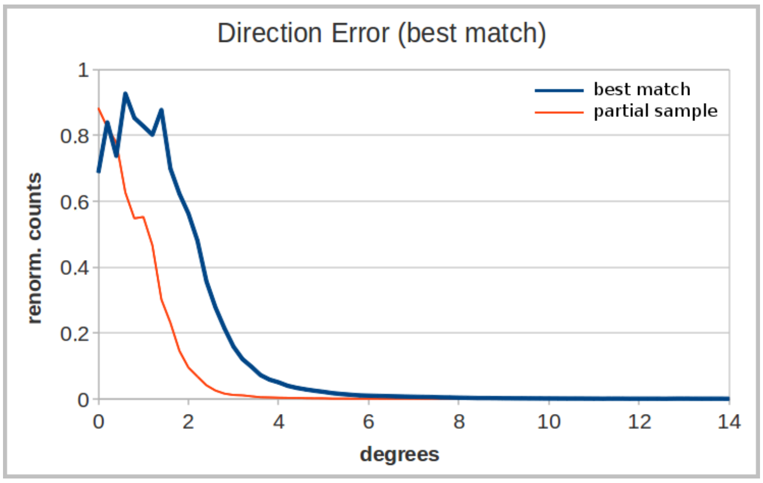

Figure 4.4: Error distribution for direction (in blue), $\sigma \simeq 2.7$ degrees. The distribution has been computed using all successfully matched high-speed records, assuming the best match to be the correct one. The error distribution relative to a partial sample (in red), $\sigma \simeq 1.2$ degrees, has been plotted for comparison.

The aim of this comparison is just to allow for a qualitative estimate of the results of the data matching procedure. Bearing this in mind, we can say that what we observe is an overall consistency among each type of distribution. In particular, both *prior* distribution (before data matching) and *posterior* distribution (after data matching) display the same slow-decaying tails.

The most apparent difference is clearly the higher error dispersion of posterior distributions. The most affected is the error distribution on direction (prior distribution has a standard deviation of ~ 1.2 degrees, while posterior distribution error has a standard deviation of ~ 2.7 degrees). In Fig. 4.4 we plotted also the prior distribution for comparison purposes. The error distributions on distance have, instead, a similar dispersion measure (prior distribution has a standard deviation of ~ 8.5 meters, while posterior distribution has a standard deviation of ~ 9.5 meters).

We remark that the higher error dispersion of posterior distributions, especially for direction, are not unexpected. In fact, prior distributions have been computed on a limited sample of data unambiguously matchable to long straight road segments, while posterior distributions have been computed using all successfully matched high-speed records, taken in a variety of conditions for measurement quality and road geometry configuration (length, bending, etc).

In particular, we observe that the error distribution for direction depends on the accuracy of angle measurements which are affected by many sources of uncertainty. In fact, the value for *ang* (see Tab. 2.1) is a derived measure that depends on vehicle speed, moreover *ang* is rounded off to the nearest even integer when is stored to memory. Further, the direction difference between road heading and vehicle heading is very sensitive to inaccurate road digitization. Given all these possible sources of errors, we consider a standard deviation for direction error of ~ 2.7 degrees, as we measure, a very reasonable one.

4.2.2 Fundamental diagram of traffic flow

As described in this chapter, the aim of the data matching phase is that of finding a list of possible road matches for each record. Each road match is associated with an affinity that measures the correctness probability of the match itself. Matching each record with its best match is thus a sensible approximation.

This is exactly what we did with our data. For many roads in the network the number of data matched with this criterion is enough to compute a robust estimate of the daily evolution of vehicle flux (ϕ), speed (v) and density (ρ).

For each matched data record we have *vel*, *ds* and R, the best road match. For each road R we can then estimate $\phi_R(t)$, $v_R(t)$ and $\rho_R(t)$ as follows:

$$\phi_R(t) = \frac{1}{T} \cdot \sum_i^N \frac{1}{(L/ds_i)} = \frac{N}{T} \cdot \frac{\langle ds \rangle}{L},$$

$$v_R(t) = \frac{1}{N} \cdot \sum_i^N vel_i = \langle vel \rangle,$$

$$\rho_R(t) = \frac{1}{L} \cdot \sum_i^N \frac{(L/vel_i)}{T} \cdot \frac{1}{(L/ds_i)} = \frac{N}{L} \cdot \frac{\langle ds/vel \rangle}{T},$$

where L is the length of road R, T is the time resolution and N is the total number of records matched to R that were measured between t and $t+T$.

We observe that for the definitions of $\phi_R(t)$ and $v_R(t)$ only one between the derived measures of *vel* and *ds* are used, while the definition of $\rho_R(t)$ requires the use of the ratio ds/vel. Given the measure uncertainties for both *ds* and *vel*, using their ratio decreases the accuracy of the estimate for ρ, in comparison with the estimates of the other two quantities. For this reason we also estimated $\rho'_R(t) = \phi_R(t)/v_R(t)$, deriving it from the other two quantities so to have an estimate less affected by fluctuations. However, we observed systematically that the two estimates for density do not manifest substantial divergences, so we decided to use former, the independent density estimate $\rho_R(t)$.

In Fig. 4.5, 4.6 and 4.7 we show the daily profiles of $\phi_R(t)$, $v_R(t)$ and $\rho_R(t)$ respectively, computed for a road in Rome with high traffic intensity. In order to have a robust sample, average values have been computed on the ensemble of all the 21 workdays of the month of May 2010. The road chosen is a portion of the southern stretch of the *Grande Raccordo Anulare*, the ring-shaped highway encircling Rome. The total number of data matches, counted

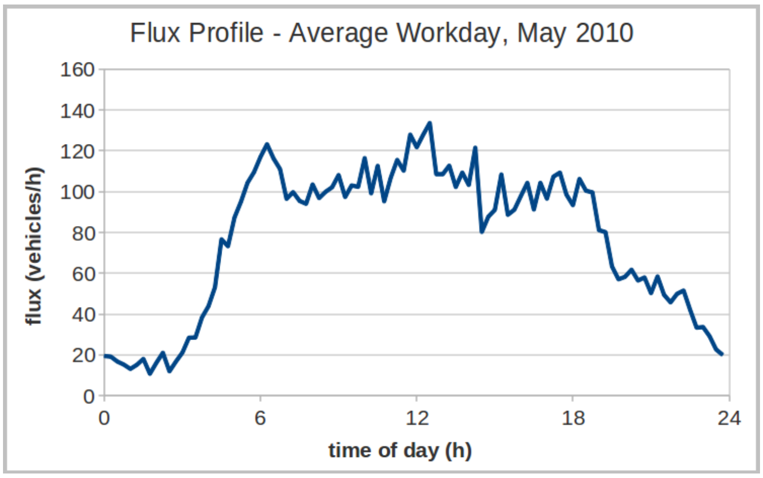

Figure 4.5: Daily profile of $\phi_R(t)$. Profile computed from data matches for the average workday of May 2010.

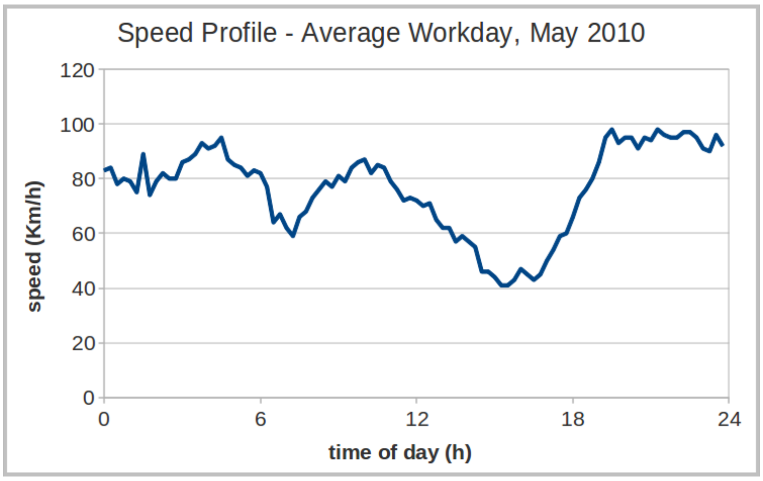

Figure 4.6: Daily profile of $v_R(t)$. Profile computed from data matches for the average workday of May 2010.

on the entire period of 21 workdays, is 56'335, but the profiles have been rescaled on the daily count average. Time resolution T is 15 minutes.

From these profiles it is easy to reconstruct the daily traffic trend that is characteristic of that road: during the night hours the road is clearly less populated, so speed is high, flux and density are low; in the morning, around 6am, the road begins to fill up, so density increases, average speed decreases and flux reaches a regime level; during the central hours of the day, the road is less populated, so that speed can rise back to optimal values and the regime flux is maintained; traffic conditions worsen during afternoon rush hour, around 5pm, when the density is at is peak, speed drops to very low values so that signs of congestion appear on the flux profile too.

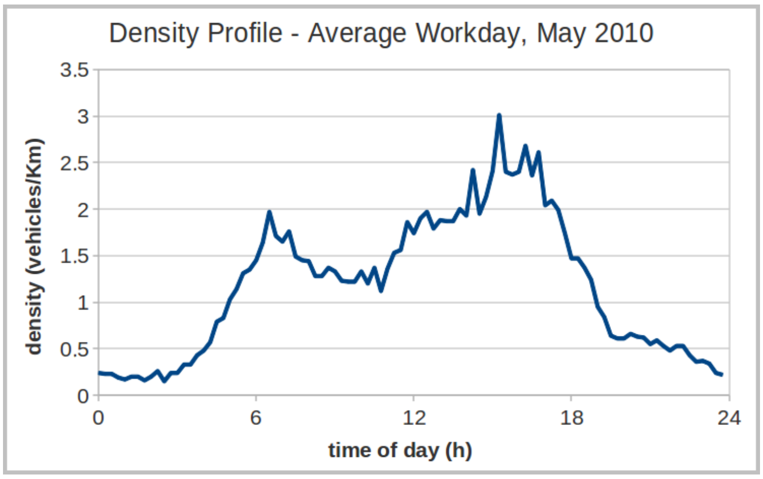

Figure 4.7: Daily profile of $\rho_R(t)$. Profile computed from data matches for the average workday of May 2010.

By comparing the evolution of these three quantities we get what is called the *fundamental diagram of traffic flow* [25]. As an example, Fig. 4.8 shows ϕ_R vs. ρ_R, where every data-point corresponds to a bin of the average daily profiles.

This plot is very interesting because it allows to separate two different flow regimes of the road. The initial part of the diagram represents a free-flow regime, where an increase in vehicle density linearly corresponds to an increase in flux. The last part of the diagram, instead, represents flow regime affected by different degrees of congestion. When the vehicle density crosses a certain critic threshold characteristic of the road, vehicles can no longer proceed at free-flow speed. This causes a decrease in flux that is no longer linearly dependent on density.

It is apparent, then, that these profiles are very important. Thanks to them it is possible to characterize the flow properties of each road. For example, the effective free flow speed of a road can be computed from its $v_R(t)$ profile. Moreover, from the fundamental diagram it is possible to extrapolate the regime value for the flow, as well as the critical density value that triggers congestion.

As already noted, however, the profiles we describe here are averaged on several days. An accurate service for traffic monitoring and now-casting requires, instead, robust real-time estimates of those quantities. As we will see in the next chapter, vehicle path reconstruction brings us closer these requirements.

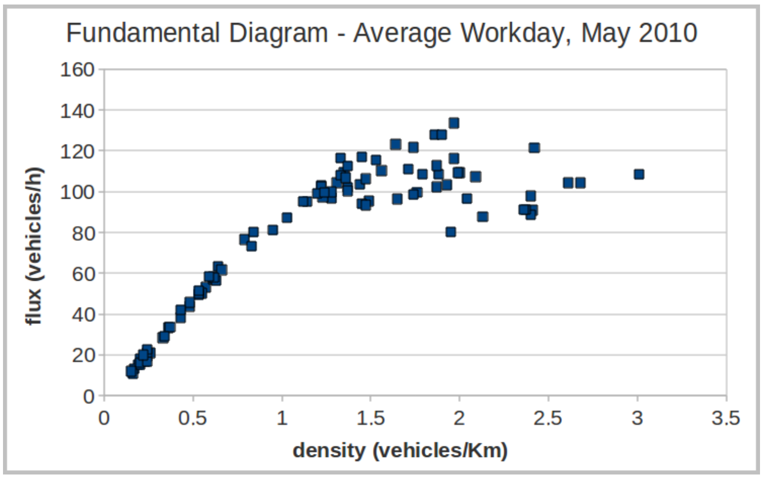

Figure 4.8: Fundamental diagram of traffic flow: v_R vs. ϕ_R. Diagram computed from data matches for the average workday of May 2010.

Chapter 5

Global Path Matching

The aim of this phase of map-matching is to match each GPS trajectory to a path on the road network.

The approach we take is based on two principles: We assume that the most probable path match is the one requiring the shortest travel-time. We verify that the length of the chosen path match and its estimated duration are consistent with the measured values and we otherwise discard the match.

These assumptions are simple and clear-cut and they prove to be very robust, as we will show in this chapter. Other works in the field of low-sampling map matching use more elaborated constraints on the length L of the path match [19], on its estimated duration T [18] or on both [17].

However, we justify our choice on the basis of two observations on length L and estimated duration T. Given two subsequent data records, located in an urban context at an average spatial separation ($\sim$ 2Km, in our case), there are typically many different paths of similar length L that connects the records along the road network. Given no other constraint, we can hardly evaluate which is the most appropriate among them. The best we can do, in our opinion, is then just to chose a reasonable path using a simple criterion. This is what we do by looking for the path with the shortest travel-time. Moreover, for what concerns the constraints on the estimated duration T of a path match, we point out that uncertainties are even greater. In order to compute an accurate estimate of T, a precise knowledge of traffic conditions along the path would be necessary, along with assumptions on the time cost of turns and stops at intersections, as [19] also points out. For this reason, in our path matching procedure we do estimate a free-flow duration T for each path match, but we just use it as a lower bound to check the consistency of the match.

We point out that all the path matches that are found to be not consistent with measured values are discarded. Obviously, this means also that all the path matches that are kept, which are the wide majority, are to be considered very reliable. This map-matching procedure has been developed to study the global properties of traffic flows. Thus, from this perspective, being able to count on robust path matches is much more important then matching every single trajectory regardless of match accuracy or reliability.

The global path match procedure is divided into a sequence of four steps. First, for each couple of subsequent records in a trajectory we use an optimized A* path finding algorithm to connect the data matches of the first record to the data matches of the second record. This step creates different alternatives for the global path match of the whole trajectory. The second step identifies the best global path match as the path match that requires the least total average travel-time. Then, we check the consistency of the different parts of the best global path match and we discard the inconsistent ones. Finally, we compute the road transits (see Tab. 1.1) along the whole global path match.

5.1 Path Matching Procedure

5.1.1 A* algorithm for path finding

The A* algorithm is a path finding algorithm that identifies the least-cost path between any two nodes in a weighted network [20], [21]. In a weighted network, each link is associated to a value representing its weight (or cost). The least-cost path between two nodes is the one that connects them with links having the least cumulative cost. In this context, we refer to the leas-cost path as the best path.

The principal merit of this algorithm is its efficiency. It reaches its goal by limiting the search area in a clever way. This behavior is possible because the

A* algorithm uses an heuristic cost function to establish the order in which the search visits the nodes in the network. The heuristic cost function K at a generic node n is a sum of two components:

$$K(n) = C(n) + H(n),$$

where $C(n)$ is the cost of the best path connecting the origin node to node n and $H(n)$ is an heuristic estimate of the cost necessary to reach the destination node. In order for the algorithm to work efficiently, $H(n)$ must not overestimate the real cost of the best path from node n to the destination node. If $H(n)$ overestimates the real cost, the path found by this algorithm is not guaranteed to be the least-cost one. However, if the overestimation is contained, the path found is still one among the least-cost ones. For example, if the cost of a path is given by its length, then a robust heuristic estimate $H(n)$ would be the euclidean distance between node n and the destination node.

More in details, the algorithm works in the following way:

- The search begins from the origin node n_o and stops when the destination node n_d is reached. Obviously, at the origin node $C(n_o) = 0$ holds.
- Let n be the currently visited node. For all the nodes n' directly connected to n we evaluate:

  $$K(n') = C(n') + H(n') = C(n) + C_{n,n'} + H(n'),$$

  where $C_{n,n'}$ is the cost of the link connecting n to n'.
- Then, nodes n' and the respective values for $K(n')$ are memorized in a list, together with all the other nodes n' evaluated in the previous iterations but not yet visited. This list is ordered with respect to K values and the node n*, corresponding to the smallest total heuristic cost $K(n*)$ is picked out of the list and becomes the new visited node.
- Since every node n' in the list has memory of the node n it was evaluated from, when the destination node n_d is finally visited, the best path can be reconstructed by simply tracing the best links backwards.
- Obviously, at the destination node $H(n_d) = 0$ holds and the heuristic cost function K coincides with the cost of the best path: $K(n_d) = C(n_d)$.

As said previously, the effect of the heuristic estimate is that of aiming node exploration towards the direction of the destination node right from the beginning, instead of wasting time with a uniform radial exploration. In Fig. 5.1 we see an example of this effect. From the perspective of computational costs, the choice of an heuristic estimate that does not overestimate the real cost guarantees that the algorithm does not visit the same nodes more than once. On the other hand, if the underestimation is too strong, then part of the directional advantage of this algorithm is lost and the total number of visited nodes increases unnecessarily.

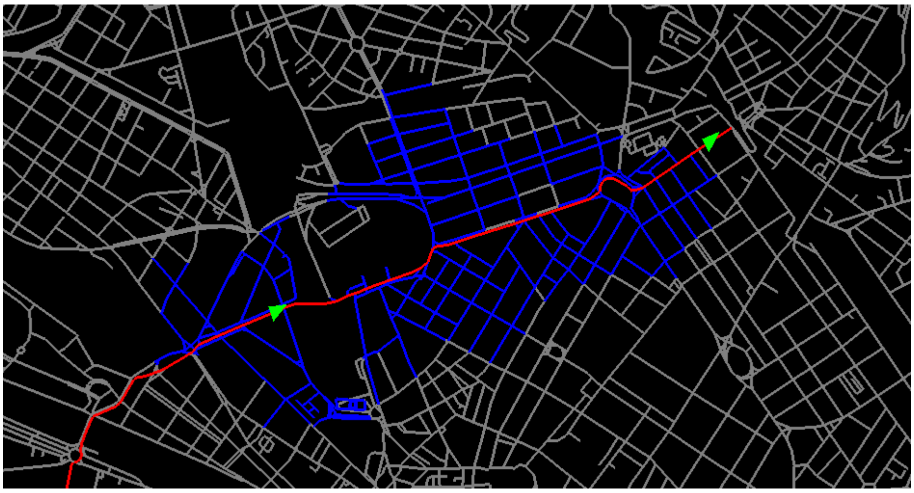

Figure 5.1: Example of path finding with the A* algorithm. Green triangles represent two data records to be connected by the best path (origin record on the left, destination record on the right), blue roads are the roads explored by the algorithm during the best path search, red path is the best path. It is apparent how the algorithm pushes the exploration of the network in the direction of the destination record.

5.1.2 Path finding between records

In order to find the possible paths connecting subsequent records in a trajectory, we developed a specific modified version of the A* algorithm. Our goal is to find the best paths connecting the data matches of the first record (the *origin*) to the data matches of the second record (the *destination*). More precisely, for each of the data matches of the origin, we want to identify the best path among all the possible paths connecting it to all the data matches of the destination.

Thus, the main differences with respect to the original algorithm are the following:

- Paths do not begin and end at road nodes. They begin and end at the location of data projections onto the matched roads (see section 4.1.2).
- So, for each data match, both nodes of the matched road are considered as origin nodes (for the origin record) or destination nodes (for the destination record).
- The driving direction of the data match is taken into consideration by increasing the cost of the road node corresponding to a U-turn.
- For each destination data match, all origin data matches are considered together as possible path origins during a single algorithm run.

Obviously, controls are put in place to ensure that the procedure employs the computational time efficiently. Given the known time interval between the two data records, for each path explored during the best path search it is possible to compute the corresponding average travel speed. If the best path search is still on-going but the length of the current best path candidate grows so high that the average travel speed becomes bigger than 200 Km/h, than the search is aborted. We interpret this fact by supposing that the corresponding destination data match is a wrong match and we delete it.

As presented in the introduction to this chapter, we assume that the most probable path match is the one requiring the shortest travel-time. Coherently to that statement, we chose a cost function for best path finding that is proportional to the free-flow travel-time:

$$C_{n,n'} = \alpha \cdot \frac{L_{n,n'}}{V_{n,n'}},$$

where $C_{n,n'}$ is the cost of the road $R_{n,n'}$ joining node n to node n', $L_{n,n'}$ is the length of the same road, $V_{n,n'}$ is its free-flow speed and $\alpha = 30$ is just a normalization factor. As we described in section 4.2.2, free-flow speed can be computed from daily average speed profiles derived from data. For all the roads where this parameter could not be determined from data, we used the average speed associated to each polyline specification (Tab. 2.2).

If $R_{n,n'}$ corresponds to the no-way direction of one-way road, $C_{n,n'}$ is set to a very high value, so that the best path is prevented to go through this road. If $R_{n,n'}$ corresponds to a limited-access road (Road Access = 3, see Tab. 2.2), then the value $C_{n,n'}$ is set to depends on the vehicle. If the vehicle has access to this special category of roads, then $C_{n,n'}$ is computed in the default way. If the opposite holds, $C_{n,n'}$ is set to a very high value as in no-way roads. In order to establish if a vehicle has access to limited-access areas we perform a very simple check. If more than 1% of the best data matches of the vehicle fall on limited-access roads then we assume the vehicle has access to them.

Consistently with the cost function, we define the heuristic cost estimate as:

$$H(n) = \alpha \cdot \frac{L_n^{eucl}}{V_{max}},$$

where $\alpha = 30$ as above, L_n^{eucl} is the euclidean distance between node n and the destination data match and V_{max} needs to be set to an appropriate value. Choosing a reasonably high value for V_{max} ($\sim$ 120 Km/h) we are guaranteed that $H(n)$ is robust so that the algorithm does not visit the same nodes more than once. However, this choice of V_{max} corresponds to a rescaling of L_n^{eucl} by a factor of ~ 0.25, which is on average a very strong underestimation of the minimum cost to the destination and is therefore not the most efficient choice at all. Experimentally we determined that with a rescaling factor of ~ 0.8 the trade-off between node revisitation and total visited nodes is globally the most efficient. We are aware that with this choice we are on same occasions overestimating $H(n)$. However, we think that the choice is reasonable because in these conditions the overestimation is contained and the path finding algorithm still chooses a path among the least-cost ones. The results on path length consistency presented in section 5.2.1 are in support of this choice.

The outcomes of this procedure are as many path matches as the number of destination data matches. To each path we associate its length L, free-flow travel-time T, final cost K and the list R_i of roads that compose it. Length L is given by the sum of road lengths L_i. Symmetrically, free-flow travel-time T is given by the sum of road free-flow travel-times T_i, given by $T_i = L_i/V_i$, where V_i is the road free-flow speed described above.

It can happen that the path finding algorithm is aborted for all the destination data matches. We interpret this fact by supposing that one of the two data records has a very high positioning error. It could possibly be an outlier due to GPS signal reflections on buildings. Another possible cause that we observed is the absence from the road network database of the road the vehicle is driving onto. This can happen for newly built roads. In any case, we consider the original trajectory interrupted at this point and we perform the global path matching procedure independently on the two parts.

5.1.3 Global best path matching

As declared in the introduction of this chapter, the aim of this phase of map-matching is to match each GPS trajectory to a path on the road network.

At this point of the map-matching procedure, the sequence of GPS data for each vehicle is grouped into trajectories and each data record has a list of possible road matches. For each couple of subsequent data records in a trajectory we apply the path finding procedure described in section 5.1.2. This procedure finds for every data match of the second record the path with shortest travel-time connecting it to the data matches of the first record. So, considering the sequence of records in a trajectory, we now have different alternatives for the global path match of the whole trajectory (Fig. 5.2). Each alternative global path is associated with its travel-time cost, which is the sum of the travel-time costs associated with the parts that compose it. Coherently with the previous step, we chose as global best path match the global path with lowest travel-time cost.

As apparent, in the process of choosing the global best path match, no weight is given to data match affinity. The reason behind this choice is the decision to give more relevance to the correct positioning of an entire path and prevent the estimates on positioning uncertainties of just one point (the data match) to have a radical influence. Thus, we give equal weight to all matches. Nonetheless, the existence of more than one match per data is important because it allows for the evaluation of alternatives for the global path match.

For subsequent regular trajectories (see section 3.1.1) linked by joint parked data records (see section 4.1.1), we proceed in a slightly different way. All the steps described above are applied to the whole group of subsequent regular trajectories. Obviously, for what concerns path finding, subsequent joint engine-off and engine-on records are considered as just one record with a common set of road matches. Thus, the global best path match describes the entire joint sequence of trajectories. This is consistent with the assumption that joint engine-on and engine-off records represent the same physical position.

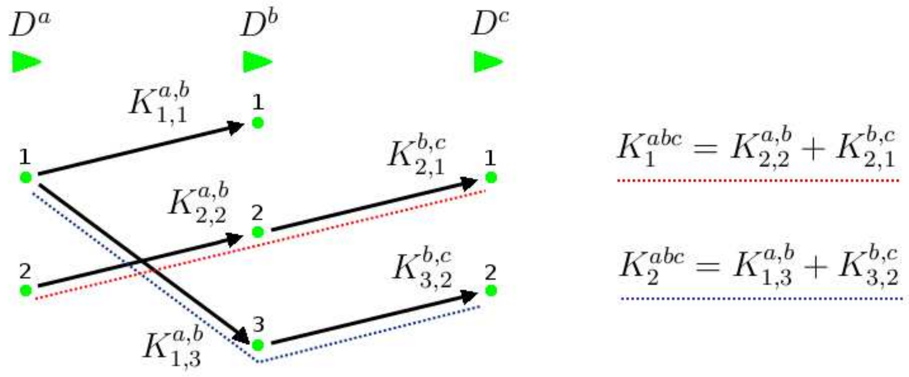

Figure 5.2: Alternative global path matches. Figure shows a trajectory with three records (green triangles): D^a, D^b, D^c; Each record has several road matches (green dots), labeled with a progressive number; Each destination road match j is connected with an origin road match i and each connecting path (black line) is associated with its travel-time cost $K_{i,j}$; Globally, the trajectory has two alternative global path matches (red and blue dotted lines), each associated with its own global travel-time cost: K_1 and K_2.

5.1.4 Path consistency check

The next step of this phase is to check the consistency of the global path match. To perform this check, the different parts of the global best path match are considered separately. For each couple of subsequent records we know: the length L of the part of the global path match that connects them, the time interval dt between the records and the length ds of the second record that estimates the length of the real path driven by the vehicle between the two records.

To begin with, we compute the residual $\Delta = |L - ds|/ds$ between matched path length and measured path length. If the residual Δ is greater than 10%, then the relative part of the global path match is considered inconsistent and is discarded. We interpret this fact by assuming that the real path taken by the vehicle is radically different from the shortest travel-time path.

We observe that, if this is the case, it is difficult to define a robust criterion to isolate which other path the vehicle has taken, in an urban context. If the vehicle did not follow one of the paths with the shortest travel-time, how can we reasonably identify which of the many others it did take? We could look for paths whose lengths are the closest to ds, but these are many in general and far apart from each other and we do not have any reliable criterion to chose one among the others. A possible way of making this choice less arbitrarily is to refer to the past data for the vehicle. In fact, the paths that the vehicle repeats in time with high frequency can be reasonably considered more probable than others in general. Thanks to the availability of past vehicle data that we have, we tested this approach and we obtained promising results. However, for now, this technique is still under development and has therefore not been employed in the process described here.

To continue, if the length consistency check is passed we then evaluate $V = L/dt$. If this average speed is higher than 200 Km/h, then the relative part of the global path match is considered inconsistent and is discarded.

We note that such a violation corresponds to a measured travel time dt that is much smaller than the free-flow travel-time T computed for the matched path, which we consider a lower bound. We interpret this fact by assuming that one of the data records delimiting the path are affected by a very high positioning error. Again, the record could possibly be an outlier due to GPS signal reflections on buildings.

The results presented in section 5.2.1 will show that discarded paths from these filters are comparatively few. This fact indirectly supports the choice of building the global path match phase on the simple assumption of least travel-time.

5.1.5 Road transits computation

Finally, it is now possible to compute the road transits (see Tab. 1.1) along the whole global best path match. Again, the different parts of the global best path match are considered separately. Obviously, the paths discarded because of inconsistency are not considered. For each couple of subsequent records we know: the list of road R_i that compose the path that connects them, the path length L and the time interval dt between the records.

Road transits are computed according to those quantities and basing on the assumption that the path is taken driving at constant speed $V = L/dt$. So, transit time dt_i for each road R_i is:

$$dt_i = \frac{L_i}{V} = dt \cdot \frac{L_i}{L},$$

where L_i is the length of road R_i.

We observe that during the step for path consistency check (section 5.1.4) we did not discard paths whose average travel speed V is very low. In fact, we interpret paths with consistent length L and small speed V as correct path matches, driven under congested traffic conditions. Thus, after this interpretation, these paths represent a very important piece of information on traffic conditions and they are obviously not to be discarded. However, because we approximate that the path is driven at constant average speed, the effects of congestion are diluted along the whole path. We think that this consequence is acceptable because it does consistently approximate the real propagation of congestion along neighboring roads. However, as we will see in the following section, occasions of matched paths with very low speed are comparatively not frequent.

5.2 Results and Applications

5.2.1 Path matching for Florence dataset

The global path matching procedure described in this chapter has been applied on the Florence test dataset. Here we present an overview of the results.

As detailed in this chapter, path matching procedure attempts to find the path on the road network that connects subsequent data records in a GPS trajectory. Thus, obviously, this procedure is not applied to connect subsequent data that belong to different trajectories (except the case of joint parked records, see section 4.1.1). Moreover, for data records where data match failed, path matching procedure could not be run either.

The following table reports the results for the whole dataset, showing the number of successful path matches, the number of failures and the number of cases where this phase could not be applied:

Successful path matches		2'982'391
standard path match	2'543'399	
joint parking	438'992	
Failed path matches		836'617
inconsistent length	809'327	
inconsistent speed	27'290	
No attempt: data match error		31'445
no matches for path origin	12'830	
no matches for path destination	18'615	
No attempt: trajectory interruption		191'555
irregular interruption	144'527	
disjoint parking	47'028	
TOTAL data		4'042'008

We observe that the path matching procedure was successful for 78% of the cases where it was applied, while it failed for the remaining 22% of the cases.

As described in section 5.1.4, if the length L of a path match is not consistent within 10% with the measured length ds, the path is discarded. As reported in the results table, the number of path matches discarded because of this length consistency criterion amount to 21% of the cases where the procedure was applied. The distribution shown in Fig. 5.3 allows to evaluate qualitatively the number of data records filtered with this criterion. The figure shows the distribution of the ratio L/ds for all 3.8 million matched paths. The area delimited by the two vertical lines represents the number of paths accepted by the length consistency criterion.

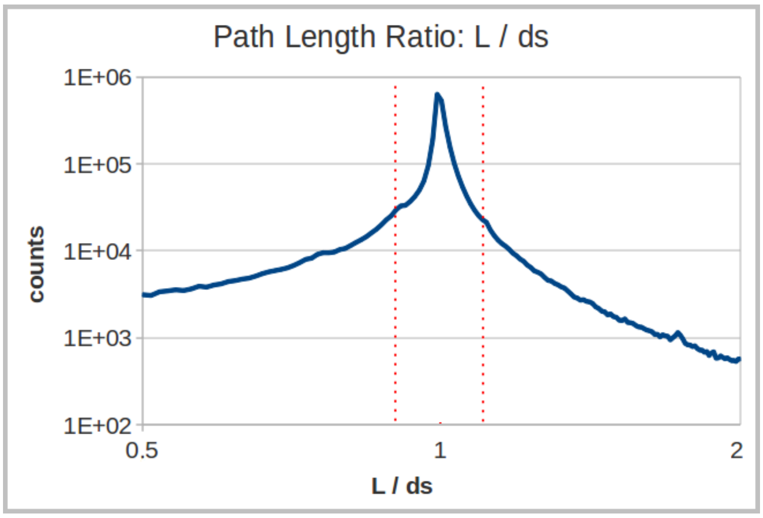

Figure 5.3: Distribution of the ratio L/ds used for the path length consistency criterion. The area between the two vertical dotted lines represents the number of paths accepted by the criterion.

We report that the percentage of paths showing signs of strong congestion (average path speed $V < 3$ Km/h) amounts to 2% of the total successful path matches (see section 5.1.5 for more details).

Moreover, we report that the percentage of best data matches chosen as part of global best paths amounts to 80% of the total chosen data matches. This fact gives indirect support to the overall consistency of our map-matching approach.

5.2.2 Towards real-time traffic monitoring

In this chapter we described how road transits are computed. For each couple of subsequent records in a trajectory we identify the best path that connects them. This path is composed by a sequence of roads and, for each of these roads, we compute a road transit. Thus, path matching allows to have a greater quantity of information on the dynamical state of roads than what can be achieved by using data matches alone (see section 4.2.2). In particular, for many roads of the network the number of road transits available is enough to compute a robust estimate of the daily evolution of vehicle flux (ϕ), speed (v) and density (ρ), without averaging on more days.

Foe each road transit we have the transit duration dt_i and the end time τ_i (see Tab. 1.1). For each road R we can then estimate $\phi_R(t)$, $v_R(t)$ and $\rho_R(t)$ as follows:

$$\phi_R(t) = \frac{N}{T},$$

$$v_R(t) = \frac{1}{N} \cdot \sum_i^N \frac{L}{dt_i} = L \cdot \langle 1/dt \rangle,$$

$$\rho_R(t) = \frac{1}{L} \cdot \sum_i^N \frac{1}{(T/dt_i)} = \frac{N}{L} \cdot \frac{\langle dt \rangle}{T},$$

where L is the length of road R, T is the time resolution and N is the total number of road transits on road R whose end time τ_i falls between t and $t+T$.

In Fig. 5.4, 5.5 and 5.6 we show the daily profiles of $\phi_R(t)$, $v_R(t)$ and $\rho_R(t)$ respectively, computed in the way described above for the same road in Rome described in section 4.2.2. In this case, the profiles derive from 2'322 road transits counted on May 5, 2010. Time resolution T is 15 minutes.

We compare these figures with Fig. 4.5, 4.6 and 4.7 respectively, showed in section 4.2.2. We observe that the profiles computed in this section are in agreement with the profiles computed previously, even if the new profiles are slightly more noisy. This is obviously due to the fact that the profiles from data matches are averaged over 21 days, while the profiles shown here are computed with data relative to a single day. However, we can not exclude that part of the fluctuations in in Fig. 5.4, 5.5 and 5.6 describe real hourly variations of the computed quantities.

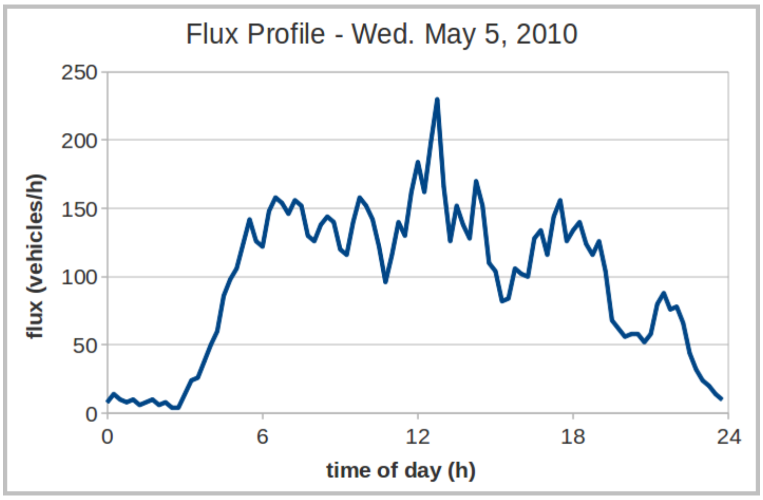

Figure 5.4: Daily profile of $\phi_R(t)$. Profile computed from road transits for Wednesday May 5, 2010.

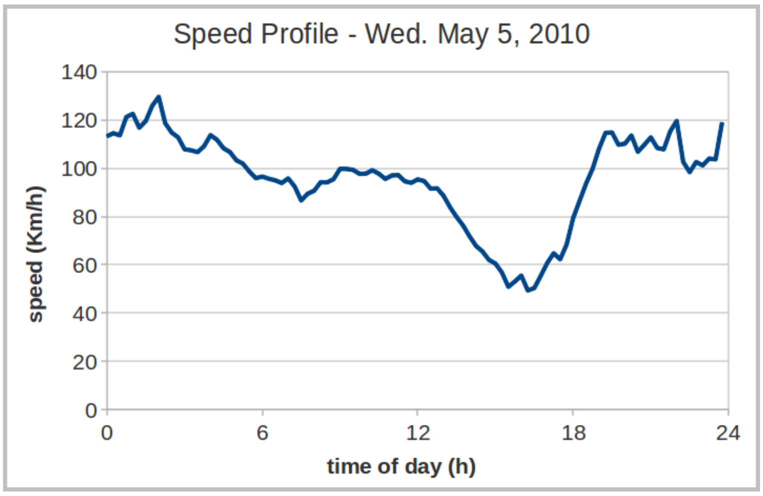

Figure 5.5: Daily profile of $v_R(t)$. Profile computed from road transits for Wednesday May 5, 2010.

Again, the evaluation of the fundamental diagram yields interesting and detailed information on traffic dynamics [25]. Fig. 5.7 shows ϕ_R vs. ρ_R for a different road. Here the phases of congestion formation and dissolution are very clearly visible. The image shows two curves with a similar trend. Each curve describes a different rush-hour event for the road. We observe the temporal evolution of the curves. Before rush-hour begins, the road is in the free-flow regime. Then, as the vehicle density increases, the road enters the congested regime, where flux stops increasing linearly with density. When rush-hour is over, vehicle density decreases and the road dynamical state goes back towards free-flow conditions following a different path in the fundamental diagram. The overall evolution forms a closed loop that defines an hysteresis cycle.

Being able to compute $\phi_R(t)$, $v_R(t)$ and $\rho_R(t)$ as they change along the day is a very important result. From the perspective of the study of the dynamic system of vehicular traffic, the availability of accurate profiles for those quantities is very useful to develop, test and compare accurate models of vehicle dynamics. But this result is also very important in the context of the development of a real-time traffic monitoring system. For example, as an immediate application, real-time computation of these quantities allows to warn in advance when critical conditions are arising. Moreover, the availability of this detailed information for virtually all the roads in the network, provides the basis to study how critic traffic conditions propagate along the network. And acquiring knowledge on this phenomenon is fundamental in order to define reliable models for traffic now-casting.

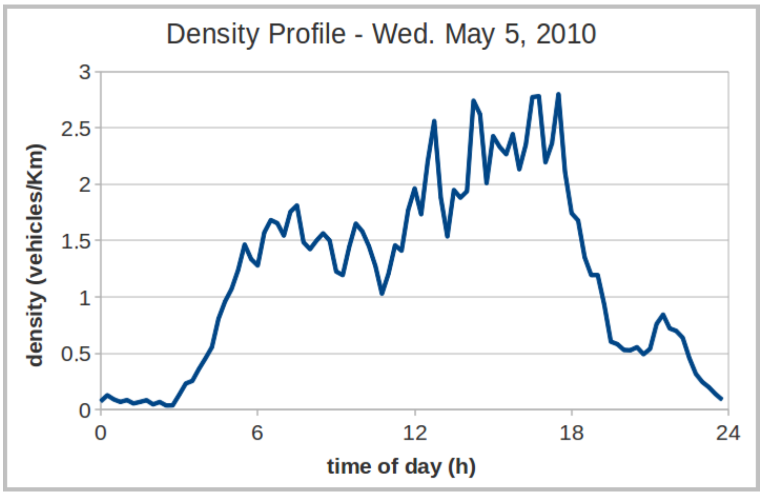

Figure 5.6: Daily profile of $\rho_R(t)$. Profile computed from road transits for Wednesday May 5, 2010.

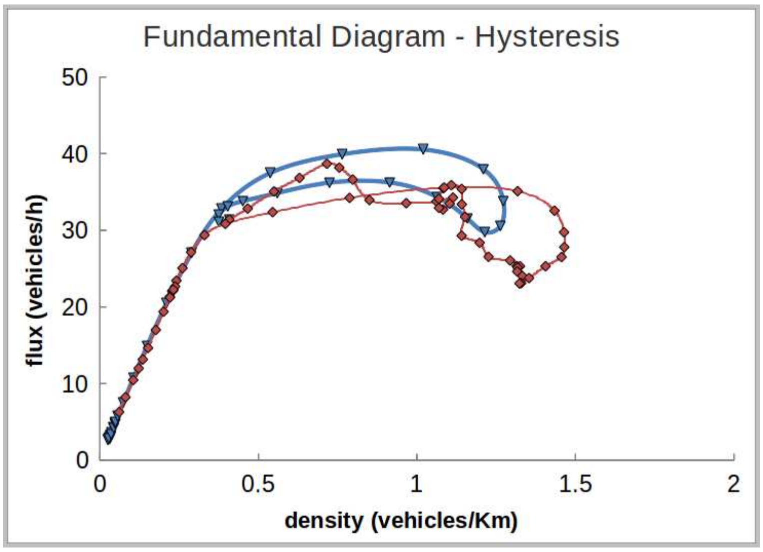

Figure 5.7: Fundamental Diagram showing hysteresis cycles for two rush-hour events.

Conclusions

In this thesis we described a novel procedure for the map-matching of low-sampling GPS data from vehicles. The procedure is based on simple assumptions and gives great importance to check the consistency of the results of each of its phases.

The performance of our approach has been showcased on an extended off-line path reconstruction task on the metropolitan area of Florence, Italy. For each phase of the map-matching process we presented a detailed analysis of the results and discussed validation plots. Globally, the algorithm proved to be robust and accurate.

In addition to the description of the characteristics and functionalities of the algorithm, we also showed its relevance as a tool for the study of traffic dynamics. At the end of the respective chapters, we presented the research opportunities that each of the three main phases of the map-matching process made available. In particular, we remark the importance of the result obtained by computing the daily profiles for road flux, density and average speed for the whole urban road network of the test area.

Fig. 5.8 shows an example of the amount of information on traffic state that this result makes available to us. The figure shows the color-coded speed map for the urban road network of Florence at a certain moment in time. Speed values are taken from the profiles of this quantity computed for each road. In an analogous way, we can compute flux maps and density maps for all the road in the network where we have enough data.

This type of measures, and its richness, will allow for a detailed study of the properties of traffic flow. Of particular importance will be to study how

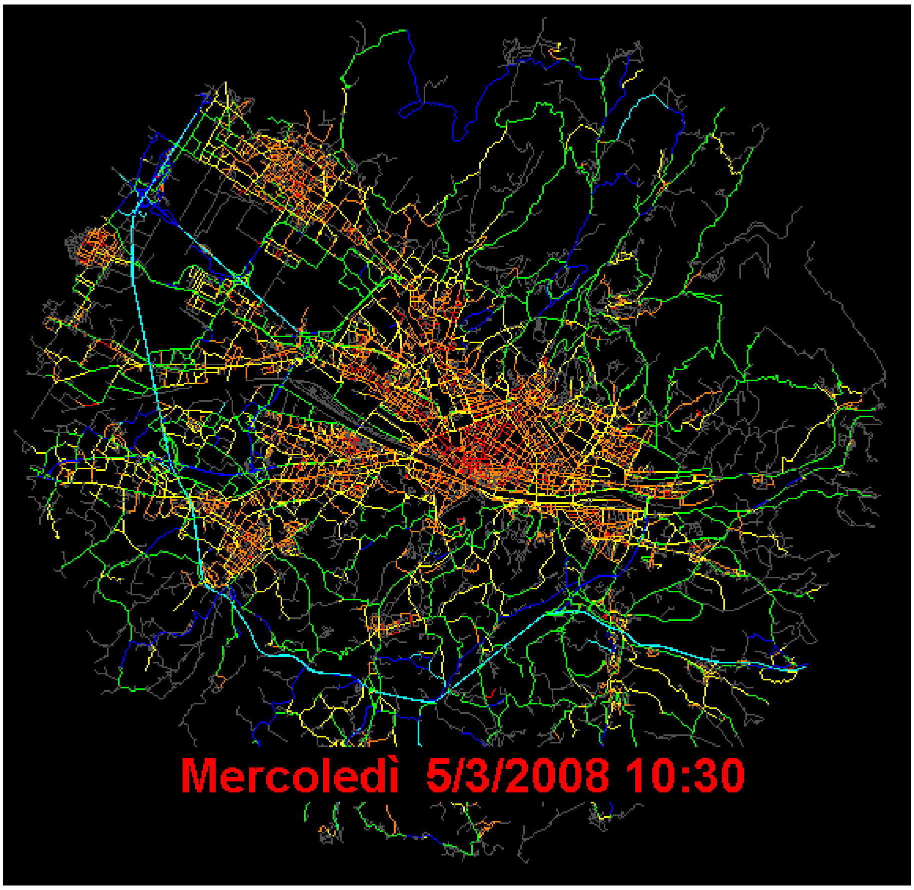

Figure 5.8: Speed map for the urban road network of Florence, computed for 10.30am on Wednesday March 5, 2008. Colors code speed in Km/h: $0 < Red < 10 < Orange < 20 < Yellow < 30 < Green < 50 < Blue < 80 < Cyan$.

instabilities arise from steady conditions and how they propagate along the road network.

For what concerns the future developments to the map-matching procedure, the algorithms that compose it are under constant evolution and improvement. In particular, we plan to model more accurately the shape of the error distribution for GPS data positioning. In this way, we expect to improve the ability of the measure of affinity between data and roads to represent the quality of a match. Moreover, we are testing different ways of taking into account past data for the vehicles in the phase of global path finding. In fact, we expect that the information on travel habits, specific for every vehicle, available to us from these past data can improve the efficiency of consistent path identification. Then, we are working on an overall improvement in computation efficiency and optimization of the various algorithms.

All these improvements are important in the framework of the expected use of this map-matching procedure in the immediate future. Specifically, we plan to adapt the procedure to real-time elaboration and to extend the working area to the entire Italian road network. In fact, the final aim is the set up of an infrastructure for nation-wide real-time traffic monitoring. The implementation of this service is part of the contributions of the Physics of the City Laboratory to the Pegasus project [22] on intelligent traffic management, financed by the "Industria 2015" action of the Italian Government [23].

The development of such a service will be an interesting result on its own. However, the plans are to further evolve the monitoring infrastructure into a traffic now-casting system. This will be possible by integrating the information of real-time traffic conditions with short-term predictive models for vehicle dynamics.

In conclusion, as nowadays we are used to weather forecasts and to its impact on our daily lives, in the near future, we can reasonably expect to be able to rely also on a similar system for traffic forecast, even if short-term, with all the improvements to the quality of our lives that this implies.

References

[1] *Octo Telematics S.p.A.: company providing telematics services and systems for the insurance and automotive market*, http://www.octotelematics.it/.

[2] R. Gallotti (2009), *Mobilitá Urbana: Individui razionali su una rete complessa* (in Italian), MSc Thesis, University of Bologna.

[3] A. Bazzani et al. (2010), *Statistical laws in urban mobility from microscopic GPS data in the area of Florence*, Journal of Statistical Mechanics, Volume 2010, Issue 05.

[4] D. Obradovic et al. (2006), *Fusion of map and sensor data in a modern car navigation system*, Journal of VSLI Signal Processing 45, pp. 112122.

[5] D. Bernstein and A. Kornhauser (1996), *An introduction to map matching for personal navigation assistants*, New Jersey TIDE Center, http://www.njtude.org/reports/mapmatching.pdf.

[6] M. A. Quddus et al. (2007), *Current map-matching algorithms for transport applications: State-of-the art and future research directions*, Transportation Research Part C: Emerging Technologies, Volume 15, Issue 5, pp. 312-328.

[7] J. Greenfeld (2002), *Matching GPS observations to locations on a digital map*, Proc. of 81th Annual Meeting of the Transportation Research Board, Washington, DC.

[8] M. A. Quddus et al. (2003), *A general map matching algorithm for transport telematics applications*, GPS Solutions, Volume 7, Number 3, pp. 157-167.

[9] H. Yin and O. Wolfson (2004), *A Weight-based Map Matching Method in Moving Objects Databases*, Proc. of 16th International Conference on Scientific and Statistical Database Management (SSDBM'04), pp. 437.

[10] W. Y. Ochieng et al. (2004), *Map-matching in complex urban road networks*, Brazilian Journal of Cartography (Revista Brasileira de Cartografia) 55 (2), pp. 118.

[11] S. Brakatsoulas et al. (2005), *On map-matching vehicle tracking data*, Proc. of 31st international conference on Very large data bases (VLDB '05), pp. 853864.

[12] D. Pfoser and C. S. Jensen (1999), *Capturing the Uncertainty of Moving-Object Representations*, Advances in Spatial Databases, Volume 1651, pp. 111-131, Springer.

[13] B. Hummel (2006), *Map Matching for Vehicle Guidance*, Chapter in Dynamic and Mobile GIS: Investigating Changes in Space and Time, CRC Press.

[14] O. Pink and B. Hummel (2008), *A statistical approach to map matching using road network geometry, topology and vehicular motion constraints*, Proc. of IEEE Conference on Intelligent Transportation Systems (ITSC 2008), October, pp. 862-867.

[15] M. A. Quddus et al. (2006), *A high accuracy fuzzy logic-based map-matching algorithm for road transport*, Journal of Intelligent Transportation Systems: Technology, Planning, and Operations 10 (3), pp. 103115.

[16] G. Nassreddine et al. (2009), *Map matching algorithm using interval analysis and Dempster-Shafer theory*, Proc. of IEEE Conference on Intelligent Vehicles Symposium, June 2009, pp. 494-499.

[17] Y. Lou et al. (2009), *Map-matching for low-sampling-rate GPS trajectories*, Proc. of 17th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems, New York, NY, USA.

[18] J. Krumm et al. (2007), *Map Matching with Travel Time Constraints*, Society of Automotive Engineers World Congress (SAE 2007), Detroit, Michigan, USA.

[19] P. Newson and J. Krumm (2009), *Hidden Markov map matching through noise and sparseness*, Proc. of 17th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems, New York, NY, USA.

[20] P. E. Hart et al. (1968), *A Formal Basis for the Heuristic Determination of Minimum Cost Paths*, IEEE Transactions on Systems Science and Cybernetics SSC4 (2), pp. 100-107.

[21] P. E. Hart et al. (1972), *Correction to "A Formal Basis for the Heuristic Determination of Minimum Cost Paths"*, SIGART Bull. 37 (December 1972), pp. 28-29.

[22] *Pegasus Project: urban mobility management via infotelematics for the safety of passengers, vehicles and goods*, http://pegasus.octotelematics.com/.

[23] *Industria 2015: strategic action for the development of the Italian industial sector*, http://www.industria2015.ipi.it/.

[24] R. Kölbl and D. Helbing (2003), *Energy laws in human travel behaviour*, New J. Phys. 5, pp. 48.1-48.12.

[25] B. S. Kerner (1999), *The physics of traffic*, Physics World Magazine 12, pp. 25-30.