# ALGORITHMS FOR MAP-AIDED AUTONOMOUS INDOOR PEDESTRIAN POSITIONING AND NAVIGATION

THÈSE N° 3961 (2007)

PRÉSENTÉE LE 23 NOVEMBRE 2007

À LA FACULTÉ DE L'ENVIRONNEMENT NATUREL, ARCHITECTURAL ET CONSTRUIT
LABORATOIRE DE TOPOMÉTRIE
PROGRAMME DOCTORAL EN MATHÉMATIQUES

ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE

POUR L'OBTENTION DU GRADE DE DOCTEUR ÈS SCIENCES

PAR


Ivan SPASSOV

Engineer - Geodesist, University of Architecture, Civil Engineering and Geodesy, Sofia, Bulgarie
et de nationalité bulgare


acceptée sur proposition du jury:

Prof. T. Mountford, président du jury
Prof. B. Merminod, Prof. M. Bierlaire, directeurs de thèse
Prof. A.-G. Dumont, rapporteur
Dr N.-E. El Faouzi, rapporteur
Prof. M. Wieser, rapporteur

ÉCOLE POLYTECHNIQUE
FÉDÉRALE DE LAUSANNE

Suisse
2007

To Anastassia

# Acknowledgements

The doctoral thesis is a journey in the labyrinth of knowledge where crossing the other people and sharing ideas you will answer your questions and discover the way to a new challenge.

I would like to thank the numerous professors and collaborators of the EPFL for their help and discussions on various thematic.

I would like to thank my colleagues from the geodetic engineering laboratory - TOPO for their cordial reception, for their collaboration and friendship. I have no words to express my gratefulness and I am proud of being part of that outstanding team for the time of my thesis. Particularly, I would like to thank Pierre-Yves Gilliéron for the exceptional advices and pertinent discussions. Special thanks to the secretary of the laboratory, Véronique Boillat Kireev, for her help in all administrative tasks.

Especially, I would like to thank Prof Bertrand Merminod, my thesis supervisor, for affording me the opportunity to make this doctoral thesis, for the excellent supervising, for the confidence, for the patience, for all professional and personal suggestions.

I would like to thank Prof Michel Bierlaire, the director of the TRANSP-OR laboratory and my thesis co-supervisor, for the encouragement, for the brilliant advices, for leaving the door open for all my questions.

Special thanks are dedicated to Dr Quentin Ladetto for his cooperation and to the team of Vectronix AG for their assistance. I would like to thank as well the collaborators from DII-EPFL and the team of Camp-to-Camp SA for their collaboration.

I would like to thank cordially my wife and children for the support, the energy and the love they are giving me all the time.

Thank you all !

Lausanne, 23 November 2007

Ivan Spassov

# Summary


The personal positioning and navigation became a very challenging topic in our dynamic time. The urban canyons and particularly indoors represent the most difficult areas for personal navigation problematic. Problems like disturbed satellite signals make the positioning impossible indoors. Recently developed systems for indoor positioning do not assure the necessary positioning accuracy or are very expensive.

Our concept stands for a fully autonomous positioning and navigation process. That is, a method that does not rely on the reception of external information, like satellite or terrestrial signals. Therefore, this research is based on the use of inertial measurements of the human walk and the map database which contains the graphic representation of the elements of the building, created by applying the link-node model. Using this reduced set of information the task is to develop methodology, based on the interaction of the data from both sources, to assure reliable positioning and navigation process. This research is divided in three parts.

The first part consists in the development of a methodology for initial localization of the person indoors. The problem to solve is to localize the person in the building. Consider a person equipped with a system which contains set of inertial sensors and map database of the building. Speed, turn rate and barometric altitude are measured and time-stamped on each step of the person. A pre-processing phase uses these raw measurements in order to construct a polyline, thus representing user's trajectory. In the localization approach central place takes the association of the user's trajectory with the graph representation of the building, process known as map-matching. The solution is based on statistical method where the determination of the user's position is entirely represented by its probability density function (PDF) in the frame of Bayesian inference. Initial localization determines the edge of the graph occupied by the person.

The second part aims at continuous localization, where user's position is estimated on every step. Besides the application of the classical map-matching techniques, two new methods are developed. Both rely on the similarity of the geometry of the trajectory and the elements of the graph. The first is based on the Bayesian inference, where the estimation is computed considering the walked distance and azimuth. The second method represents a new application of the Fréchet distance as degree of similarity between two polylines.

The third part is pointed at the pedestrian guidance. Once the user's position is known it is easy to compute the path to his destination and to give him directions. The problem is to assure continuance of the process of navigation in the case when the person has lost his path. In that case the solution consists in either giving instructions to the user to go back on the path or computation of a new path from the actual position of the user to his destination.

Based on that methodology, algorithms for initial localization, continuous localization, and guidance were created. Numerous tests with the participation of several persons have been provided in order to validate the algorithms and to show their performance, robustness and limits.


# Keywords

Pedestrian, localization, navigation, guidance, indoors, map-matching, map database, autonomous, Bayesian inference, Fréchet distance, Dead reckoning, IMU

# Résumé

La localisation et la navigation personnelles sont devenues un domaine très à la mode dans une société de la mobilité. Les zones urbaines et, en particulier, l'intérieur de bâtiment représentent les zones les plus exigeantes pour la navigation personnelle. Des problèmes comme la réception des signaux satellitaires rendent le positionnement impossible à l'intérieur d'un bâtiment. Parallèlement, les infrastructures de télécommunications sont en développement croissant, toutefois le positionnement basé sur ces systèmes (sans fils), n'assurent pas la précision nécessaire en localisation et n'offrent pas une couverture complète.

Le but de cette recherche est de développer un nouveau processus de positionnement et de navigation entièrement autonome. C'est-à-dire, une méthode qui ne dépend pas de la réception d'informations externes, comme des signaux satellitaires ou terrestres. L'approche présentée ici est basée sur l'utilisation des mesures du déplacement de personnes par des capteurs inertiels et sur le contenu d'une base de données géographique qui représente les espaces de circulation dans les bâtiments. En utilisant cet ensemble d'information, on a développé une méthodologie, basée sur l'interaction des données des deux sources de données, pour assurer un processus de positionnement et de navigation fiable. Cette recherche est divisée en trois parties.

La première partie consiste en l'élaboration d'une méthodologie pour la localisation initiale de la personne à l'intérieur d'un bâtiment. On considère une personne équipée d'un système qui contient l'ensemble des capteurs inertiels et la base de données géographique. La vitesse, la direction de marche et l'altitude sont mesurés par les capteurs et datés à chaque pas de la personne. Une phase de prétraitement utilise ces mesures brutes pour construire une polyline, représentant ainsi la trajectoire de l'utilisateur. Dans cette approche de localisation, l'association de la trajectoire de l'utilisateur avec le contenu de la base de données géographique (le graphe) prend une place centrale, ce processus est appelé map-matching. La solution proposée est basée sur des méthodes statistiques où la détermination de la position de l'utilisateur est entièrement représentée par sa fonction de densité de probabilité dans le cadre de l'inférence bayésienne. Le processus de localisation initiale détermine l'arête du graphe occupé par la personne.

La deuxième partie est consacrée à la localisation continue, où la position de l'utilisateur est estimée à chaque pas. Outre l'application des techniques classiques de map-matching, deux nouvelles méthodes ont été développées. Toutes les deux sont basées sur la ressemblance entre la géométrie de la trajectoire et les éléments du graphe. La première méthode est basée sur l'inférence bayésienne, où l'estimation de la position est calculée en considérant la distance et l'orientation des époques précédentes. La deuxième méthode s'appuie sur une nouvelle

application de la distance de Fréchet afin d'évaluer le degré de ressemblance entre deux polylines.

La troisième partie est consacrée au guidage de piéton. Une fois que la position de l'utilisateur est connue, il est facile de calculer le chemin jusqu'à son point de destination et de lui indiquer les directions à suivre. Le problème est d'assurer la continuité du processus de la navigation dans le cas où la personne a perdu ou quitté son chemin. Dans ce cas la solution consiste à donner de nouvelles instructions à l'utilisateur pour retrouver son chemin ou à calculer un nouveau chemin depuis la position courante de l'utilisateur jusqu'à son point de destination. Cette approche est entièrement dépendante de la qualité de localisation continue.

Basé sur ces méthodologies, des algorithmes pour la localisation initiale, la localisation continue, et le guidage ont été créés. Des nombreux essais avec la participation de plusieurs personnes ont été effectués afin de valider les algorithmes et de montrer leur performance, leur robustesse et leur limites en milieu construit.

# Mots-clés

Pédestre, localisation, navigation, guidage, map-matching, base de données cartographiques, autonome, approche Bayesienne, distance de Fréchet, Dead reckoning, IMU

TABLE OF CONTENTS

TABLE OF FIGURES

# Chapter 1

## Introduction

The dynamism of our time imposes more and more the need for personal navigation especially in urban areas. With the development of complex buildings and structures inevitably arises the question of indoor personal navigation. Guiding the customer in the commercial centre; directing the visitor in the hospital or evacuating a worker out of a dangerous area; these are just few examples of indoor personal navigation problematic.

Directing the user to his final goal, known as route guidance, is the main task of the navigation process. The route guidance is preceded by computation of the optimal path from the user's actual location to his destination point, referred to a map. Normally, the destination point is defined from the beginning and can be redefined at any moment. But in order to make the route guidance possible we need to determine the location of the person.

The motivation in this research is to develop a methodology for autonomous indoor personal navigation, which assures reliable and precise localization and route guidance to the user. An autonomous navigation method is a method that does not rely on external information. That would set the user independent from the availability and drawbacks of the external positioning systems discussed further.

The personal localization can be considered as the first task of the navigation process. It is defined as the detection of the person and determination of his position on the map [Legat 2002].

Many sensor-based methods for personal positioning have been developed recently. These methods rely either on the reception of satellite signals or on the signal transfer between a portable device and a network of distributed sensors.

The world's most famous positioning system is GPS. There exist many handheld GPS receivers on the market with very good performance in open sky areas. However in the city GPS positioning suffers from degraded satellite availability or multipath error arising from signals reflected by the buildings [Syed 2004]. Moreover indoors the application of GPS generally is out of question [Lee et al. 2000].

Assisted GPS (AGPS) dramatically improves the performance of GPS receivers [LaMance et al. 2002] in urban areas and even indoors. However, in the best case the precision of positioning indoors is not better then 15 meters, which is not enough for most personal navigation applications. On the other hand, indoors AGPS alone is not enough. It requires the installation of a large number of devices, which can be very expensive [Global Locate 2003]. In addition the positioning process depends on the performance of those devices, which is in contradiction with our concept for autonomy. For example, a simple power cut in the building is sufficient to set the positioning process impossible.

There exist modern positioning systems that use WiFi technology to detect and react to the position of a person [Köbben et al. 2006]. Although their high positioning accuracy, these systems are very expensive and like the A-GPS do not allow for autonomous positioning.

Location-based services (LBS) become very popular these days [Wierenga et al. 2005]. LBS are offered by some cell phone networks as a way to send information to cell-phone subscribers based on their current location. The cell-phone service provider gets the location from a GPS chip built into the phone, or using radiolocation and trilateration based on the signal-strength of the closest cell-phone towers [Magon et al. 2001]. LBS services use a single base station to determine the location of a phone with maximal accuracy of about 100 m which is largely insufficient for personal navigation purposes. The mentioned above autonomy limitations are valid here as well.

The only positioning method that allows autonomy is dead reckoning (DR). It is a process of determining one's position with respect to a known initial position using relative information on heading, speed and time. Such positioning can be provided by an inertial navigation system (INS). Based on Micro-Electro-Mechanical Systems (MEMS) technology such system contains inertial sensors, it has its own power supply and can be easily carried by the user [Ladetto 2003]. A major problem with DR is that a small error in heading grows over time and results in a large error in position. Therefore, frequent calibration of DR sensors is required.

The other major component of the localization process is the map. In the navigation systems we consider the digital map of some region. The digital map, together with the topological and spatial information, constitutes the map database. The map database contains information on the position, dimensions, capacity, functionality, etc. of the geographical objects. For the purposes of the navigation process the connections between these objects are of interest. For instance, in the vehicle navigation these connections are defined by the road network. The mathematical representation of such network is a planar graph [Bernstein, Kornhauser 1998]. Based on the link-node model, that representation aims at the definition of the topological connections between the elements of the network. The computations of the optimal path on the map are based on that graph.

In the context of the navigation process the information of the graph is used to enhance the localization of the user [Quddus 2006]. User's trajectory, constructed by raw measurements, is confronted with the elements of the graph in order to identify the correct road segment on which the user is and to determine his location on that segment. The association of the user's trajectory or part of it with the contents of the road network is known as *map-matching*. The enhanced localization by map-matching assures a feedback to the navigation system allowing the calibration of its sensors.

In the frame of indoor personal navigation the same link-node model can be applied to create a graph representation of the building [Büchel 2003]. Like the road network, the corridors, passageways, staircases and elevators in the building can be defined as a graph.

Following our motivation in this research we ignore positioning methods like GPS, WiFi, etc. Thus, our choice is limited to the following two components: INS and the map database. Consider this minimal equipment the condition of autonomy is fulfilled and the question is how to localize and guide the user indoors.

## 1.1. Aim of the thesis

The aim of this thesis is to investigate how the information of the map database can be used to solve the problem of personal navigation indoors. Having the INS as the only source of measurements, the information from the map database can be used as an independent source of information for the localization process. The fusion of data from both sources will be the main principle for the development of algorithms for indoor personal localization and guidance. In this research we treat the following aspects:

- **Definition of the map database for indoor navigation purposes**. Generally a building is represented by a 2D plan of every level. That representation is not convenient for the needs of the navigation. Specific map database must be created representing the building as a graph.
- **Choice of INS**. Different systems and modules for personal localization exist. Some of them consist of distributed sensors on the human body. Others are centralized in a measurement unit. We need to find a device that combines the requirements for accuracy, performance and autonomy.
- **Development of an algorithm for initial localization of the person**. This algorithm is dedicated to solve the first task of the navigation process. Using a very limited quantity of input data (INS, map database) that algorithm has to assure a robust performance. The aim is to determine on which edge of the graph the person is.
- **Development of an algorithm for continuous localization of the person**. This algorithm must take into account the output from the previous one. The aim is to assure continuous localization of the user at every step.
- **Development of a framework for indoor personal navigation**. Consider the user's location as known. Thus the process of navigation can proceed to the next tasks, i.e. path computation and navigation to the destination. The aim is to assure the continuity of the navigation process. The special case of person lost the path will be tackled.

For the definition of map database as for the test ground for all the algorithms we refer to the campus of EPFL. Five years ago, its complex structure imposed the development of a dynamic web tool for online assistance. The first version of that tool, named CartoWeb and developed by Camptocamp SA, allows for finding spatial information on the rooms, buildings and the POI (points of interest) of the campus. Since 2005, the functionality of CartoWeb is upgraded allowing for path computations (refer to *plan.epfl.ch*). The path computations are based on the graphic representation of the buildings. A huge map database is created, applying the well known link-node model discussed in Chapter 2.

## 1.2. Methodology

In our approach the positioning depends entirely on the measurements from the inertial sensors (speed, turn rate) and barometric altitude. During the movement these measurements are time-stamped and registered on each step of the person thus representing user's trajectory as a sequence of points [Ladetto et al. 2001]. The position of each step is determined as a function of the previous step position and relative measurements.

The other source of data is the digital map database. It contains the graph representation of all corridors and passageways in the building. That graph is created in a fixed coordinate system using the link-node model [Philipona 2002]. That link-node model is largely applied for the vehicle navigation where the graph defines the street network of some region.

The problem to solve is to determine the user's location using information from the map database and inertial measurements of the navigation system. In this research the user's trajectory is associated with the elements of the graph, applying statistical methods in combination with map-matching. The methodology is divided in two stages. The *initial localization* consists in finding the edge occupied by the person and the person's orientation on that edge. The *continuous localization* aims at updating the edge and the position of the person on the current edge.

For the association of elements of the trajectory with elements of the map database, similar geometric forms must be identified in both the trajectory and the graph. Since the trajectory is defined by a sequence of points, this set must be transformed to a polyline before searching an association with the graph. This step is necessary, because unlike the set of points, a polyline can be recognized in a graph. The methodology applies first a pre-processing procedure to create this polyline. The pre-processing procedure consists in a number of functions capable to detect critical elements of the pedestrian trajectory like a turn, a stop, a vertical movement, etc. These critical elements are defined as points and connected in a 3D polyline thus representing an adequate input for the process of localization.

The map database cannot be modified. Instead it can be pre-analyzed so only the critical nodes like turns and crossings need to be considered. After the pre-analysis of the map database and the pre-processing of the trajectory, we have two data sources and we must associate similar details from both. The 3D polyline can be considered as the history of the route and its last segment as the actual location of the user.

In this research the proposed solution is based on statistical methods where the history of the route and actual measurements are treated at the same time. The determination of the user's location is entirely represented by its probability density function (*PDF*) in the frame of Bayesian inference. Following this approach the posterior estimation of the user's location is calculated whenever new measurements become available.

The problematic of continuous localization is tackled using three different approaches:

- Classical matching techniques based on specific weighting system;
- Bayesian inference;
- Fréchet distance;

The aim is to estimate the correct edge among several candidates and to determine the user's location on the edge at each step. The first approach is based on geometric and topologic constrains and takes into account only the measurements of the last step. The second approach estimates the user's location using the history of measurements on the edge. Both approaches can work in parallel with the algorithm for initial localization, which will assure a periodic control of the edge estimation. The third approach estimates the user's location on the edge, but its main task is to control the estimation of the other two approaches.

The route guidance treats the estimation at each step in order to detect a deviation from the computed path. The problem of path finding is tackled with re-computing the optimal path and sending updated instructions to the user, if necessary.

## 1.3. Contribution

The main contributions of this thesis are the algorithms for autonomous indoor personal navigation. They assure absolute localization of the person using only inertial measurements and information from the map database.

The algorithm for initial localization does not use any preliminary information for the estimation. The person is localized after several iterations.

The first algorithm for continuous localization, based on classical matching techniques applies a methodology developed for the vehicle navigation. Its performance for pedestrian trajectory has been evaluated.

The second algorithm for continuous localization is based on Bayesian inference. For the first time it applies a statistical approach to personal localization using only inertial measurements.

The third algorithm for continuous localization presents a completely new application of the Fréchet distance. It is applicable in post-treatment only and it shows a robust performance.

# Bibliography


Bernstein, D. and Kornhauser, A. (1998). Map matching for personal navigation assistants. Research Report. Operations Research and Financial Engineering, Princeton University, Princeton, NJ.

Büchel, D. (2003) Méthodes de guidage applicables au plan d'orientation de l'EPFL, Travail de diplôme, Laboratoire de topométrie, EPFL, Suisse

Global Locate (2003) A-GPS technology overview, Press Release, San Jose, California, March 17, 2003

Kobben, B., van Bunningen, A., Muthukrishnan, K. (2006) Wireless Campus LBS, Springer, 2006. (Lecture Notes in Geoinformation and Cartography) ISBN: 978-3-540-34237-3. pp. 399-408

Ladetto, Q. (2003) Capteurs et Algorithmes pour la Localisation Autonome en Mode Pédestre, Phd Thesis, École Polytechnique Fédérale de Lausanne, 2003.

Ladetto, Quentin, Gabaglio, Vincent, Merminod, Bertrand (2001), Combining Gyroscopes, Magnetic Compass and GPS for Pedestrian Navigation, Geodetic Engineering Laboratory, EPFL

LaMance, J., DeSalas, J., Jarvinen, J. (2002) Assisted GPS: A Low-Infrastructure Approach, GPS World, Mar 1, 2002

Lee, Seon-Woo and Mase, Kenji, (2000) Incremental Motion-Based Location Recognition, ATR Media Integration & Communications Research Laboratories, Kyoto 619-0288 JAPAN

Legat, Klaus (2002) Pedestrian navigation, Graz University of Technology, April 2002

Magon A., Shukla R., (2001), LBS, the ingredients and the alternatives, RMSI A-7, Sector – 16, Noida-201 301, UP, India, Asian GPS Conference 2001

Philipona, C. (2002) Ne perdez pas le nord!, Camptocamp SA, Parc scientifique, Lausanne, Switzerland, Flash informatique, EPFL, 7-17 septembre 2002

Quddus, M. A. (2006) High integrity map matching algorithms for advanced transport telematics applications. PhD diss., Centre for Transport Studies, Imperial College London.

Syed, S. (2004) GPS Based Map Matching in the Pseudorange Measurement Domain, ION GNSS, Long Beach, CA, September 21-24, 2004

Wierenga, J., Komisarczuk, P. (2005) SIMPLE – Developing A LBS Positioning Solution, Victoria University of Wellington

# Chapter 2

## State of the art

In this chapter we present an overview of different aspects of navigation. First we discuss the geographic map database, then we present the existing methods for personal positioning. Finally, we describe the concept of map-matching and we discuss the existing methods.

## 2.1. Geographic map database

The main component in the process of navigation is the map. Here we put the accent on the digital form of the map information and on the principal producers and users of map database. Then we discuss the graphical representation of the road network and the relevant computations. We develop the concept of a digital map database suitable for pedestrian navigation.

### 2.1.1. Digital map and geocoding

The base for traditional mapping as seen in most atlases and road maps are the line drawings and shaded backgrounds.

In digital form it consists of vector data that defines lines, points and polygons, where lines represent features such as roads and political boundaries, points represent the location of items of interest without significant dimension, such as the location of a town, a building or a monument on small scale maps, and polygons which are used to define areas of consistent classification such as a state, a park or a municipality on a map of appropriate scale. These lines, points and polygons all have a specific location on the surface of the Earth directly or indirectly indicated by latitude and longitude coordinates.

This is referred to as *geocoded* data. In addition to being geocoded these lines, points and polygons typically have attribute data associated with them. Such attribute information can be a name, such as the name of a town or highway, a population in the case of cities and towns, a definition of its general properties indicating whether the item is a state, a park, a forest or some other feature, to mention but a few of the possibilities.

The geocoding is the process of recording a location identifier as part of a data record. If a data record has been geocoded, an identifier for relevant location has been added to the record. The identifier is translatable by a computer into a location on a map. For example, an accurately geocoded record of a residence can be placed by a computer at the position on a map that the owner would recognize as the location of his/her home.

The fact that the data is geocoded allows very different sets of data, derived from different sources and at different times, to be overlaid one on the other and matched with other data sets such as satellite imagery and digital elevation data.

The term digital elevation model or DEM is frequently used to refer to any digital representation of a surface, however, most often it is used to refer specifically to a raster or regular grid of spot heights. Digital terrain model (DTM) is used more specifically for a topographic surface, excluding the trees and the buildings to refer to any digital representation of a topographic surface. The DTM is the most common form of digital representation of the topography.

The digital map contains information about the routes and other relevant facilities such as bus stops, intersections, landmarks etc. The position of these features must be sufficiently accurate to be clearly identified on the map [Greenfield 1998]. Moreover digital map must assure the basis for path computations. That means it must include the graph representation of the road network, discussed further.

### 2.1.2. Geographic Information System (GIS)

Maps are required to show the relative position of objects to each other. GIS technology, or more precisely the introduction of topology concepts into the information technology is the key for the automatic processing of geospatial data, making visualization redundant in many cases.

A geographic information system can manage, analyze, and display geographic knowledge, which is represented using a series of information sets. The information sets include:

- *Maps and Globes*. Interactive views of geographic data with which to answer questions, present results, and use as a dashboard for real work. Maps and globes provide the advanced GIS applications for interacting with geographic data.
- *Geographic Data Sets*. File bases and databases of geographic information - features, networks, topologies, terrains, surveys, and attributes.
- *Processing and Work Flow Models*. Collections of geoprocessing procedures for automating and repeating numerous tasks, as well as for analysis.
- *Data Models*. GIS data sets are more than database management system (DBMS) tables. They incorporate advanced behavior and integrity like other information systems. The schema, behavior, and integrity rules of geographic data sets play a critical role in GIS.
- *Metadata*. Documents describing the other elements. A document catalogue enables users to organize, discover, and gain access to shared geographic knowledge.

Geographical data is most often separated into two components: spatial data and attribute data. Spatial data is used in the visualization and manipulation of real world objects in a computer model, e.g. roads, buildings, crime locations. Typically, spatial data is presented as features on a digital map. Attribute data (textual, numeric, photographic, etc.) describes these real-world objects, e.g. name, cost, size, and an image of what the object looks like. These two components often are stored in different data structures, in separate databases.

The sources of spatial data encompass all current survey techniques. These include field survey, photogrammetry, remote sensing, GPS, digitization, digital photography, airborne LIDAR (light detection and ranging), etc. All these methods now produce coordinate referenced data in an acceptable digital form for input into a GIS. Moreover, existing paper maps still provide a major source of spatial data.

The structure of the database and format of the spatial data will always affect the types of algorithms available for spatial analysis [Taylor et al. 2006]. Conversely, the type of analysis may require a certain data format or database structure. For example, the analysis of utility of the transport networks is really only feasible with vector data format.

Network modelling, management, and analysis are common tasks for the geographic information systems. Network analysis includes routing (path computation) in transportation networks, tracing (accessibility) in utility networks, and resource allocation in decision-making and customer relationship management (CRM) applications.

There are two common types of GIS applications: applications that deal with spatial proximity, and applications that deal with both spatial proximity and connectivity. For many applications queries on spatial proximity, which mainly consider metrics like Euclidean or geodetic distances, are sufficient. However, there are instances when connectivity needs to be taken into account, such as in finding the shortest distance between two locations in a road network.

TransCAD is probably the best example of GIS-based travel demand modelling application. It is a tool for transportation planning that supports many styles of travel demand modelling including sketch planning methods, models for multiple choices, and extensive set of traffic assignment models. Travel forecasting models are used to predict changes in travel patterns and the use of the transportation system in response to changes in regional development, demographics, and transportation supply. Modelling travel demand is a challenging task, but one that is required for rational planning and evaluation of transportation systems.

### 2.1.3. Street network and graph representation

Many applications require the ability to model and analyze relationships among objects of interest. A network (or a graph) is one of such modelling representations.

The graph theory developed a topological and mathematical representation of the nature and structure of transportation networks [Rodrigue 2007]. The core purpose of a network data model is to provide an accurate representation of a network as a set of links and nodes. Topology is the arrangement of nodes and links in a network defining their location, direction and connectivity (Fig. 2.1).

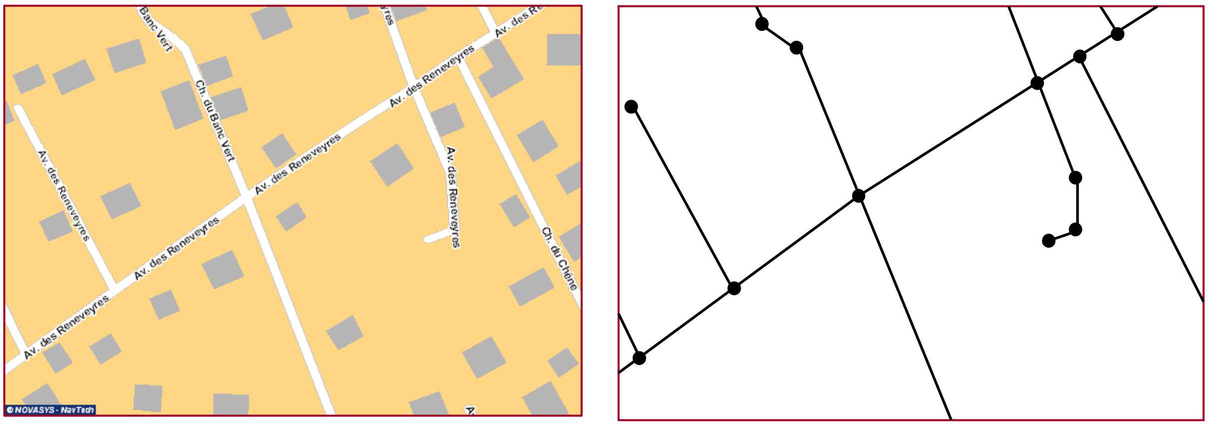



Fig. 2.1 Graph representation of street network

Two fundamental tables are required in the basic representation of a network data model that can be stored in a database:

- Node table. This table contains at least three fields; one to store a unique identifier and the others to store the node's X and Y coordinates. Although these coordinates can be defined by any Cartesian reference system, longitudes and latitudes would insure an easy portability to a GIS.
- Link table. This table also contains at least three fields; one to store an unique identifier, one to store the node of origin and one to store the node of destination. A fourth field can be used to state if the link is unidirectional or not.

Once those two tables are linked, a basic network topology can be constructed and all the indexes and measures, referred to the graph theory, can be calculated.

### 2.1.4. Road map database

Network modelling transforms network applications into a network representation (nodes, links) so that network analysis can be performed. Applications extract connectivity information and maintain mapping relationships between network elements and application features.

A street network consists of streets, roads and highways, which are mapped to links and nodes.

**Geographic Data Files (GDF) standard**

The navigation systems need a lot of detailed and dedicated geographical information. Partners like Philips and Bosch have proposed a standard, called GDF, for the acquisition and representation of this information. The first release has been published in October 1988 [Claussen et al. 1988].

GDF is divided into 3 parts, which can be considered as independent of each other.

- The Specification of Data Content (SDC) specifies which information is required for the navigation system.
- The Specification of Data Acquisition (SDA) defines a set of features, attributes, and relationships for the representation of the non-spatial aspects of the required information. Further, the SDA defines a set of so-called cartographic primitives, meant for the representation of the spatial aspects. These primitives are related to the three different representation levels which correspond to the different needs of the different functions of a car navigation system.
- The Exchange Format (EF) defines a collection of records and fields to enable the digital representation of these data.

The main concepts in the GDF data model are features, attributes and relationships. A feature is a formalized entity that is used to represent a topographical object. The properties and particularities of the objects are represented by means of attributes. Properties involving more than one feature are described by means of relationships. A group of features that is strongly related is called a layer.

- Layers

Seven layers are defined: Administrative Areas, Settlements, Buildings, Roads & Ferries, Bridges & Tunnels, Railways and Waterways.

- Features

Here only the features referred to navigation problematic are shown. The Roads & Ferries layer contains three elementary feature types and three complex ones. The elementary feature types are Road Element, Ferry Connection and Junction. The complex ones: Road, Ferry and Intersection. These allow for a more generalized view of the road network by aggregating two or more elementary features. The Buildings layer has only the Building feature. The Bridges & Tunnels layer contains only one feature called Brunnel (a contraction of bridge and tunnel). It is meant to represent also objects like fly-overs, elevated highways etc. The Railways layer contains the Railway Element and Railway Junction features.

- Attributes

A Road Element can have the attributes Direction of Traffic Flow (to represent one way roads and roads entirely closed to motorized traffic), Form of Way (motorway, dual carriageway, roundabout, square, parking place), Road Class (primary road, secondary road etc.), Route Number (A-56, E-34), Maximum Height, Maximum Length, Maximum Width, Maximum Weight, Special Traffic Restrictions (closed to dangerous loads, private roads), House Number Range and Traffic Sign along Road Element.

A Ferry Connection can have the attributes Road Class and Route Number (ferries are considered to be an integral part of the whole road network), Maximum Height, Maximum Length, Maximum Width and Maximum Weight. Furthermore, it can have the attribute Ferry Type (to enable the distinction between boat ferries and train ferries).

The Building feature can have the attributes Building Class (Hotel, Theater etc.), Opening Days followed by Opening Hours and Brand Name (Hilton, Renault).

The Brunnel feature can have the attribute Brunnel Type to distinguish between bridges, viaducts, aqueducts or tunnels.

- Relationships

Some relationships have been defined to represent traffic-related information involving more than 1 feature. It concerns the relationships Prohibited Turn, Right of Way and Sign Post Information (meant to represent the place names and route numbers that occur on the signposts related to a particular turn).

Other relationships are created to enable the description of situations in which the logical tie between two features is not equivalent to their spatial one, e.g. a Building that has its entrance at a Road Element other than the one which is closest.

The relationships that have been defined for these situations are: Road Element in Municipality, Junction in Municipality, Building in Municipality, Settlement in Municipality and Building along Road Element.

A third type of relationship is created to describe what Road Element is on top of another Road Element. The relationship that was defined for this purpose is named "X over Y through Brunnel" where X and Y may stand for a Road Element, a Waterway Element or a Railway Element.

- Geometry and topology

Common property of all geographic features is that they have an (almost fixed) location relative to the Earth and often also a shape. These properties are commonly known as the geometry. For some features (e.g. Road Elements), an explicit knowledge of the mutual connectivity and the relative spatial positions between the individual elements is of vital importance.

This knowledge is called topology. In principle, GDF admits the representation of a geographic feature without the description of the geometry. This option can be used if one is only interested in the logical relationship between features (e.g. Road Elements that belong to a Municipality) and not in the location and shape.

- The cartographic representation model

The next important choice to be made is how each feature type shall be represented by means of points, lines and other kinds of cartographic primitives and how names and attributes shall be attached to them. The choice for such a set of primitives, together with the representation rules for each feature, is known as the cartographic representation model.

A main property of the GDF model is that it distinguishes three different representation levels, called Level-0, Level-1 and Level-2. These levels should not be considered as completely separate representations, but as different structures in one and the same representation, the higher ones embedded in the lower ones. Each representation level has a corresponding set of primitives.

A Level-0 representation is built up of Segments, Intermediate Points, Nodes and Chains. Nodes and Intermediate Points are represented by exactly one pair (XY) or triplet (XYZ) of coordinate values. A Segment is bounded by exactly two Intermediate Points and/or Nodes. A Chain contains always one or more Segments and is always bounded by exactly two Nodes. Nodes and Chains together form a planar graph.

The elements of a Level-1 representation are Spots, Lines, and Polygons. A Line is related to 1 or more connected Chains and is bounded by exactly two Spots. A Polygon is bounded by one or more Lines, describing the outline of the Polygon. Lines and Spots together may form non-planar graphs. However, the graph of Lines and Spots which are used to describe the outline of a Polygon has to be planar. The cartographic primitives at Level-2 are called Composite Spots, Composite Lines and Composite Polygons. These primitives are meant to aggregate Level-1 primitives.

In the GDF specifications, much attention has been paid to the so-called "global data", .i.e. data that are needed to interpret the feature data in a correct way. There are different groups of data. Here we will mention two of them. The set of geodetical data gives information about items such as the horizontal datum, projection system, projection parameters, origin of the coordinate system, coordinate offsets, vertical datum, geoid ondulation and magnetic declination. The other group is a group of global data describing the quality of the feature data. The quality aspects that are addressed are: geometrical accuracy, update status, completeness of features and attributes and error rates in attribute values.

The GDF standard is used by leading providers of geographic map databases for most of the European and American countries, such as TeleAtlas, NavTeq and EGT [Claussen 1993].

**TeleAtlas, NavTeq, EGT**

TeleAtlas provides a database of geographic content which encompasses currently 54 countries, 787 million addresses, and nearly 13 million miles of roadway around the world, the most of any geographic data provider. The database also includes enhancements such as turn-by-turn data, POIs, dynamic data, brand icons, detailed building polygons, 3D landmarks, phonetic spelling, and geocoded products and services. Tele Atlas relies on more than 50,000 data resources. These sources include aerial imagery; public and government sources; and utility, fleet, and postal drivers. Tele Atlas is the only provider to deploy Mobile Mapping vans with 360-degree visibility throughout the US and Europe to validate the accuracy and increase the detail of road information.

NAVTEQ is another leading provider of comprehensive digital map information for automotive navigation systems and location-based services. The company has built robust and accurate geographic databases of most of the countries of the world.

EGT is a consortium which includes Philips Electronics (Netherlands), Renault (France), QC Data (Ireland), Institut Géographique National (France), and Navigation Technologies (U.S.).

EGT's mission is to create a complete, accurate digital map database for entire European metropolitan areas and the intercity highway system. These areas will be integrated to form a single seamless map database of continental scope [Cass 1992]. The principal initial application for the database is real-time route guidance and other in-vehicle Road Transport Informatics (RTI) applications. For route guidance, the roadway geometry must be accurate to within 10 meters. This is quite stringent and the database will be able to support many other applications as well.

Along with the creation of the database, EGT is focusing on its maintenance and licensing. EGT sees the database as a large-scale generalized data resource which, instead of being designed for specific applications, provides data which may be adapted to almost any application requiring geographic information.

Basic geometry comes from detailed aerial photographs and base maps obtained from government agencies. Other information, such as street names and highway information, is collected from local and regional agencies. Value-added navigation information is acquired from municipalities where available, through direct field work where necessary.

**Oracle, ESRI, Magellan, Garmin**

Oracle is one of the major companies developing database management systems (DBMS) and tools for database development.

One of the latest releases, *Oracle Database 10g*, lets users model and analyze networks. Several features simplify network modelling, analysis, and management and focus on application logic. The network data model provides an open, generic data model with many common GIS analysis capabilities.

For some navigation applications and transportation management a fusion between Oracle and NavTeq is created.

ESRI designs and develops the world's leading GIS technology. One of the most popular products is ArcGIS, an integrated collection of GIS software products for building a complete GIS. ESRI provides a full spectrum of ready-to-use geospatial data products delivered either as a Web service or as packaged media (CD). ESRI transforms the map information in a specific spatial data format, shapefile (.shp).

A shapefile stores non-topological geometry and attribute information for the spatial features in a data set. The geometry for a feature is stored as a shape comprising a set of vector coordinates. Because shapefiles do not have the processing overhead of a topological data structure, they have advantages over other data sources such as faster drawing speed and editing ability. Typically, they also require less disk space and are easier to read and write. Shapefiles can support point, line, and area features.

Magellan is one of the leading producers of navigation systems and GIS solutions. This company introduced the first commercial handheld GPS receiver. The digital map information used in Magellan products is provided by NavTeq.

Garmin is another leading producer of navigation systems for personal and vehicle applications. Unlike its competitors this company develops its own map database (MapSource).

There exist many producers and users of map database. They can be grouped in the following figure.

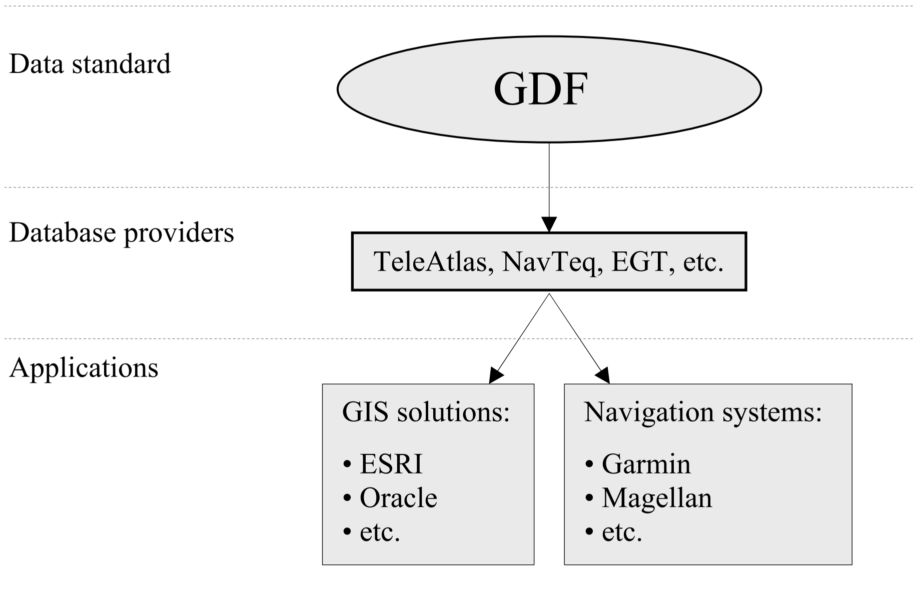



Fig. 2.2. Schema for development and application of the database.

In terms of data storage the road map database may be stored in solid state read-only memory (ROM), optical media (CD or DVD), solid state flash memory, magnetic media (hard disk), or a combination. A common scheme is to have a base map permanently stored in ROM which can be augmented with detailed information for a region the user is interested in. A ROM is always programmed at the factory; the other media may be preprogrammed, downloaded from a CD or DVD via a computer or wireless connection (Bluetooth, WiFi) or directly used utilizing a card reader.

### 2.1.5. Path computations

In order to apply the link-node model for the street network only the connectivity information is needed. The concept of *cost* (or *weight*) is introduced for the links and nodes if it needs to be taken into account during the analysis. The distance or travel time can be reflected by the cost associated with each link.

A *path* is an ordered list of links. When a routing analysis is performed, the shortest (or fastest) path is returned in an ordered link list. The link list is mapped back to the application domain for turn-by-turn directions.

A path from a source (origin) node to a destination node is said to be the shortest path if its total cost is minimum among all paths (Fig. 2.3).

The Dijkstra algorithm is the best known general-purpose shortest path algorithm [Dijkstra 1959]. It determines the length of the shortest path between the given node and any other node in a graph.

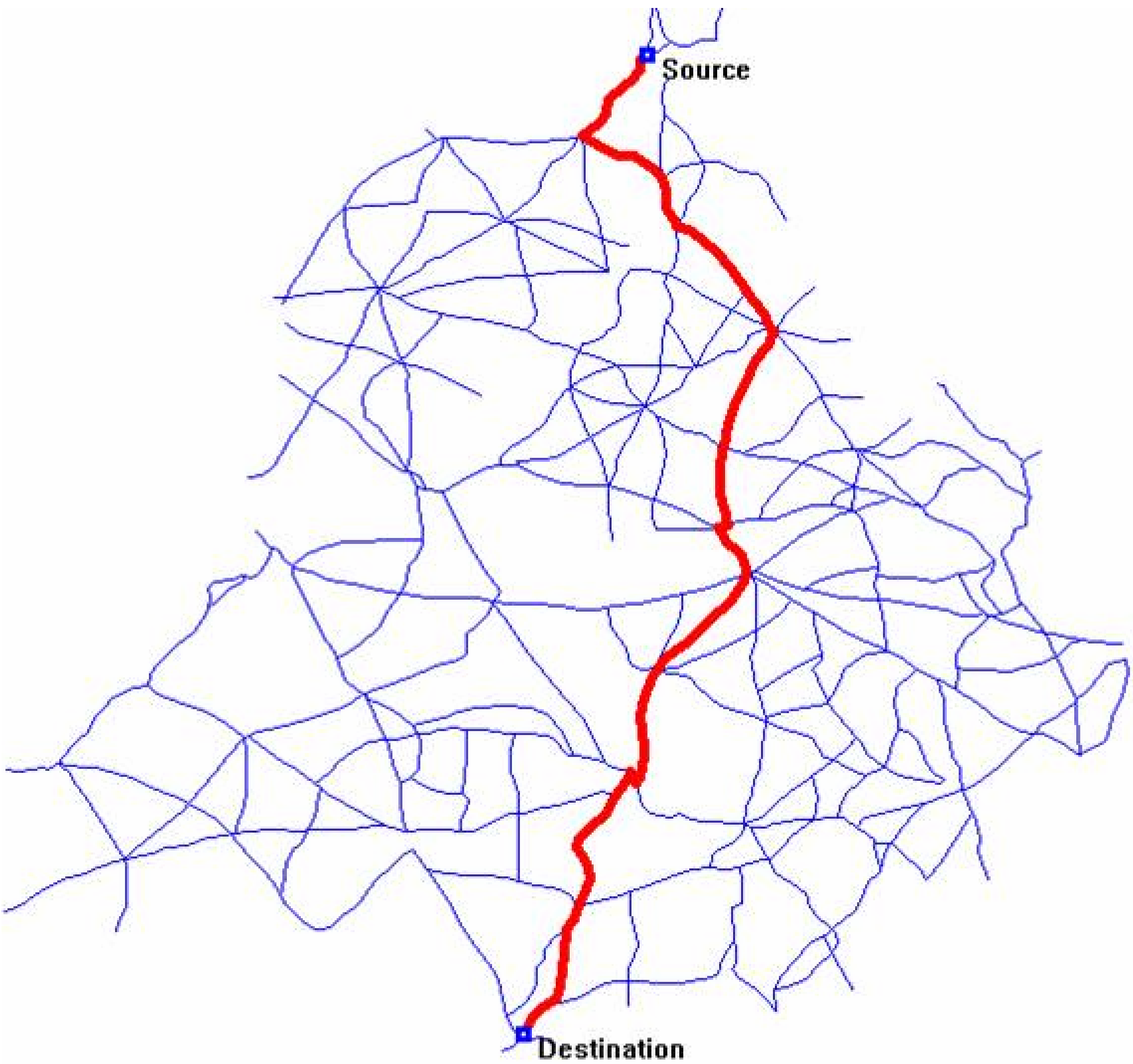


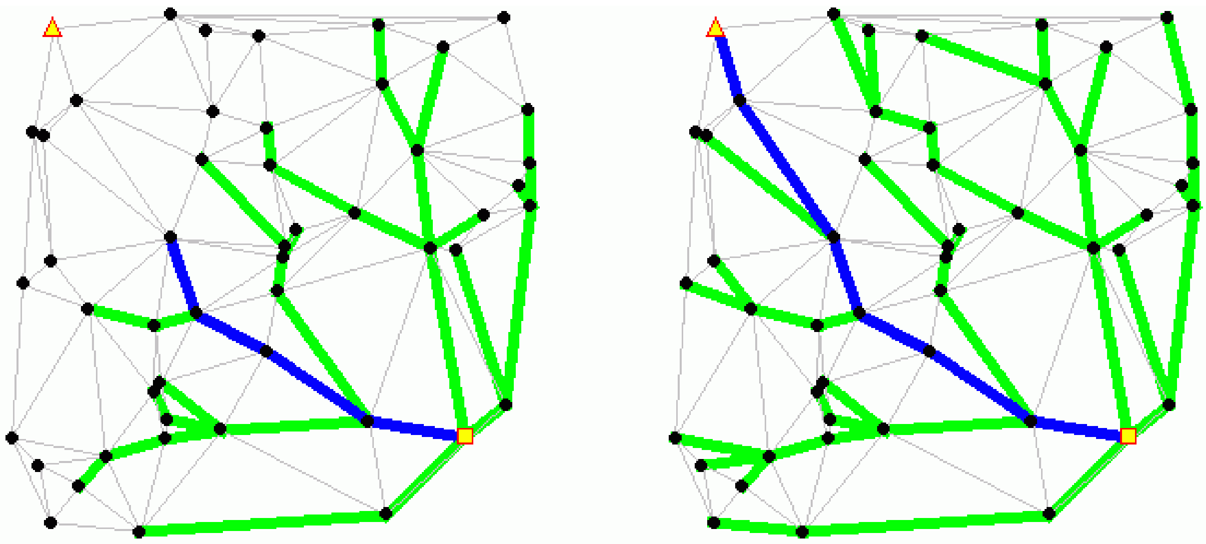

Fig. 2.3 Shortest path between two nodes in a graph

The underlying principle of the algorithm may be described as follows. Let $G = (V, E)$ be a directed graph in which $V$ is the set of nodes and $E$ is the set of links. Link $(u, v)$ has an associated length $l(u, v)$. If link $(u, v)$ exists then $v$ is said to be a *successor* of $u$ and $u$ is said to be a *predecessor* of $v$ [Chandy and Misra 1982].

The algorithm starts with the source node and maintains tentative distance $d(v)$ for each node $v$ during the execution. It visits the nodes in order of increasing distance, and maintains a set of visited nodes whose distance from the source node has been computed.

Initially, $d(s)$ is set to 0 for the source node $s$ and $d(v)$ is set to $\infty$ for node $v$. A so called priority queue stores reached nodes ($d(v) < \infty$) using $d(v)$ as the priority (initially the priority queue contains the source node $s$ only). On each iteration, the algorithm removes a node $u$ from the priority queue and scans the links coming out of $u$. That means for each link $(u, v)$ the distance $d(v)$ is checked if $d(v) > d(u)+l(u, v)$, where $l(u, v)$ is the length (cost) of the link $(u, v)$. If so, $d(v)$ is set to $d(v) = d(u)+l(u, v)$, and $v$ is put into the priority queue (Fig. 2.4a).

When the target node $t$ is reached, $d(t)$ is the shortest path cost from node $s$ to vertex $t$. By executing the shortest path computation from a single node $s$ to all the other nodes, we can construct a Dijkstra tree whose root node is $s$ (Fig. 2.4b). [Chen 2003], [Yanagisawa 2006].



a) b)

Fig. 2.4 Dijkstra tree ( □ - source node $s$, △ - target node $t$, ▬▬ - $d(v)$, ▬▬ - $d(t)$)

Often in the transportation management we need to find the optimal path, which is not necessarily, the shortest one. As example we can give the fastest path, the path excluding highways, the path excluding tunnels, etc. In such cases the Dijkstra algorithm is applied and the attributes of the links are used to compute the generalized cost of the path. These computations and other services are provided by many web sites like ViaMichelin.

### 2.1.6. Map database for indoor pedestrian navigation

The construction of any map database consists in three general phases: the analysis of the environment; the definition of the conceptual model; the data implementation. For the case of indoor pedestrian navigation these three phases are referred to the following [Gilliéron et al. 2004].

The first phase implicates the analysis of the existing information of the building. Commonly, any building is presented by 2D plans which describe the situation of each floor (Fig. 2.5).

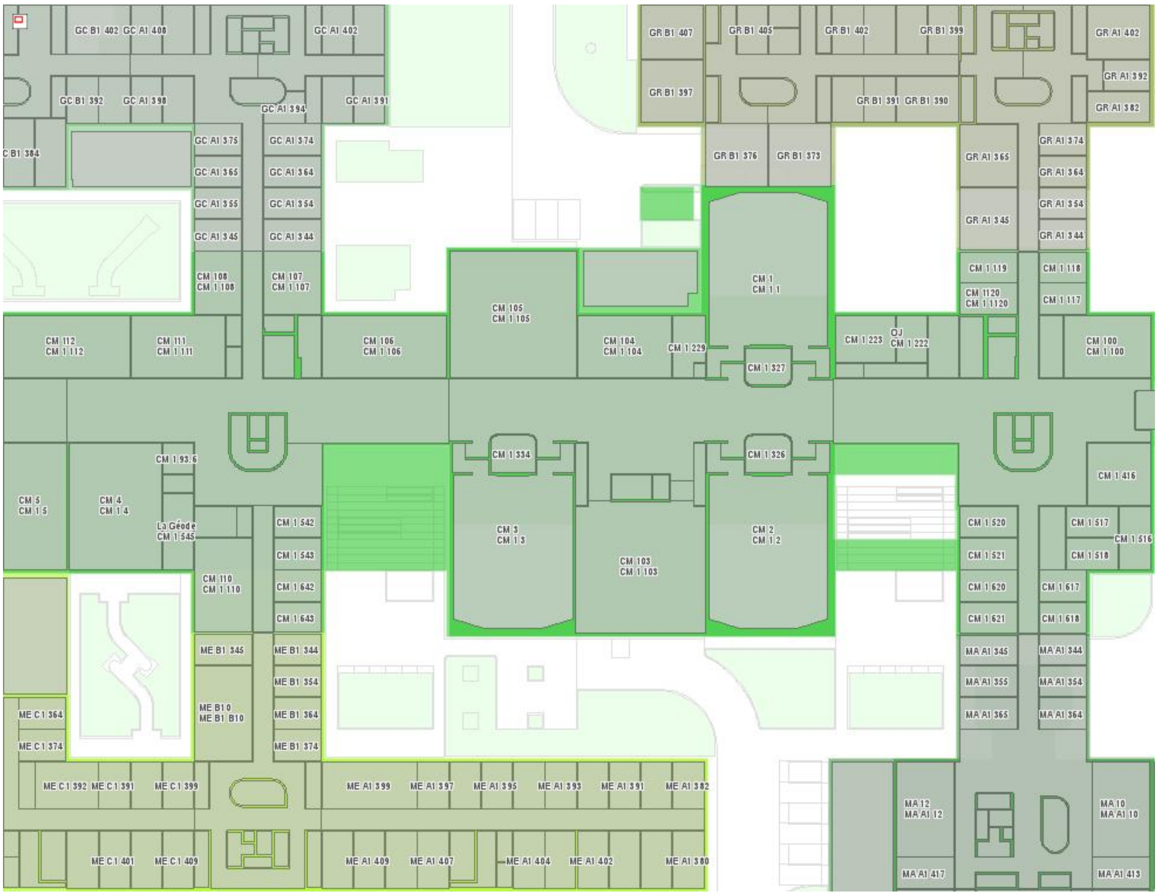


Fig. 2.5 Floor plan of part of the EPFL

This representation is not appropriate for the navigation process. A more detailed analysis of the objects on the plan will permit the definition of certain attributes, e.g. the type of the room, the number of the office, etc. This basic approach can be used to find different points of interest (POI). However, the process of indoor navigation requires the construction of a specific map database [Büchel 2003]. The model applied for the definition of that database must respect the following conditions:

- To assure the correspondence between the object's location and its geographic coordinates. That is, the database model must be capable to recognize a geographic object by its coordinates and vice versa.
- To allow the path computations and guidance.
- To be compatible with the process of map-matching.

These conditions are satisfied by the well known link-node model (refer to 2.1.3.). That model can be applied to create the map database from the 2D plans of the building. The network model is based on the principal elements, nodes and links (Fig. 2.6).

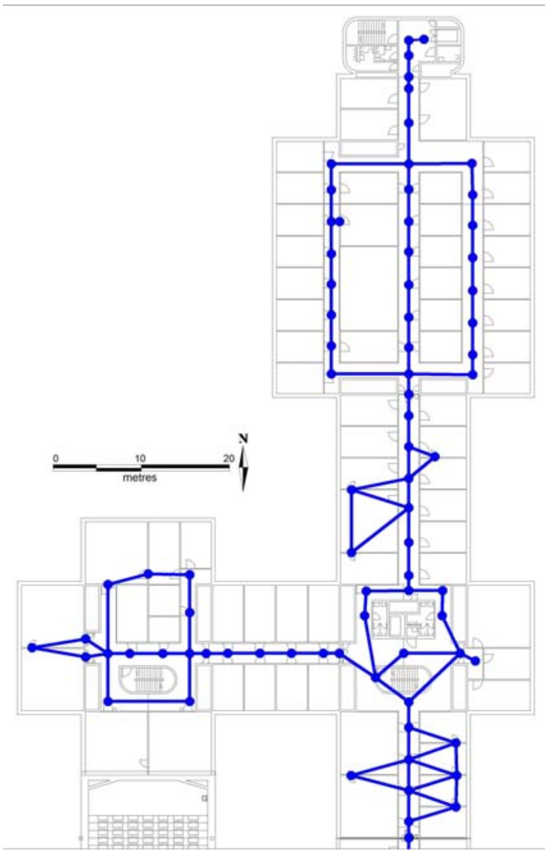



Fig. 2.6 Development of network model for the campus of EPFL

A link is modeled as a straight line, defined by two nodes. The links correspond to the corridors; passageways; entrances of an elevator, a staircase or a room.

The node is a point defined by its coordinates (East, North, H), where H corresponds to the number of floor. A node can constitute the end of one or several links. The nodes correspond to the junctions and to the points of interest in the building.

An important property of the building database model is that the vertical connections are considered. The elevators and staircases are represented as links connecting two nodes from different floors (Fig. 2.7). For simplicity we call these links *vertical links*. That representation is chosen because we consider the staircases and elevators as devices for moving from one floor to another, assuming their functionality as topological connections rather than spatial connections.

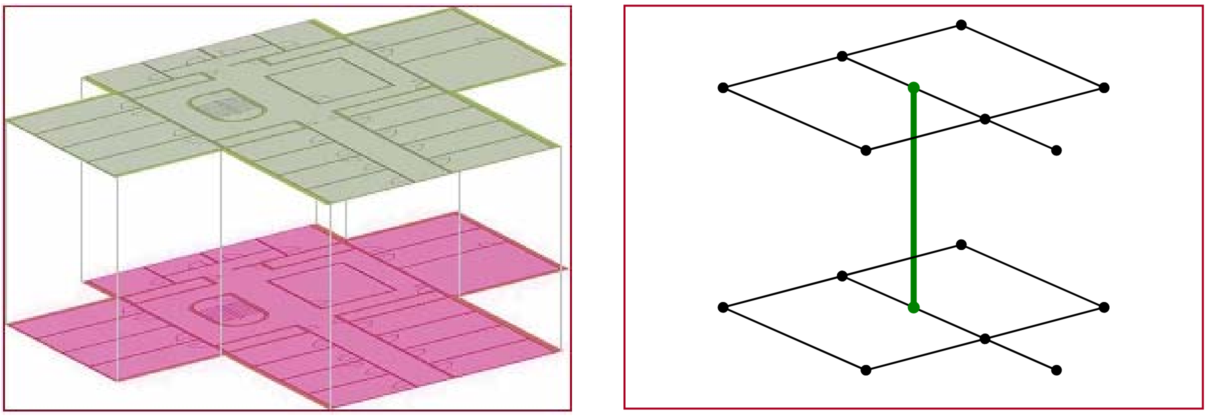


Fig. 2.7 Vertical link connecting two floors

In the context of indoor pedestrian navigation the database could be constituted by the limits of the building. However, it could be connected with the street network database or with the database of other buildings.

The topologic modeling implies the definition of attributes for the nodes and links. As example for such attributes we can give: access restriction; elevator; staircase; internal/external passage; etc. Another very important attribute is the initial cost of the links. These attribute reflects a specific hierarchy of the links in accordance with their usage. That is, an important passage which is frequently used by many people corresponds to a link with high initial cost. All these attributes are taken into account in the path computations.

With the definition of the map database for indoor navigation we constitute the first source of data for the process of navigation.

## 2.2. Systems and methods for personal positioning

### 2.2.1. Existing positioning systems and methods

Here we formulate a brief overview of the most popular positioning systems and methods, making reference to the pedestrian applications.

**GNSS**

- GPS

The world famous satellite based positioning system is the Global Positioning System (GPS). It is a radio-navigation system, funded and controlled by the U.S. Department of Defence (DoD). While there are many thousands of civil users of GPS world-wide, the system was designed for and is operated by the U.S. military [Dana 1994]. GPS provides coded satellite signals that can be processed in a dedicated receiver, enabling it to compute position, velocity and time (Fig. 2.8).

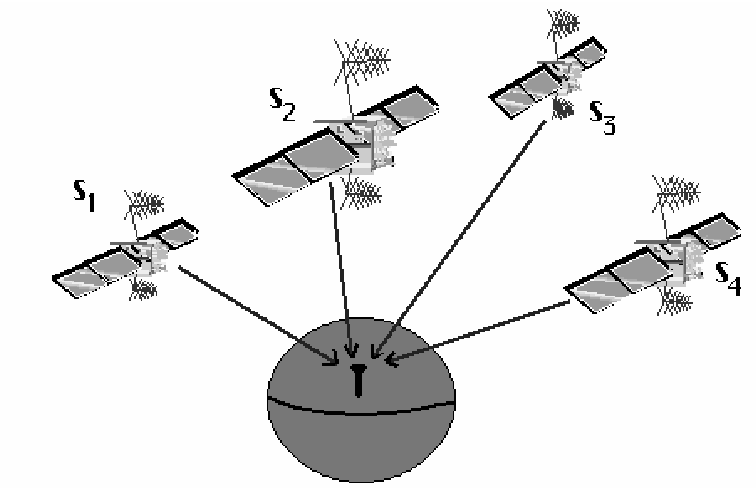



Fig. 2.8 Minimal number of GPS satellites needed to position the receiver

The Global Positioning System is comprised of three segments: Space segment, Operational Control segment and User segment [Parkinson et al.1996].

The Space segment of the system consists of the GPS satellites. The nominal satellite constellation consists of 24-satellite (the first was launched in 1978 and the 24th in 1994). The satellites are positioned in six Earth-centred orbital planes with an inclination of 55° to the equator and four satellites in each plane. The nominal orbital period of a GPS satellite is one half of a sidereal day or 11 h 58 min. The orbital radius (i.e., distance from satellite to centre of mass of the earth) is approximately 26,600 km (Fig. 2.9).

The specifications call for a minimum of 24 operational satellites and three spares to provide a minimum of four visible satellites at any place and any time on the planet. At present, the constellation has 27 satellites.

GPS satellites transmit two low power radio signals, designated L1 and L2. Civilian GPS uses the L1 frequency of 1575.42 MHz in the UHF band. A GPS signal contains three different bits of information: a pseudo-random code, ephemeris data and almanac data [Hoffmann-Wellenhof et al. 1994]. The pseudo-random code is simply an ID code that identifies which satellite is transmitting information.

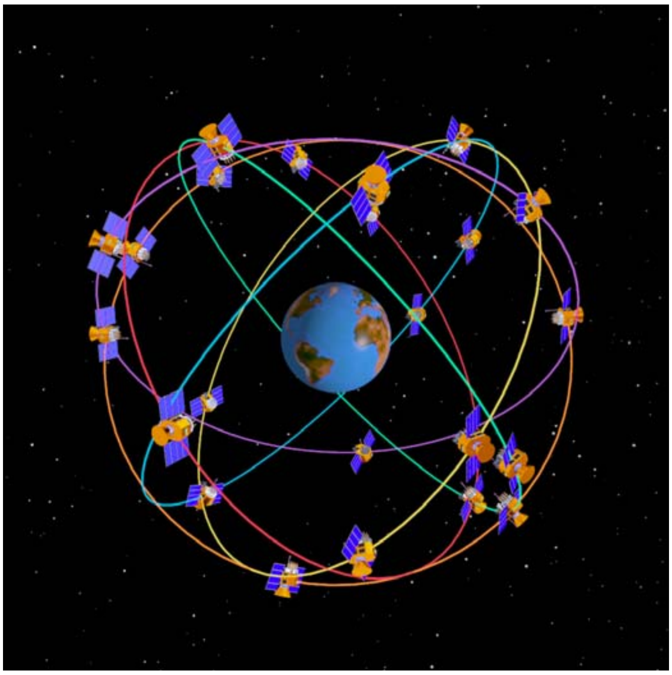


Fig. 2.9 The GPS nominal satellite constellation

The Operational Control segment (OCS) has responsibility for maintaining the satellites. This includes controlling their orbital positions (called station keeping) and monitoring their health and status. The OCS also monitors the satellite solar arrays, battery power levels and propellant levels used for manoeuvres. It also activates spare satellites. The overall structure of the operational ground/control segment is as follows: Remote monitor stations constantly track and gather C/A and P(Y) code from the satellites and transmit this data to the Master Control Station, which is located at Falcon Air Force Base, Colorado Springs. There is also the ground uplink antenna facility, which provides the means of commanding and controlling the satellites and uploading the navigation messages and other data. The unmanned ground monitor stations are located in Hawaii, Kwajalein in the Pacific Ocean, Diego Garcia in the Indian Ocean, Ascension Island in the Atlantic and Colorado Springs, Continental United

States (Fig. 2.10). Ground antennas are located in these areas also. These locations have been selected to maximise satellite coverage.


Fig. 2.10 GPS ground monitor stations

The User segment consists in the user receiving equipment, typically referred to as a GPS receiver. The L-band signals transmitted from the satellites are processed to determine the user's position, velocity, and time. There has been a significant evolution in the technology of GPS receiving sets since they were initially manufactured in the middle 70's. Initially, they were large, bulky and heavy analogue devices primarily used for military purposes. With today's technology, a GPS receiver of comparable or more capability typically weighs a few pounds or ounces, and occupies a small volume. Today, the smallest has the site of a wrist watch, while the largest is a naval shipboard unit (weighing about 32 kg). The basic structure of a receiver is the antenna, the receiver and processor, the display and a regulated DC-power supply. These receivers can be mounted in ships, planes and cars, and provide exact position information, regardless of weather conditions.

The basic idea behind GPS is to use satellites in space as reference points for locations on Earth. Four or more satellites are normally required to compute a GPS position [NAVSTAR press 1996]. However, a variety of different errors can occur within the system, some of which are natural, whilst others are artificial. As a GPS signal passes through the charged particles of the ionosphere and then through the water vapour of the troposphere it gets slowed down, which causes important errors. *Selective Availability* (SA) was used until May 2000 to degrade the performance of civilian users in single-point mode through downgrading of the orbital information and dithering of the satellite clock offset. Different techniques are introduced to eliminate these and many other positioning errors.

Using a modified form of GPS called Differential GPS can significantly reduce the above errors. Even with SA eliminated, DGPS continues to be a key tool for highly precise navigation on land and sea. DGPS can yield measurements accurate to a couple of meters in moving applications and even better, up to 0.1m, in stationary situations. Differential GPS involves the co-operation of two receivers, a stationary one and another one roving around making position measurements [Stephen 2000].

Other methods for decrease the positioning errors are based on satellite augmentation. That means other available information is used to increase the availability and reliability of GPS positioning. That implicates the reception of signals from other systems.

- GLONASS

The Russian Federation's satellite navigation system is very similar to GPS. There are 24 satellites in the full constellation and the system was declared fully operational by January 1996. In many ways, GLONASS has a more economical design than GPS. However, there are a number of quality control issues and much more serious funding problems. Even in its current weakened state, GLONASS still has potential as a stand-alone navigation system and as an augmentation to GPS.

- Galileo

The Galileo satellite radio navigation system is an initiative launched by the European Union and the European Space Agency (ESA). The project architects plan deployment in 2006-2010, becoming operational in 2012. It is projected Galileo to be operational and in service by 2012 with constellation of 30 satellites. The technology behind Galileo is designed to be more accurate and more reliable than GPS or GLONASS. This will allow safety-critical systems - such as air traffic control, and ship and car navigation - to be run. The system should also guarantee coverage to previously inaccessible areas such as those that are either blocked by buildings or isolated areas at high latitudes.

There exist many handheld GPS receivers on the market with very good performance in open sky areas [Mayr 2006]. Such receivers could determine the user's position with an accuracy of up to 5 meters [Legat et al. 2000]. Some receivers, compatible with EGNOS (The European Geostationary Navigation Overlay Service), can achieve a precision of 1 to 3 meters.

However in the city GPS, positioning suffers from degraded satellite availability or multipath error arising from signals reflected by the buildings [Syed 2004]. Moreover indoors, the application of GPS is generally out of question [Li et al. 2005].

- AGPS

Assisted GPS (AGPS) dramatically improves the performance of GPS receivers [LaMance et al. 2002]. That assistance is provided from a terrestrial RF network which sends information about the satellite constellation directly to the GPS receiver via a wireless connection (Fig. 2.11). Thus the receiver is searching only for specific signals. The time taken to obtain the location of the receiver or time-to-first-fix (TTFF) is reduced from minutes to seconds [Abwerzger et al. 2004].

AGPS is very useful in urban areas and even indoors. However, this method presents some limitations. In the best case the precision of positioning indoors is not better then 15 meters, which is not enough for most personal navigation applications.

Fig. 2.11 Functional schema of AGPS technology (source Qualcomm).

On the other hand, indoors AGPS alone is not enough. It requires the installation of a large number of devices, which can be very expensive [Global Locate 2003]. The positioning process is dependent on the performance of those devices, which is in contradiction with our concept of autonomous personal positioning. For example, a simple power cut in the building is sufficient to set the positioning process impossible.

- HS-GPS

High-sensitivity GPS (HS-GPS) receivers can acquire and track signals 20–25 dB below the threshold of a conventional receiver [Lachapelle et al. 2003]. This allows GPS positioning in environments where other receivers might not provide enough observations. HS receivers increase the number of satellite observations available and thus allow GPS positioning in environments like urban canyons, forests, or even inside buildings [Mezentsev 2005].

However, exceptionally weak signals are usually not only attenuated but also delayed. They arrive at the receiving antenna indirectly e.g. after reflection or diffraction, rather than along the line-of-sight [Wieser 2006]. The associated range errors are significantly larger than the typical errors of line-of-sight observations. So, the signals additionally provided by an HS receiver are usually less accurate than those which can also be tracked with a conventional GPS receiver.

**INS**

The inertial navigation system (INS) is a self-contained navigation system assembling inertial sensors, which automatically provides the user's position, heading, and velocity. The principle is based on the measurement of rotational and linear movements without reference to external coordinates. These movements can be measured by gyroscopes and accelerometers, which are the two main types of sensors.

The gyroscope (or gyro) measure rotational values without reference to external coordinates. Most gyros measure the speed of rotation (also known as "rates") in one axis and are known as "single axis gyros". Speed of rotation is normally measured in units of degree per second or hour. A gyro's accuracy and performance is connected to its output errors: bias, scale factor error and noise. The bias (degree per second or hour) is the offset error output, which can be measured when the sensor is static [GeneSys press 1997]. The scale factor error is the linear deviation of the measured rate from the true rate (normally given as a percentage of full scale). Noise is highly important because, after signal integration, the noise results in non-deterministic behavior (known as "Random-Walk" behavior). It should be noted that all these errors are temperature dependant.

The accelerometers measure the linear acceleration which represents the actual dynamics of a moving body. This measurement is integrated once to receive the velocity and twice to receive the position. Following double integration so as to obtain position, the accelerometer bias gives rise to a positional drift error raised to the power of 2. The signals with their drifts are transformed and so the drifts are also presented in the reference co-ordinate system. This means that the errors of orientation and position are increasing proportionally with the elapsed time. The accelerometer accuracy is limited by several factors: bias, linear and non-linear scale factors and non-orthogonality of the sensors [Legat 2002].

Often, the inertial sensors are mounted orthogonally in a measurement unit with respect to a defined coordinate frame. The sensors can detect acceleration and angular rate in each one of the directions [Gabaglio 2002]. An example is the Dead Reckoning Compass (DRC) developed by Vectronix AG, which uses accelerometers and magnetometers to deliver a continuous 2D position.

The inertial sensors used in the pedestrian navigation are based on MEMS technology. They are assembled in Inertial Measurement Unit (IMU) and output the user's position based on the principle of Dead Reckoning (DR). Dead reckoning is the continued application of the first principle task of geodesy calculating each new position based on previous position information. With known starting coordinates and the transformed sensor signals true orientation and position can then be calculated. The IMU's are often coupled with other systems like GPS in order to calibrate the inertial sensors [Ladetto et al. 2001].

**Wireless positioning**

Positioning by diffusion is a method by which mobile and static devices can calculate their positions, by means of exchanging information over short range wireless links [Köbben et al. 2006]. There are two methods for providing wireless position location information: handset-based solutions and network-based solutions. Handset-based solutions make use of the Global Positioning System (GPS), discussed above.

Network-based solutions rely on the signal transmitted from the wireless handset and received at multiple fixed base stations (Fig. 2.12). That category of positioning systems makes use of existing communication network infrastructures, such as GSM and Wireless LAN (WLAN) [Wang et al. 2007].

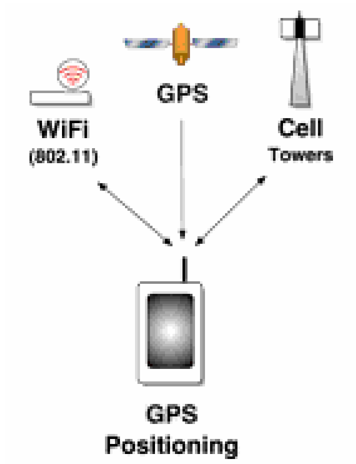



Fig. 2.12 Basic structure of a wireless network.

Basically, a wireless local-positioning system consists of at least two separate hardware components: a measuring unit that usually carries the major part of the system “intelligence” and a signal transmitter. The transmitter in the simplest case is just a beacon [Vossiek et al. 2003]. Mainly, three different measurement principles are used today: angle-of-arrival (AOA), received signal strength (RSS), and propagation time based systems that can further be divided into three different subclasses: time-of-arrival (TOA), roundtrip-time-of-flight (RTOF) and time-difference-of-arrival (TDOA).

Network-based solutions face a number of difficulties, including multipath propagation, diffraction, weak signal conditions, base station availability and expensive upgrades [Qualcomm press 2003].

Another technology that became very popular is the ultra-wideband (UWB). It is a radio technology which sends low energy pulses spread across a wide spectrum of frequencies, enabling high data rate communications and high precision location tracking [Evennou 2007]. Early UWB applications include the position location in radar systems. Recently UWB has drawn significant attention from communication researchers and industry [Yu et al. 2004]. Systems for advanced real-time location based on UWB technology are developed [Ubisense press 2007].

In spite of their positioning accuracy, the wireless positioning systems are not considered further in this thesis, because they are very expensive and, like the AGPS, they do not allow for autonomous positioning.

**LBS**

Location-based services (LBS) become very popular these days [Wierenga et al. 2005]. LBS are offered by some cell phone networks as a way to deliver geographical information to the cell phone users based on their current location.

The technology needed to provide LBS is referred to the following components [Magon et al. 2001]:

- The location or the position of the user;
- The geographic data of that location;
- The application to process the location information along with the geographic data.

The location of the user can be estimated using several methods:

- GPS, for the cell phones with integrated GPS chip;
- Network based methods like Cell of Origin, Time of Arrival, Angle of Arrival, AGPS;
- Manually, by calling dedicated phone number like E911 (in US) or E112 (in Europe).

The second component, the geographic data, contains spatial information like the road network, geocoded customer addresses, buildings and different points of interest. The quality of the needed geographic data is dependent on the type of service provided to the user.

The last component, the application that delivers the information to the users, is controlled by the cell phone provider in the so called centre of data management. The transmission of the information is made via radio channels, SMS or GSM.

Normally, LBS services use a single base station to determine a phone's location with maximal accuracy of about 100 m which is largely insufficient for the indoor personal navigation purposes. The autonomy limitations mentioned above are valid here as well.

**Other methods**

There is a variety of methods for personal positioning that use images taken from a wearable camera [Kourogi et al. 2003], [Aoki et al. 1997] or Audio and video signals [Checka et al. 2001] as supplement data to other positioning systems. These methods will not be discussed here.

### 2.2.2. Sensor fusion

A positioning system may consult many different positioning sensors to determine the position of a mobile device. In many cases, though, sensor data is noisy and influenced by the environment. Even worse, position estimates from different sensors may conflict with each other [King et al. 2005]. For example the satellite positioning with GNSS (GPS, GLONASS etc.) does not work under all environmental conditions (e.g. in urban canyons with no satellite visibility and indoor). A combination and integration with other sensors (e.g. dead reckoning sensors, inertial navigation systems (INS), cellular phone positioning, etc.) is essential.

For pedestrian navigation systems, suitable location technologies include GPS/GNSS and indoor location techniques, dead reckoning sensors (e.g. magnetic compass, gyros and accelerometers) for measurement of heading and travelled distance as well as barometric pressure sensors for altitude determination [Retscher, 2004].

Sensor fusion is a method for integrating data provided by various sensors, in order to obtain the best estimate for the states of a dynamic system. That fusion may implicate not only sensors, but different types of methods, like map aiding [Legat 2000]. The sensor fusion manages the redundancy in the navigation information. On the other hand, the combining of data from various sensors improves the positioning accuracy and reliability.

Following that paradigm several integrated pedestrian navigation systems have been developed: NAVIO [Retscher, 2006], MobileNav [Legat 2002], PNM [Ladetto 2002].

### 2.2.3. Autonomous navigation system

An autonomous navigation system (ANS) is a system that provides precise and reliable positioning without relying on external data input [Ladetto 2003]. The external data consists in measurements using infrastructures such as GPS and WLAN. We have shown that in terms of precision or availability these systems are inefficient for the needs of indoor personal navigation. The use of personal ANS would set the user independent from the availability and drawbacks of the external positioning systems.

The only positioning method that allows this autonomy can be provided by an inertial navigation system (INS). Such system has its own power supply and can be easily carried by the user [Macheiner 2004]. As additional input to the system, a digital map database of the region of interest can be introduced. That input is not in conflict with the autonomy requirement.

## 2.3. Map matching

The combination of different positioning systems (sensor fusion) assures redundancy in the positioning process. The introduction of a georeferenced map database and involving information from it in the positioning estimation will make the process more reliable. The map database and the positioning systems have been presented in the previous chapters. Here we discuss the possible fusion of information from both data sources.

### 2.3.1. Concept of map matching

Map matching algorithms are often used to determine the location of a vehicle on a road. Most of the formulated algorithms utilise navigation data from GPS or GPS/DR and road network data from a digital map. The fact that vehicles are generally constrained to a finite network of roads provides computer algorithms with digital information that can be used to correlate the computed vehicle location with the road network [Taylor et al. 1999].

Many frameworks have been developed resulting in a map-aided estimation process that takes into account the measurement noise statistics to optimally translate the raw position measurements onto the road network [Scott 1994]. Map matching based methods are designed to support the real-time navigational function for advanced car driving assistance [Syed et al. 2004].

The influence of different factors on the sensors of a positioning system results in erroneous position estimation. Problems like multipath and drifts in the inertial sensors are considered as unavoidable. Instead, these errors could be compensated via an interaction with the map database. Suitable algorithms are based on the association of the measured position with the street network. The application of map matching algorithms increases the accuracy of the position. The results of such algorithms serve to compute corrections for the recalibration of the sensors of the positioning system.

**Definitions**

Consider the graph representation of the street network of some region of interest (refer to chapter 2.1.3.). We can define the graph as $G(V,E)$, where $V$ is the set of vertexes (nodes) and $E$ is the set of the edges (links). In order to simplify the exposition, we assume that there is a perfect correspondence between the elements in $G$ and the streets of the network. This assumption can easily be relaxed, however [Bernstein et al. 1998]. Consider a user (vehicle or person) equipped with a positioning system and moving along the street network. His trajectory is presented as a sequence of points, referred to as *raw positions*.

We note the raw position at moment $t$ as $P_t$ and the task is to determine the corresponding matched position, noted as $\hat{P}_t$ (Fig. 2.13). The map matching is a process of association of one or several raw positions to the elements of the graph. The inputs to the process are raw measurements from the positioning systems and information from the map database. The main output of the process is the matched position. However, additional information (e.g. ID of the chosen graph element) can be presented as well.

The existing map matching techniques consider mainly the association of the user's raw position to the road network. In certain specific applications (flight navigation) the relief information on the map is associated with radar measurements from the airplane, applying the so called terrain elevation matching. Here we will discuss the conventional map matching techniques applied to the road matching.

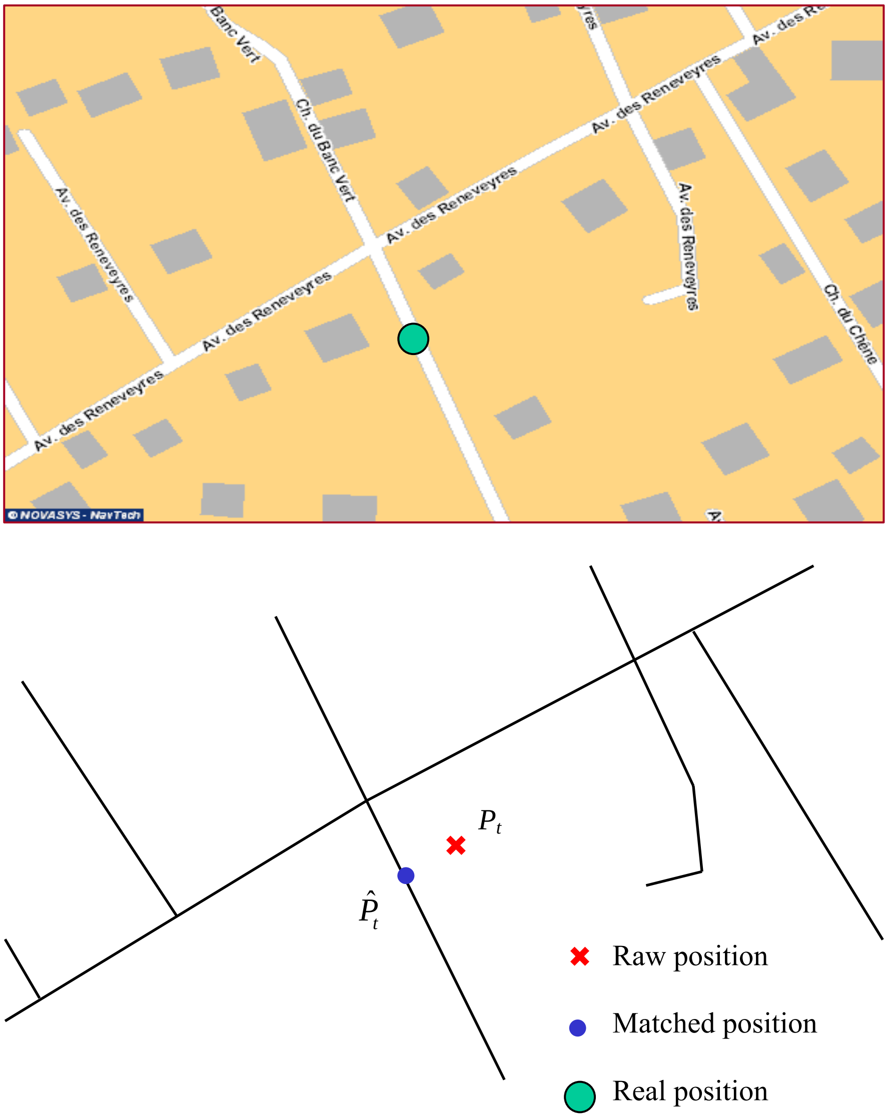



Fig. 2.13 Problematic of map matching

The approaches for map matching algorithms found in the literature can be categorised into three groups: geometric, topological and advanced [Quddus 2006]. The following sections briefly describe these algorithms.

**Existing techniques**

A geometric map matching algorithm makes use of the geometric information of the digital road network by considering only the shape of the links [Greenfeld 2002]. It does not consider the way links are connected to each other. This approach is based on a simple search technique where the raw position is matched to the closest element. We distinguish two techniques: *point-to-point* matching and *point-to-curve* matching.

The point-to-point matching associates the raw position to the closest node of the graph. The closest node is found after comparison of the Euclidean distances between the raw position and each node of the graph. The Euclidean distance between two points $x = (x_1,\ldots,x_n)$ and $y = (y_1,\ldots,y_n)$ in Euclidean $n$-space is computed as (2.1).

$$\sqrt{(x_1 - y_1)^2 + (x_2 - y_2)^2 + \ldots + (x_n - y_n)^2} = \sqrt{\sum_{i=1}^{n}(x_i - y_i)^2} \qquad (2.1)$$

This technique can be easily implemented and provides fast computation. However it is very sensitive to the graph geometry and can give incorrect results. On figure 2.14 a set of points is matched using the point-to-point technique.

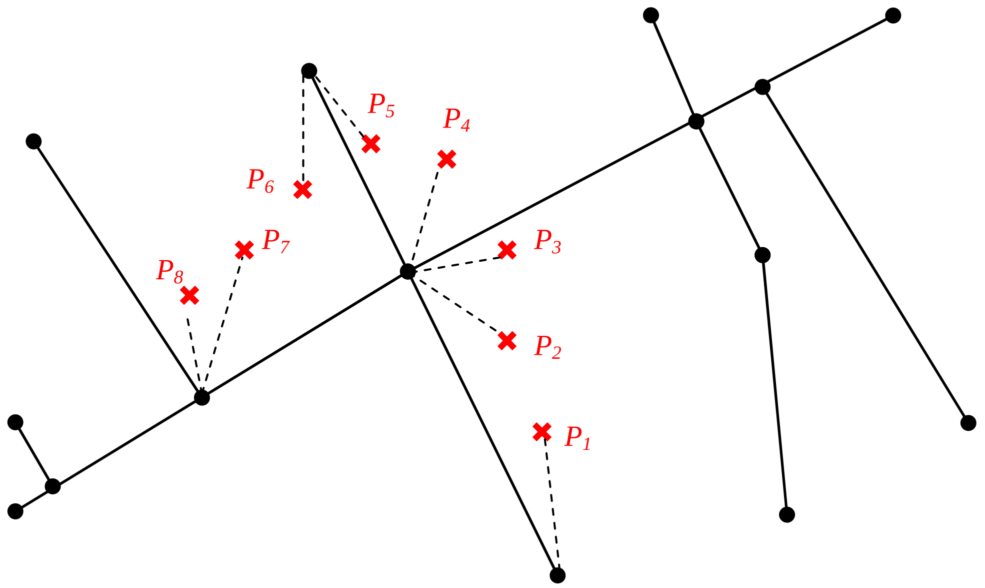



Fig. 2.14 Set of raw positions matched to the closest node

One drawback of this technique is that the raw positions are matched always to a node, even if they are rather closer to the middle of a link of the graph (e.g. $P_1$ and $P_2$ on Fig. 2.14). On the other hand the movement of the user is not reflected realistically. On the example of the same figure the nodal matching of $P_5$, $P_6$, $P_7$, and $P_8$ is topologically incorrect, since the chosen nodes belong to different links and are not directly connected.

The next matching technique is point-to-curve, sometimes called *point-to-link*. This technique associates the raw position to the closest link of the graph. In that case the notation of proximity is referred to the distance from a point to link (2.2).

$$c = \sqrt{\frac{\left(\left(y_1 - y_2\right)^2 x_p + \left(x_2 - x_1\right)^2 y_p + \left(x_1 y_2 - x_2 y_1\right)\right)^2}{\left(y_1 - y_2\right)^2 + \left(x_2 - x_1\right)^2}} \tag{2.2}$$

In (2.2) $c$ is the distance from the point $P$ to a link defined by its end points 1 and 2. The distance is calculated from the raw position to each of the links of the graph. The link which gives the smallest distance is selected and the raw position is matched to it. The matching consists in projecting the raw position on the chosen link (Fig. 2.15).

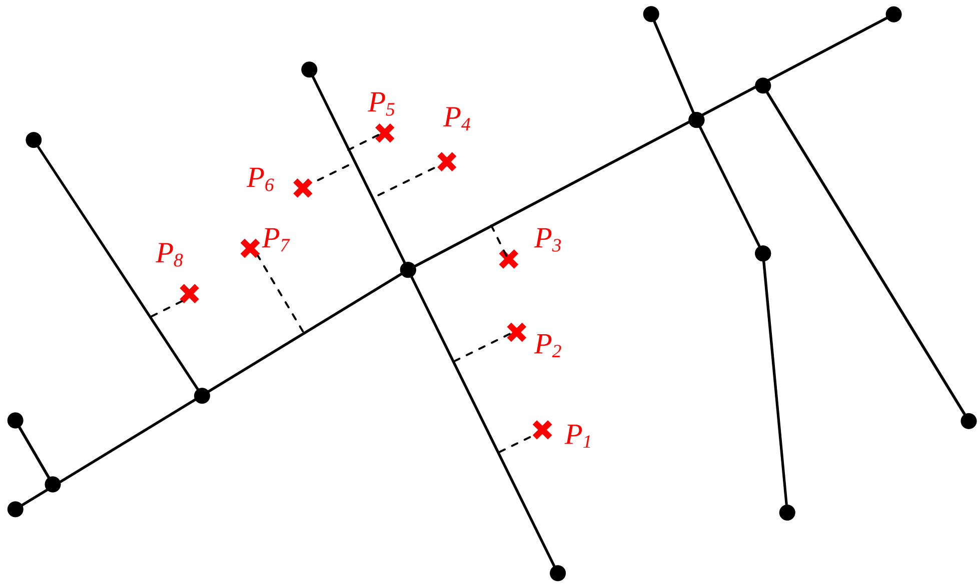



Fig. 2.15 Set of raw positions matched to the closest link

The point-to-curve matching technique gives better results than the point-to-point technique. However, it presents some problems. It is very inaccurate in an urban environment with a dense street network. On the other hand the closest link is not necessary the correct one (e.g. $P_3$ on Fig 2.15).

Another matching technique is *curve-to-curve*, presented by Bernstein and Kornhauser in 1998. The user's trajectory and a sequence of candidate nodes are defined as piecewise linear curves. Then, the distance between these curves is determined applying point-to-point matching. The aim is to find the curve which is closest to the user's trajectory. This matching technique is dependent on the drawbacks of the point-to-point technique and is sensitive to outliers.

Several other matching techniques are described in [White et al. 2000], [Taylor et al., 2001], [Bouju et al., 2002], [Srinivasan et al. 2003] and [Ochieng et al. 2004].

The above matching techniques rely on geometric information only. However, the implication of topological information in the matching process is crucial for the development of any map matching algorithm. The topology presents another type of information, different from the raw measurements. In the literature both types of information are combined mainly in a dedicated weighting system [Greenfeld et al., 2002]. Such weighting system computes specific weights for the candidate elements of the graph. Then the raw position is matched to the element with the highest weight.

### 2.3.2. Map matching for vehicle navigation

Actually, map matching algorithms are developed mainly for the needs of vehicle navigation and partially for the localization of mobile robots [Lee et al. 2003]. Here we discuss some of the algorithms applied for vehicle navigation.

**Constraints**

The common constraint in the vehicle navigation is that the vehicle is travelling on the road. That road is part of a finite road network, so we can assume that the real position of the vehicle is on the network at any moment. In particular cases like public transport localization, the road network is limited to the elements of the bus line. That reduction of the size of the road network simplifies the localization process (Fig. 2.16).

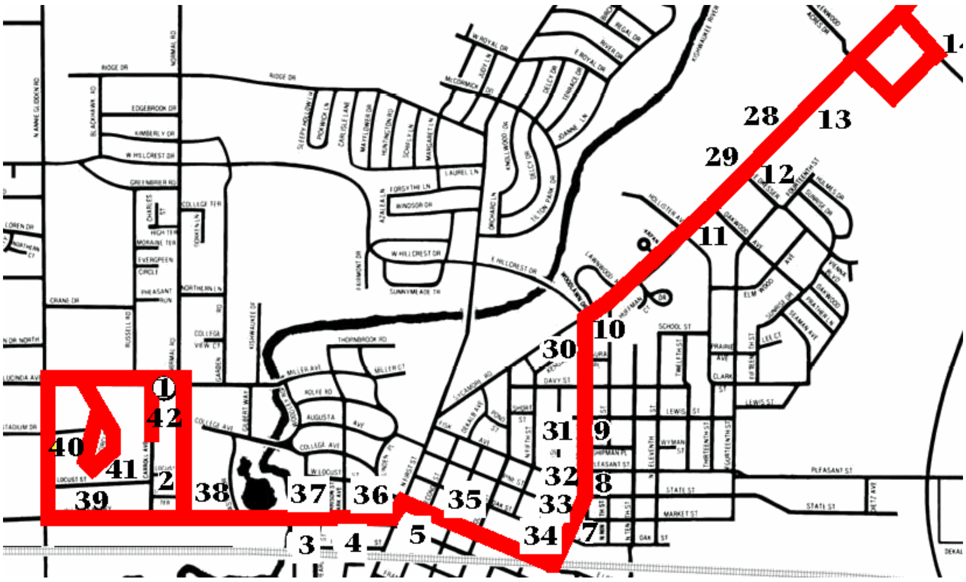



Fig. 2.16 Bus line on city street network

Another constrain is that usually only the planimetric movement is considered and the graph representation of the network is defined in 2D.

**Algorithms for real time map matching**

A weighted topological algorithm is proposed by Greenfeld in 2002. It is based on topological analysis and it uses only coordinate information for the observed position of the user. It assumes no knowledge of the expected travelling route and it does not use any GPS determined heading and/or speed information.

The map matching procedure is composed of two separate algorithms. The first algorithm is called Initial Mapping. The initial match is needed to locate the user somewhere on the network. Based on this initial match, subsequent topological analysis is performed. The second map matching algorithm is applied only after an initial match was found. It is the main matching algorithm that uses topological reasoning and a weighting scheme. The weighting system evaluates several candidate arcs for a correct match by computing a likelihood score based on the different criteria. These criteria are: the perpendicular distance of the GPS point from the arc segment, the degree of parallelism between the GPS line and the street network arc and the intersecting angle if an intersection exists. The last criterion is valid for the case of the GPS line *P1-P2* (Fig. 2.17), where an intersection with the arc exists in the limits of both lines.

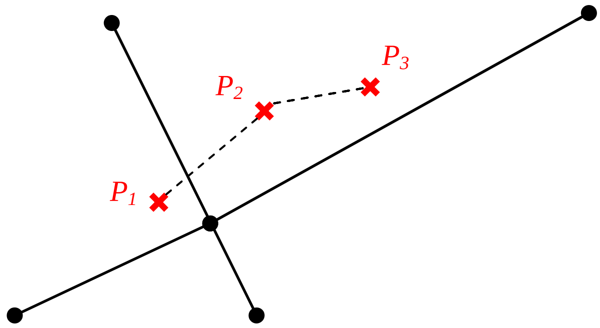



Fig. 2.17 GPS line *P1-P2* intersecting an arc of the network

The total weighting score for a particular link is given by:

$$W = W_D + W_{AZ} + W_I$$

where,

$W$ - the total score

$W_D$ - the weight for proximity

$W_{AZ}$ – the weight for similarity in orientation

$W_I$ – the weight for an intersection, if it exists

The algorithm shows sensitivity to outliers as these can make the determined vehicle heading inaccurate [Greenfeld 2002].

Fuzzy logic, based on fuzzy reasoning concepts, is one of the most widely used soft computational methods. In many circumstances, it can take noisy, imprecise input, to yield crisp (i.e. numerically accurate) output. Fuzzy logic can be applied effectively to map match the output from a HS GPS receiver in urban canyons because of its inherent tolerance to imprecise inputs [Syed et al. 2004].

In 2004 Syed and Cannon describe a map matching algorithm based on a fuzzy logic model using GPS/DR data. The algorithm consists of two sub-algorithms: First fix mode and Tracking mode. In the first sub-algorithm, a fuzzy inference system (FIS) is used to identify the correct link for the initial position fix. The characteristics of this FIS are to select a set of links which are within 50 m of the GPS/DR position fix. A link is then identified based on the direction of the vehicle relative to the direction of the links and the change in heading from the gyroscope. The location of the vehicle is then determined by an orthogonal projection of the position fix onto that link. Following the identification of the first link and the location of the vehicle on it, the algorithm then goes into the second sub-algorithm. Another FIS is used to see whether the subsequent position fixes can be matched to the link identified in the first fix mode. The inputs are proximity, orientation, and distance travelled by the vehicle along the link. If there is any outlier in the GPS/DR output or if a turn is detected, the algorithm goes back to the first fix mode. The reliability of the proposed algorithm is far better as compared to conventional algorithms based on a geometric approach. The algorithm shows a position accuracy of 16m.

A fuzzy logic approach is applied as well by Quddus in 2006. He develops an improved fuzzy logic algorithm that takes into account: the speed of the vehicle, the connectivity among road links, the horizontal dilution of precision (HDOP), and the position of a fix relative to a candidate link.

He develops an optimal estimation technique taking into account the error sources associated with the navigation sensors and the digital map data to determine the location of the vehicle on a link. The MM algorithm developed in this article has two stages: (1) the identification of the correct link and (2) the determination of the vehicle location on the selected link. The algorithm identified 99.2% of the road segments correctly with a horizontal accuracy of 5.5m.

**Algorithms for post treatment and analysis**

Map matching algorithms are commonly dedicated to real time navigational purposes. However there are domains like travel behaviour analysis, where map matching is required as well. The development of such an algorithm is presented in 2004 by Marchal, Hackney and Axhausen.

They analyse the spatial displacements of travellers based on their past trajectory. The developed map matching algorithm aims at identifying the routes taken by the traveller. Large data sets of GPS points (2.5 millions) are recorded on data loggers and used as measurement data. The authors discuss the efficiency of the algorithm in term of accuracy and computational speed using maps with different resolution.

### 2.3.3. Map matching for pedestrian navigation

Nowadays, the localization of pedestrians is possible thanks to the invention of the miniature GPS receivers. These small devices provide sufficient accuracy in zones with an open sky. However, computing the person's position on the street network is far less obvious.

The road map databases are actually dedicated for vehicle navigation applications. The integration of information useful to pedestrians, like pedestrian zones, detailed plans of administrative and commercial buildings, etc. will allow the execution of specific navigation queries [Rouiller 2002].

In urban canyons and indoors pedestrian positioning meets problems of muti-path, drifts in the INS sensors, and other problems that decrease the accuracy of the localization and accumulate localization errors. Therefore, map matching algorithms for pedestrian positioning and navigation must be developed and applied.

**Liberty of movement**

The movement of a person is much more difficult to modelize than the movement of a vehicle. The pedestrian enjoys more freedom in his movements than a vehicle. In a few seconds, he can move to the opposite pavement, enter a shopping mall or climb up stairs. Moreover these examples refer to the normal walk of the person. We can imagine more sophisticated movements like lateral movement, jumping, back steps, etc.

A special case in pedestrian displacement is the possibility for vertical movement. We refer to the use of elevators, staircases and escalators. Contrary to the vehicle navigation, where the vehicle is supposed to move in a 2D plan, for pedestrian navigation we have to consider these movements. This imposes the addition of new elements in the map database, i.e. vertical links (refer to chapter 2.1). Another example of liberty of movement is that the person can walk in both directions of the street or corridor. Thus in the graph representation the links are defined as bidirectional, except for certain links with an access restriction.

**Autonomy**

In chapter 2.2 we have discussed the autonomy aspect for the navigation process. It is valid for pedestrian navigation applications as well. The equipment that assures that autonomy is based on a combination of inertial measurement unit (IMU), map database, and advanced positioning algorithms (Fig. 2.18).

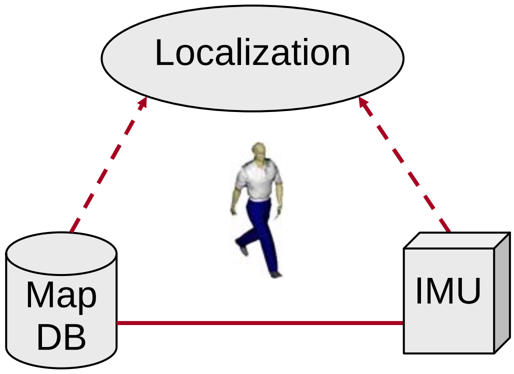



Fig. 2.18 Schema of the equipment for autonomous positioning

A person equipped with such a system (Fig. 2.18) is totally independent from the performance and drawbacks of the external positioning systems. Another question of autonomy can be the type of INS and computing unit. There is no interest to overcharge the user with sensors and devices, because this could present certain limitations as well.

**Required accuracy**

The positioning precision requirements for pedestrian navigation depend mainly on the application. In 1999 in the US a requirement was imposed to mobile phone service providers to locate emergency callers (E911) with an accuracy specified in Table 2.1 [Zagami et al. 1999].

| Type | Accuracy (m) | Reliability (%) | Accuracy (m) | Reliability (%) |
|---|---|---|---|---|
| Network based solutions | 100 | 67 | 300 | 95 |
| Handset based solutions | 50 | 67 | 150* | 95* |

Table 2.1 Accuracy requirements for mobile phone service providers
(* - with the use of GPS).

A big variety of methods for personal positioning has been developed. Based on a combination of different positioning techniques they give sufficient accuracy.

In 2004 Lachapelle presents a method based on HSGPS technology. An accuracy of 25m is achieved for indoor positioning. Another method using GPS/RFID/DR integration indoor positioning with 2-6m accuracy is presented in 2006 by Kourogi.

For autonomous pedestrian navigation systems the positioning accuracy is presented as percentage of the travelled distance. In 2003 Ladetto presents a new autonomous system – the

Personal Navigation Module (PNM). The PNM contains several sensors: a GPS receiver, three magnetic field sensors, three accelerometers, a gyroscope and a barometer. It assures positioning accuracy of 5% of travelled distance, representing the difference between the effective and predicted distance [Ladetto et al. 2000].

The personal positioning via map matching algorithms will result in computation of matched position of the user. Thus the accuracy can be presented in meters instead of percentage.

To define what accuracy we need for the indoor autonomous positioning we should take a close view on the constructed environment, reflected by the graph. In administrative buildings often several rooms can be placed in a relatively small area (Fig. 2.19).

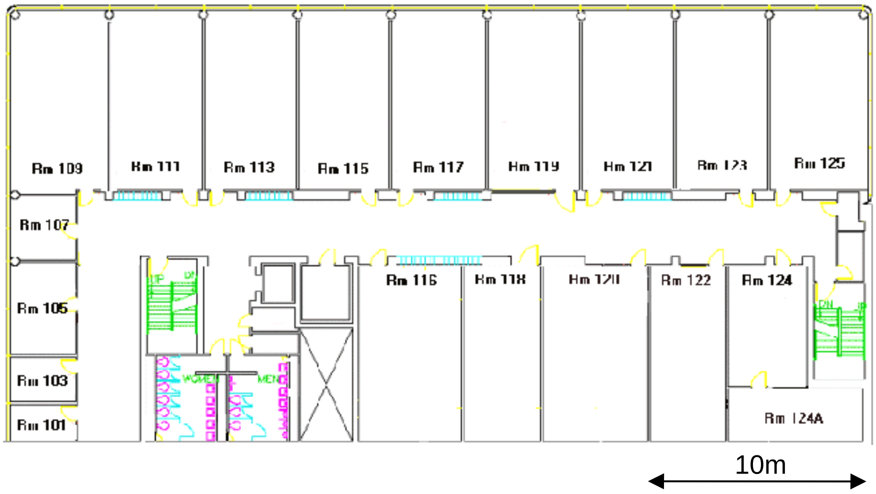



Fig. 2.19 Floor of hospital, consisting of several offices

The positioning must be sufficiently accurate to determine at least in which room the user is. Intuitively speaking, a metric accuracy is reasonable for that kind of applications.

# Bibliography

Abwerzger, G., Hofmann-Wellenhof, B., Ott, B., Wasle, E., (2004) GPS/SBAS and Additional Sensor Integration for Pedestrian Applications in Difficult Environments, TeleConsult Austria and European Space Agency (contract number 17545/03/NL/GS).

Aoki, Hisashi, Schiele, Bernt and Pentland, Alex (1997), Recognizing Personal Location from Video, MIT Media Laboratory 20 Ames St., Cambridge, MA 02139

Bernstein, D. and Kornhauser, A. (1998). Map matching for personal navigation assistants. Research Report. Operations Research and Financial Engineering, Princeton University, Princeton, NJ.

Bouju, A., Stockus, A., Bertrand, F., Boursier, P. (2002) Location-based spatial data management in navigation systems, IEEE Symposium on Intelligent Vehicle, 1

Büchel, D. (2003) Méthodes de guidage applicables au plan d'orientation de l'EPFL, Travail de diplôme, Laboratoire de topométrie, EPFL, Suisse

Cass, Raymond A. (1992), Building Navigable Databases for the Real World, European Geographical Technologies, VEHICLE NAVIGATION I INFORMATION SYSTEMS, IEEE

Chandy K. M. and Misra J. (1982) Distributed Computation on Graphs: Shortest Path Algorithms, Programming Techniques and Data structures, v.25

Checka, Neal, Wilson, Kevin (2001), Person Tracking Using Audio-Video Sensor Fusion, MIT Artificial Intelligence Laboratory, Cambridge MA, 02139 USA

Chen, Jing-Chao (2003), Journal of Formalized Mathematics, Volume 15, 2003 University of Bialystok, 2003

Claussen, H., Lichtner, W., Heres, L., Lahaije, P., Siebold, J. (1988) GDF, A PROPOSED STANDARD FOR DIGITAL ROAD MAPS

Claussen, H., Lichtner, W., Heres, L., Lahaije, P., Siebold, J. (1989), GDF, a proposed tandard for digital road maps to be used in car navigation systems, Vehicle Navigation and Information Systems Conference, 11-13 Sep 1989, Toronto, Canada

Claussen, Hinrich (1993), Status and Directions of Digital Map Databases in Europe, IEEE - IEE Vehicle Navigation & Information Systems Conference, Ottawa - VNlS '93

Dana, Peter H. (1994), Department of Geography, University of Texas at Austin, 1994

Dijkstra, E. W. (1959), A note on two problems in connection with graphs. \em Numer. Math., 1:269--271, 1959

Evennou, Frédéric (2007), Techniques et technologies de localisation avancées pour terminaux mobiles dans les environnements indoor, PhD thesis, UNIVERSITE JOSEPH FOURIER - GRENOBLE I

Gabaglio, Vincent (2002), GPS/INS Integration for pedestrian navigation, Thesis N° 2704, EPFL ; (2002)

GeneSys Elektronik GmbH (1997) Inertial Sensors and Systems An Introduction, The Engineering Department, GeneSys Elektronik GmbH, July 1997

Gillieron, P.-Y., Büchel, D., Spassov, I., Merminod, B. (2004) Indoor Navigation Performance Analysis, Proceedings of the 8th European Navigation Conference GNSS 2004, 17-19 May, Rotterdam, The Netherlands.

Global Locate (2003) A-GPS technology overview, Press Release, San Jose, California, March 17, 2003

Greenfeld, J.S. (2002) Matching GPS observations to locations on a digital map. In proceedings of the 81st Annual Meeting of the Transportation Research Board, January, Washington D.C.

Greenfeld, Joshua (1998), Digital map requirements for automatic vehicle location, Department of Civil & Environmental Engineering, New Jersey Institute of Technology

Hoffmann-Wellenhof, B., Lichtenegger, H. and Collins, J. (1994) GPS: Theory and Practice. 3rd ed.New York: Springer-Verlag.

King, Thomas, Kopf, Stephan, Effelsberg, Wolfgang (2005), A Location System based on Sensor Fusion: Research Areas and Software Architecture, Praktische Informatik IV, University of Mannheim

Kobben, B., van Bunningen, A., Muthukrishnan, K. (2006) Wireless Campus LBS, Springer, 2006. (Lecture Notes in Geoinformation and Cartography) ISBN: 978-3-540-34237-3. pp. 399-408

Kourogi, Masakatsu, Kurata, Takeshi (2003), A method of personal positioning based on sensor data fusion of wearable camera and self-contained sensors, IEEE Conference on Multisensor Fusion and Integration for Intelligent Systems 2003

Kourogi, Masakatsu, Sakata, Nobuchika, Okuma, Takashi, and Kurata, Takeshi (2006), Indoor/Outdoor Pedestrian Navigation with an Embedded GPS/RFID/Self-contained Sensor System, National Institute of Advanced Industrial Science and Technology (AIST), Umezono 1–1–1, Central 2, Tsukuba, Ibaraki, 305–8568, Japan

Lachapelle, G., Mezentsev, O., Collin, J. and MacGougan, G. (2003) Pedestrian and vehicular navigation under signal masking using integrated HSGPS and self contained sensor technologies, 11th World Congress, International Association of Institutes of Navigation, Berlin, 21-24 October 2003

Lachapelle, Gérard (2004), GNSS Indoor Location Technologies, Journal of Global Positioning Systems (2004) Vol. 3, No. 1-2: 2-11

Ladetto, Q. (2003) Capteurs et Algorithmes pour la Localisation Autonome en Mode Pédestre, Phd Thesis, École Polytechnique Fédérale de Lausanne, 2003.

Ladetto, Quentin, Gabaglio, Vincent, Merminod, Bertrand, Terrier, Philippe, Schutz, Yves (2000), Human Walking Analysis Assisted by DGPS, GNSS 2000

Ladetto, Quentin, Gabaglio, Vincent, Merminod, Bertrand (2001), Combining Gyroscopes, Magnetic Compass and GPS for Pedestrian Navigation, Geodetic Engineering Laboratory, EPFL

LaMance, J., DeSalas, J., Jarvinen, J. (2002) Assisted GPS: A Low-Infrastructure Approach, GPS World, Mar 1, 2002

Lee, Dongheui, Chung, Woojin, Kim, Munsang (2003), A Reliable Position Estimation Method of the Service Robot by Map Matching, Proceedings of the 2003 IEEE, Taipei, Taiwan

Legat, K., Lechner, W. (2000), Navigation Systems for Pedestrians - A Basis for Various Value-Added Services, ION GPS 2000, 19-22 September 2000, Salt Lake City, UT

Legat, K., Lechner, W., (2000) Navigation systems for pedestrians - a basis for various value-added services, ION GPS 2000, 19-22 September 2000, Salt Lake City, UT

Legat, Klaus (2002), Pedestrian navigation, Graz University of Technology, April 2002

Li, B., Salter, J., Dempster, A., Rizos, C. (2005) Indoor Positioning Techniques Based on Wireless LAN, UNSW, Sydney, 2052, Australia

Macheiner, K. (2004) Performance Analysis of a Commercial Multi-Sensor Pedestrian Navigation System, Master Thesis, Institute of Navigation and Satellite Geodesy, Graz University of Technology, Austria

Magon A., Shukla R., (2001), LBS, the ingredients and the alternatives, RMSI A-7, Sector – 16, Noida-201 301, UP, India, Asian GPS Conference 2001

Marchal, F., Hackney, J. and Axhausen, K.W. (2004). Efficient map-matching of large GPS data sets - Tests on a speed monitoring experiment in Zurich. Arbeitsbericht Verkehrs- und Raumplanung, Institut f¨ur Verkehrsplanung und Transportsysteme, ETH Zurich, Zurich

Mayr, H. (2006) Model-based Navigation Using GPS:One Step Closer to Intelligent, Incremental Maps, Department of Software Engineering, Upper Austria University of Applied Sciences, Hagenberg, Austria

Mezentsev, Oleg (2005), Sensor Aiding of HSGPS Pedestrian Navigation, UCGE Reports, Number 20212

NAVSTAR GPS User Equipment Introduction. (1996) Available on line from United States Coast Guard Navigation Center

Ochieng, W.Y., Quddus, M.A., Noland, R.B. (2004) Integrated positioning algorithms for transport telematics applications. In proceedings of the Institute of Navigation (ION) annual conference, 20-24 September, California, USA.

Parkinson, Bradford W. and Spilker, James J. (1996) Global Positioning System: Theory and Practice. Volumes I and II. Washington, DC: American Institute of Aeronautics and Astronautics, Inc.

Qualcomm press (2003), Hybrid Position Location Technology White Paper

Quddus, M. A. (2006). High integrity map matching algorithms for advanced transport telematics applications. PhD diss., Centre for Transport Studies, Imperial College London.

Retscher G. (2004) Multi-sensor Systems for Pedestrian Navigation, Proceedings of the ION GNSS Meeting, 21-24 September, Long Beach, California, U.S.A.

Retscher G. (2006) An Intelligent Multi-sensor System for Pedestrian Navigation, Journal of Global Positioning Systems (2006) Vol. 5, No. 1-2:110-118

Rodrigue, Jean-Paul (2007), Network Data Models, Dept. of Economics & Geography, Hofstra University

Scott, C., (1994), Improving GPS Positioning for Motor-Vehicle through map matching, Proceedings of ION GPS-94, Salt Lake City, Utah

Srinivasan, D., Cheu, R.L. (2003) Development of an improved ERP system using GPS and AI techniques, IEEE Proceedings on Intelligent Transportation Systems, 554-559.

Stephen, Jim (2000) Development Of A Multi-Sensor GNSS Based Vehicle Navigation System, UCGE Reports Number 20140

Syed, S. (2004) GPS Based Map Matching in the Pseudorange Measurement Domain, ION GNSS, Long Beach, CA, September 21-24, 2004

Syed, S., and Cannon, M. E. (2004) Fuzzy logic-based map matching algorithm for vehicle navigation system in urban canyons. Proceedings of the Institute of Navigation (ION) National Technical Meeting, 26–28 January, San Diego, CA.

Taylor, G., Blewitt, G., Steup, D., Corbett, S., Car, A. (2001), Road Reduction Filtering for GPS-GIS Navigation, Transactions in GIS, ISSN 1361-1682, 5(3), 193-207.

Taylor, George and Blewitt, Geoffrey (1999), Virtual Differential GPS & Road Reduction Filtering by Map Matching, ION99

Taylor, George, Car, Adrijana, Mehner, Henny (2006) Experimenting with Hierarchical Wayfinding, Dept. of Geomatics, University of Newcastle upon Tyne

Ubisense press (2007), EU releases Radio Frequency band for ultra-wideband UWB, Cambridge, England, Feb. 22, 2007

Vossiek, Martin, Wiebking, Leif, Gulden, Peter, Wieghardt, Jan, Hoffmann, Clemens, Heide, Patric (2003), Wireless local positioning, Microwave Magazine, IEEE, 2003

Wang, Hui, Lenz, Henning, Szabo, Andrei, Bamberger, Joachim, Hanebeck, Uwe D. (2007), WLAN-Based Pedestrian Tracking Using Particle Filters and Low-Cost MEMS Sensors, Workshop on Positioning, Navigation and Communication, (2007 WPNC), Hannover, Germany, March 2007.

White, C.E., Bernstein, D., Kornhauser, A.L. (2000) Some map matching algorithms for personal navigation assistants. Transportation Research Part C 8, 91-108.

Wieser, Andreas (2006), High-Sensitivity GNSS: The Trade-Off between Availability and Accuracy, 3rd IAG / 12th FIG Symposium, Baden, May 22-24, 2006

Yanagisawa, Hiroki (2006), Fast Shortest Path Computation for Solving the Multicommodity Flow Problem, Computer Science; Mathematics IBM Research, Tokyo Research Laboratory, IBM Japan, Ltd.

Yu, Kegen, Oppermann, Ian (2004), UWB Positioning for Wireless Embedded Networks, Centre for Wireless Communications University of Oulu, Finland2004 IEEE

Zagami, James M., Parl, Steen A., Bussgang, Julian J., Melillo, Karen Devereaux (1998), Providing Universal Location Services Using a Wireless E911 Location Network IEEE Communications Magazine, April 1999

Zweiacker, P. (2003) Système d'information géographique pour navigation pédestre à l'intérieur de bâtiments, Travail de diplôme, Laboratoire de topométrie, EPFL, Suisse

# Chapter 3

# Algorithms for autonomous indoor pedestrian navigation

## 3.1. Choice of instruments

### 3.1.1. Hardware solution (PNM)

The following two pages are dedicated to the pedestrian navigation system chosen in this research, the Personal Navigation Module (PNM) made by Vectronix AG. Developed in a close cooperation with EPFL this module was chosen because of its compactness, ease to use and good performance corresponding to the requirements for autonomy.

The PNM is a locating device for autonomous, stand-alone pedestrian navigation. This module provides a reliable 3D position thanks to a sensor fusion and advanced algorithms [Ladetto 2003]. Considering the factory instructions it needs to be clipped in the center of the back at hip level, i.e. on the user's belt, to acquire optimal results (Fig. 3.1). The control of the PNM and the acquisition of measured data are realized using pocket PC, connected to the module via serial port.

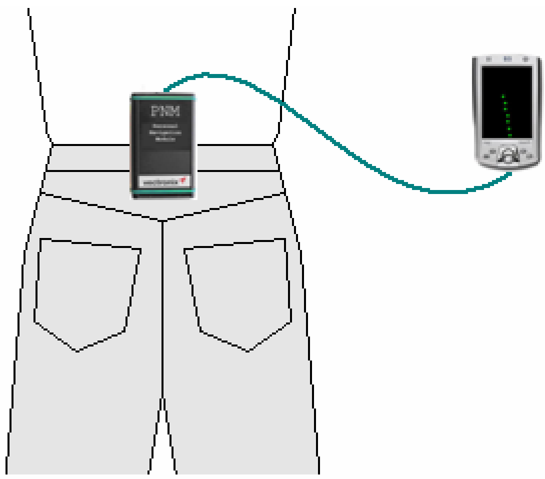



Figure 3.1: The PNM worn on the belt and connected to a pocket PC.

The PNM contains several sensors: a GPS receiver, three magnetic field sensors, three accelerometers, a gyroscope and a barometer (Fig. 3.2).

The principle of 3D positioning is based on the detection of the movement where the discrimination between walk and stop takes central place. The core of the process is the detection of the steps of the user. The event *step* is detected thanks to the vertical accelerometric signal. Then, measurements are registered at each step. The PNM can work in three different modes, GPS mode, INS mode and combined GPS/INS mode.

User's absolute position (*longitude, latitude* and *altitude*) is determined when GPS signals are available. Along with these measurements additional information is acquired, like the number of satellites and the positional precision (HDOP). Thanks to the magnetic field sensors and the gyroscope, the *azimuth* and the *heading* of movement are measured. The *altitude* is determined by atmospheric pressure measurements with the help of the barometer. The three-dimensional accelerometric signal is used to determine the *speed* of movement of the person and to detect possible changes of the direction of walk. Using the clock of the GPS receiver all the measurements are time-stamped and registered on every step.

Alternatively, the person can stop. Then the measurements are registered automatically with a frequency of 2Hz. This is mostly necessary in case of vertical movement like taking elevator when changes in the altitude must be detected.

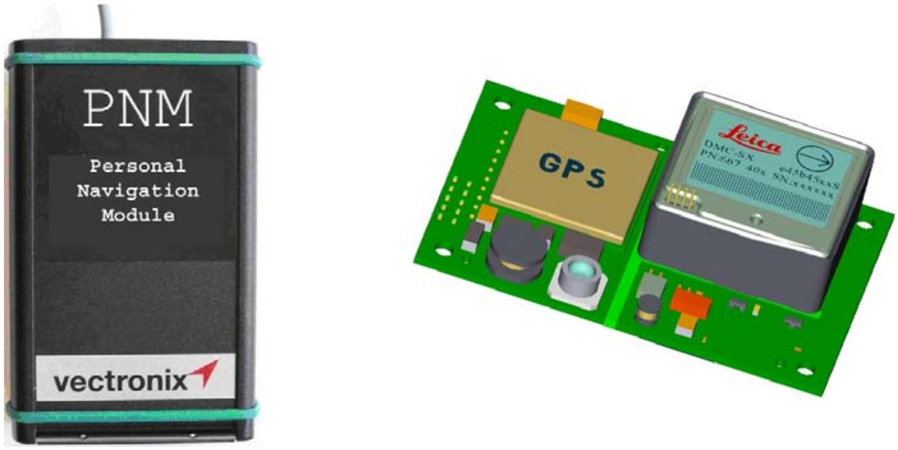



Figure 3.2: The PNM in close view

All the measurements and other information are registered using a standard navigation message *NMEA-0183*. The data is stored in ASCII format on the pocket PC.

In the package of the PNM there is a software application which needs to be installed on the user's pocket PC. The interface allows the user to send commands to the module and to receive information on the PNM performance. Thus the person can decide when begins his trajectory and when it ends, i.e. when to start and when to stop the measurements. Every trajectory is saved on the pocket PC in a separate ASCII file.

### 3.1.2. Measurement message (NMEA-0183)

NMEA is a standard for data transmission developed by the US marines. The information is organized in a predefined "sentence" (message). Most GPS manufacturers include special messages to the standard NMEA set in their products for maintenance and diagnostics purposes. These extended messages are not standardized and are normally different from vendor to vendor.

The PNM supports the following NMEA messages: *RMC* and *GGA*. (See Annex A for details). In the ASCII file which represents the trajectory, every text line corresponds to one user's step and contains both NMEA messages.

### 3.1.3. Software platform

The question of software platform is strongly connected with the performance of the hardware instruments, with the methods of raw data acquisition and with the desired final results.

Using the ASCII files and the database information as input allows treating the data in post treatment mode. Of course, the real time performance of the algorithms for map-matching is of prime importance. However, we could not intervene in to the system of the PNM and make changes of its functionality. Therefore the algorithms in this research are designed to work in real time mode, but are written and tested for post treatment mode. That means for real time implementation of the algorithms one should keep the same methodology and change only the programming language.

Most of the algorithms are developed using *Visual Basic for Applications* (VBA) and the algorithm for continuous localization based on Fréchet distance (3.3.3.) is developed with *MatLab*.

VBA is an interpreted language, meaning its instructions are executed when the source code is run. It is implemented in applications such as AutoCAD, ESRI, ArcGIS and most Microsoft Office applications. VBA programs, or *macros*, can be attached to a menu button, a keyboard shortcut, or an event in the application, such as the opening of the document. The language also provides a user interface in the form of User Forms, which can host ActiveX controls for added functionality. VBA is proprietary to Microsoft and is not an Open standard.

AutoCAD is chosen as a graphic environment, with the possibility to implement the algorithms and to illustrate the tests. This solution permits a flexible interaction with other software products for performance analysis.

### 3.1.4. Data acquisition

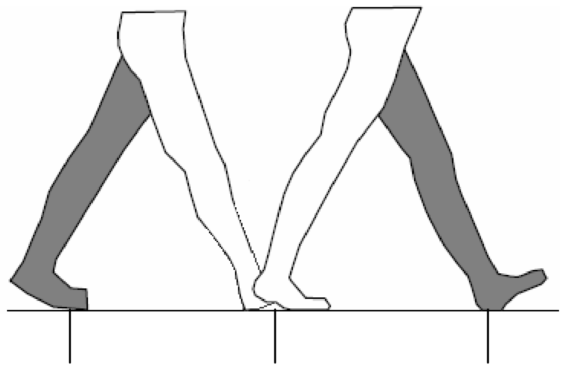
In this study all acquisitions of measurement data, the tests of the algorithms and the analysis of the performance are based on real trajectories. These trajectories are made in the buildings of the campus of EPFL. The ASCII files are transferred from the PDA to a PC and treated in post-treatment mode.

## 3.2. Initial Localization

The localization methods aim at determining the location of the user. Considering the 3D graph representation of the building, we call *initial localization* the technique of finding the edge of the graph occupied by the person and person's orientation on that edge. Two sources of raw data are used - inertial measurements and map database. The core of the process is the association of elements of trajectory to the contents of the graph, i.e. map-matching. The map database is considered as static data. Alternatively, the inertial measurements are considered as dynamic data, since the trajectory is periodically updated. The association of the elements of both sources of data relies on geometric and topologic criteria. In order to apply these criteria the raw data needs to be transformed into *adequate input* to the process of localization. That means the information from the user's trajectory and the map database must be presented in format suitable for the matching process, which is discussed in details further. Therefore, the process passes through a pre-processing phase transforming the trajectory from sequence of points into a 3D polyline. Then, the problem of localization is tackled applying statistical methods.

### 3.2.1. Pre-processing of the raw data

In our approach the first source of raw data are the inertial measurements. The problem of localization could not be solved straight away using this raw data. It needs to be transformed into an adequate input for the process reflecting the geometry of user's trajectory. In order to decrease the complexity of the problem of localization we proceed to a pre-processing step. This step will treat the raw data in order to create an adequate input for the localization process. During the walk of the person we distinguish two types of movements – *basic* and *critical*. The basic movements are the steps. The critical movements characterize the trajectory more globally. Movements like turn, stop and go are defined as critical.



$$\begin{pmatrix} v_{t-1} \\ r_{t-1} \\ h_{t-1} \end{pmatrix} \qquad \begin{pmatrix} v_{t} \\ r_{t} \\ h_{t} \end{pmatrix} \qquad \begin{pmatrix} v_{t+1} \\ r_{t+1} \\ h_{t+1} \end{pmatrix}$$

Figure 3.3: Raw measurements registered on every step

During the walk the determination of each step position is based on the inertial measurements and the previously determined step position (Fig. 3.3). These measurements are:

$v$ - Speed (knots)

$r$ - Heading (degrees)

$h$ - Barometric altitude (meters)

$t$ - Time (hhmmss.sss)

Their values are registered in the NMEA message, transmitted from the navigation module. Special attention must be paid on the speed measurement, because it is dimensioned in knots. In order to transform it in meters per second the value in knots must be multiplied by a constant as following:

$$v\ (\text{m/s}) = v\ (\text{knots}) * 0.5144456$$

In terms of geometry the trajectory can be considered as a sequence of points where the raw data (Speed, Heading, Barometric altitude and Time) is known at each point.

The first step in the pre-processing of the raw data is to define the geometric parameters between the successive step positions. Consider the walking person. Using the time-stamped inertial measurements we can easily compute the length $d_t$ of a stride, the angle $\beta_{t,t-1}$ and the elevation $e_{t,t-1}$ between any consecutive strides at moment $t$ (1).

$$\begin{aligned} d_t &= v_t.\Delta t \\ \beta_{t,t-1} &= 180° - (r_t - r_{t-1}) \\ e_{t,t-1} &= h_t - h_{t-1} \end{aligned} \tag{3.1}$$

These parameters are computed for each point of trajectory, i.e. every time new measurements become available (Fig. 3.4). Thus the raw data is treated even if the person has stopped, which will be discussed in details later.

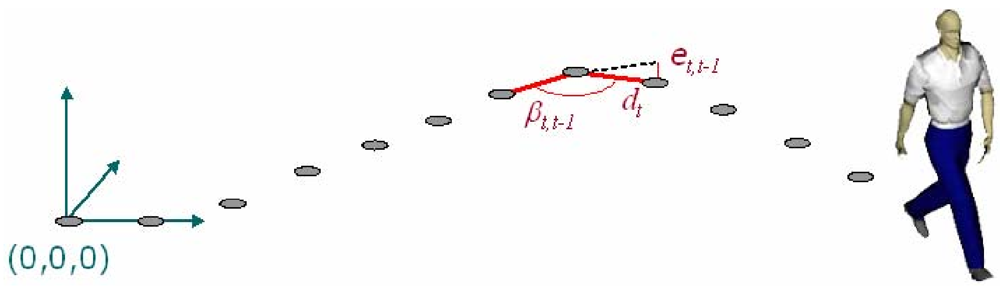



Figure 3.4: Pedestrian trajectory as a sequence of points

The second step in the pre-processing is dedicated to the creation of adequate input for the localisation process. Defined by relative parameters ($d_t$, $\beta_{t,t-1}$ and $e_{t,t-1}$) of each stride the trajectory is a sequence of points and at this stage can not be associated with the contents of the graph. For that reason we need to transform the set of points into a 3D polyline, a geometric form that could be recognized in a 3D graph. The process of localization is strongly dependent on this transformation.

The idea here is to detect the critical movements of the person like turns and vertical movements (taking elevator or stair case) and define them as *critical points* of trajectory. Then, the 3D polyline will be formed by segments connecting consequently the critical points.

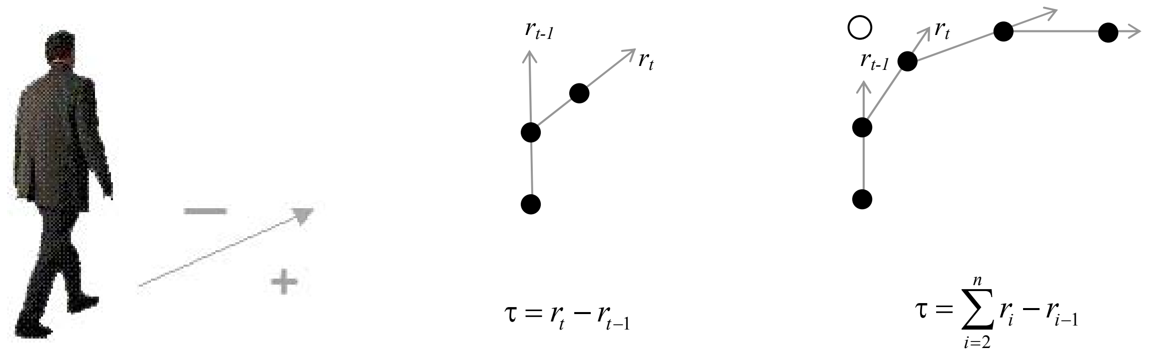
Using the relative parameters ($d_t$, $\beta_{t,t-1}$ and $e_{t,t-1}$) of each stride, we need to determine the relative parameters of the segments of the 3D polyline, i.e. length of segment $l_t$, horizontal angle $\alpha_{t,t-1}$, between two consecutive segments and elevation $\delta_t$ of segment at moment $t$.

Special attention must be paid on the distance $d_t$, which is computed using the raw measurements. Several preliminary tests show that $d_t$ is influenced by a scale factor of about 1.06, due to erroneous accelerometric measurements. Therefore a scale correction must be made of the length of each segment $l_t$.

The computation of these parameters ($l_t$, $\alpha_{t,t-1}$, $\delta_t$) imposes the determination of the coordinates of every critical point in some coordinate system. Therefore we define a user coordinate system with origin (0, 0, 0) in the first registered point of trajectory. The orientation of the coordinate system is not important, so we define the direction of 0° of the first stride (Fig. 3.4).

### Turn detection

Here we discuss in details the definition of the critical points (turns and vertical movements). We assume that on every step the change of direction of walk, reflected by angle $\beta_{t,t-1}$, is negative if the person turns left, positive if the person turns right and zero if the person goes straight (Fig.3.5a). The person can make a turn with sharp change of direction in one step only (Fig.3.5b) or spread over several steps (Fig.3.5c).




a) b) c)

Figure 3.5: Detection of the turns

During the walk in straight direction the measured heading differs from a step to another in the range of ±2°. These small changes of direction are not of interest to us and could not be considered as turns. However, if several consecutive steps are made with the same change of direction (+ or -) we need to compute the total sum of change of direction in order to detect a turn. For that reason, as shown on Fig. 3.5, the value of $\tau$ is computed. If the person changes his direction in one step the value of $\tau$ will correspond to the difference of two consecutive headings (Fig. 3.5b). We can not consider each change of direction as turn, so we need to establish a threshold for the value of $\tau$ which corresponds to a turn. For that we have proceeded to several tests in the buildings of our campus. The test trajectories were of different complexity including elements like turns in the corridors and entry in room and were made on different places covered by the map database. Finally we have got an empirically derived threshold of 18°. So, changes of direction that give $|\tau| \leq 18°$ are not considered as turns. This threshold is just a preliminary value and is modified after the tests, discussed later in chapter 3.2.5.

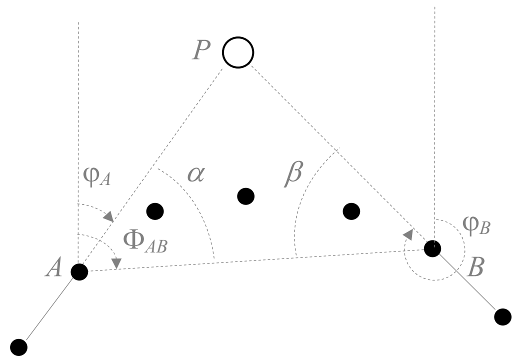



Figure 3.6: Definition of the critical point *P*.

The detection of every turn must be indicated by a critical point mentioned above. For the case of Fig. 3.5b the critical point coincides with the step position where the turn has been made. In the other case (Fig.3.5c) since the turn is made in several steps the critical point must be defined.

Most correctly the turn will be represented by a point placed near the peak of the turn (marked with ○ on Fig.3.5c). The computation of that critical point is simple and it is based on the method of geodetic intersection [Merminod 2003]. In the example on Fig. 3.6 the points *A*, *B* and the horizontal distance *s* are used to compute the coordinates of the critical point *P*.

$$\begin{aligned} X_P &= X_A + \bar{s}_{AP}\cos\varphi_A \\ Y_P &= Y_A + \bar{s}_{AP}\sin\varphi_A \end{aligned} \quad \text{, where } \bar{s}_{AP} = \bar{s}_{AB}\frac{\sin\beta}{\sin(\alpha+\beta)} \tag{3.2}$$

Points *A* and *B* are the start and the end of the turn. If we consider a sequence of points with the same change of direction (+ or -), then *A* and *B* constitute the first and the last point of this sequence.

The detection of turns defines the critical points of the trajectory in horizontal sense. These points are defined in the local coordinate system of the trajectory.

**Vertical movement detection (Change-of-floor)**

The other important critical points are those who represent a vertical movement of the user, i.e. taking elevator or staircase. The measurement that indicates the advancement of the trajectory in vertical direction is the barometric altitude. However, detecting a vertical movement is not an easy task if we use barometric measurements only. The reason is that the personal navigation system use low-cost barometer, its measurement error is large and does not allow for detection of vertical displacement on every step.

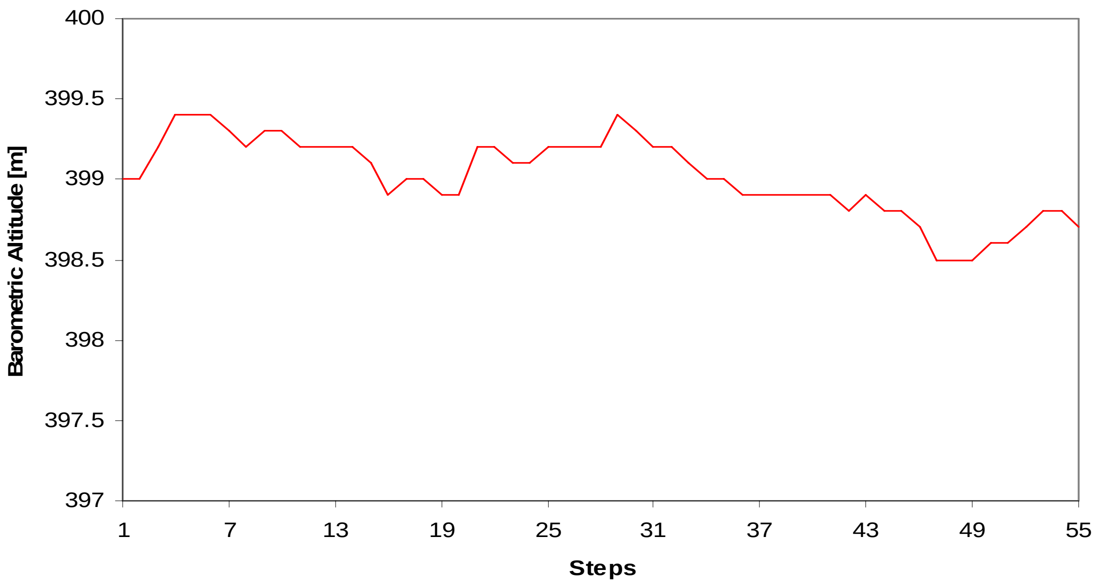



Figure 3.7: Barometric altitudes of a trajectory, made on one floor in horizontal plane.

We have made several simple test trajectories indoors and Fig. 3.7 shows the measured barometric altitudes of one of them. For 40m of walk on one floor only, the measurement error of the barometric altitude rises up to 0.75m. One of the main reasons for these erroneous measurements is the temperature change when passing through the corridors and rooms of the building. The barometer measures the pressure, not the height. The different pressure levels between the corridors, the staircases and the rest of the building are also an important reason for the errors in the altitude [Lachapelle et al. 2003].

The barometric errors can not be represented as a Gaussian distribution, because of different external factors like changes in the temperature, unknown changes in the pressure and air flows, which affect directly the barometric measurements. Taking into account the influence of these external factors, the most realistic representation of the barometric measurements will be given by the model of random walk.

In order to be able to detect a vertical movement (change-of-floor) we need additional information, besides the barometric altitude. The idea is to observe the behaviour of the user when move from one floor to another. There is a big variety of physiological phenomena that characterizes the comportment of every pedestrian. However, in a normal walk some of these phenomena are the same for different individuals. We assume that the vertical movement imposes an important change in user's behaviour and will be necessarily reflected in the measurements.

After several test trajectories consisted in changing the floor we have observed the following phenomena. In the beginning of the staircase the user slows down and when leaving the staircase he/she accelerates again. In the elevator user's behaviour is similar, the person stops when enter and goes when leave the elevator. So, the significant change in the speed is a good indicator for events like entry in elevator/staircase and leaving elevator/staircase. We will use this information together with the barometric measurements to detect vertical movements.

Based on that phenomenon we define four different state events that the user can perform: *go, accelerate, slow* and *stop*. Regarding the speed measurements we can detect the points of the trajectory where these state events occur and to mark every state event with a critical point, named for simplicity *state point*. Thus a vertical movement will be clearly marked by two state points if there is an important difference between the barometric altitudes of these state points.

A significant change in the speed can be detected by computing the speed variance for several consecutive steps. In order to assure a continuous indication of the state points the speed variance must be computed on every step. But the question is how many steps back to consider in this computation. Taking two steps only will not give a reliable indication of the state points, because we risk not detecting some of them. Taking 3 steps will assure reliable indication, and it will come just after the step with significant change in the speed. Taking 4 or 5 steps will assure reliable indication as well, but it will come with one or two steps of delay. Therefore, the speed variance is computed on every step taking the last 3 steps:

$$Var\,(V)_t = \frac{1}{2}\sum_{i=t}^{t-2}(V_i - \overline{V}_t)^2 \quad , \quad \overline{V}_t = \frac{1}{3}\sum_{i=t}^{t-2} V_i \tag{3.3}$$

Fig. 3.8 shows a trajectory composed of successive stops and goes. The significant changes of the speed which corresponds to the state events (stop and go) are indicated by the peaks of the speed variance of the last 3 steps.

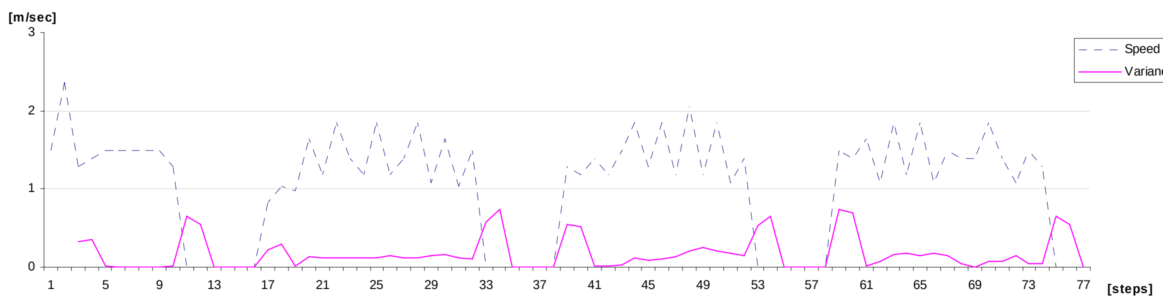



Figure 3.8: Speed and speed variance of stop-go trajectory

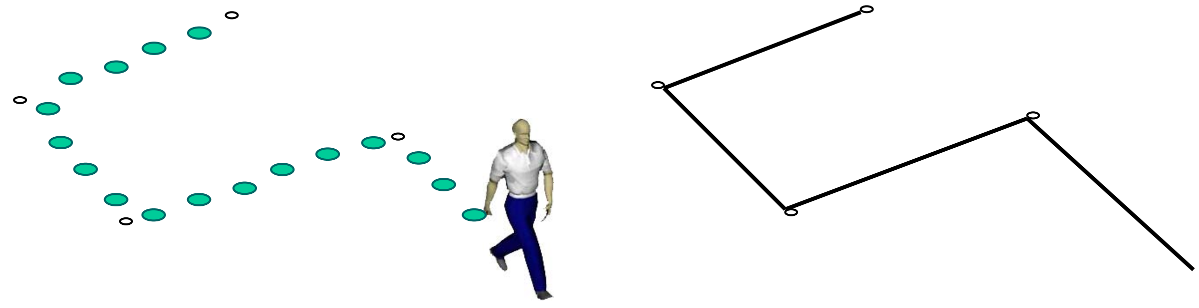
We have proceeded to several test trajectories in order to define a threshold for the speed variance for which a state point is detected. Thus, if the variance is bigger than 0.26, a state point is indicated. Then for detect change-of-floor the elevation between every pair of state points is computed by comparing their barometric altitudes $h_s$ and $h_{s-1}$.

$$\delta_t = h_s - h_{s-1} \tag{3.4}$$

This threshold of 0.26 is just a preliminary value and is modified after the tests, discussed later in chapter 3.2.5. The computed elevation $\delta_t$ approximates the total height of the floors, ascended/descended by the user. The barometric altitude accuracy is insufficient to detect vertical displacement on every step, but it is sufficient to detect change of floor.

**Trajectory transformation**

In order to assure an adequate input to the process of localization the trajectory must be transformed from sequence of points into a 3D polyline. We can say that this polyline generalizes the trajectory, reflecting the critical movements of the person. The detected critical points define the vertexes of the polyline and are connected with segments.

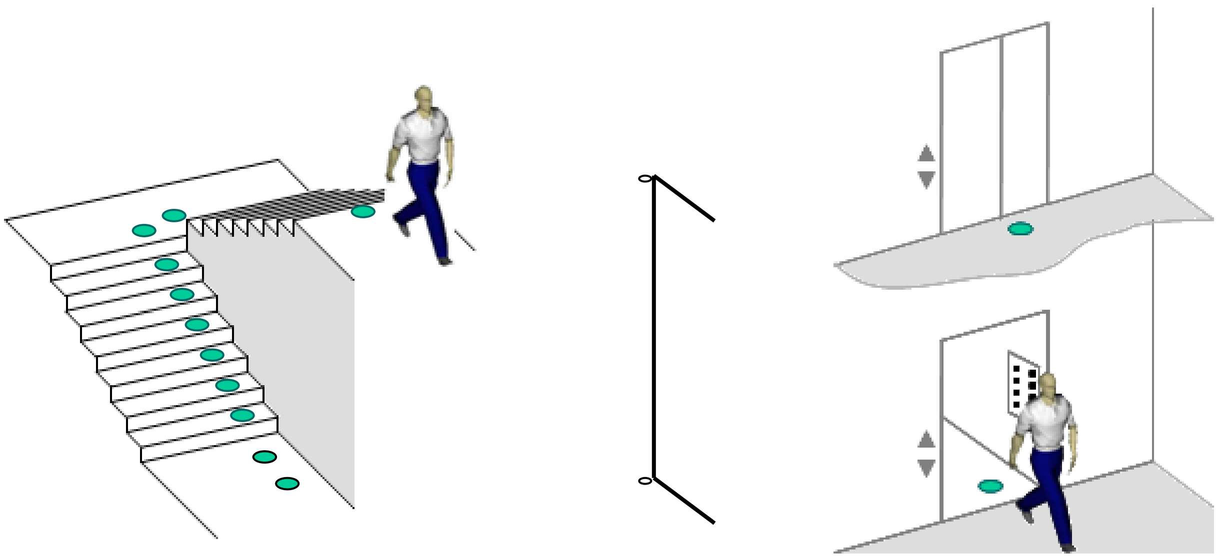


a) b)

Figure 3.9: The sequence of points (a) generalized by a polyline (b).

During the walk every time a new critical point is detected it defines a new vertex. Thus a new segment is added to the 3D polyline.

Fig. 3.9 illustrates a polyline defined by the critical points of several turns. The construction of the 3D polyline in case of change-of-floor is based on the elevation between two consecutive state points.



a) b) c)

Figure 3.10: Generalization of the sequence of points in case of change-of-floor

Considering the computed elevation we can decide if a vertical movement is performed or not. Then the number of passed floors is determined by dividing the elevation by the height of one floor. The residuals of that subdivision are insignificant and do not restrain the precise determination of the number of floors. If a vertical movement is detected the state points are connected with a segment, named *vertical segment* (Fig. 3.10-b). If the user has passed several floors, each floor corresponds to a vertical segment.

In the pre-processing step the changes of the floor (staircase or elevator) are always represented by vertical segments. Considering the staircases and elevators as devices for move from one floor to another, we assume their functionality as topological connections rather than spatial connections. It will be sufficient to determine whether the person has taken one floor up or one floor down. That information is clearly represented by a vertical segment.

**Special cases in the movement**

The detection of critical movements in the pre-processing step deals with the common cases of turn and vertical movement. We stay attached to our assumption that the person performs a normal walk. However, due to the big liberty of movement, many special cases can be observed during the walk. Here we will discuss some of the movements most frequently performed by the individuals: step back, step aside, half-turn and contouring an obstacle.

These movements increase drastically the complexity of the problem of detection of critical movements. Our efforts are pointed to the detection of most of these special movements.

- Step back and step aside

These movements can be considered as basic movements. There are different situations where such movements are performed. A typical example of step back is when pulling to open a door. Another example is when taking an elevator - the user enters, pushes the button and takes orientation to the door. All these operations are supported by steps back and steps aside.

Normally, these basic movements are accidental and are necessarily performed after the user has stopped or after he has gone. However, such basic movements are reflected in the measurements. They can impose the definition of needless critical points that complicates the detection of the critical movements and polyline representation of the trajectory.

On the other hand the locating device chosen in this research, the PNM (refer to chapter 3.1.), is mainly designed to detect user's steps in a normal walk and not in lateral or backward displacement. In the latter two cases the acquisition of desired results is not guaranteed.

Therefore the basic movements like steps back and steps aside must be ignored in the pre-processing of the raw measurements. The simplest way is to exclude the critical points defined after the steps back and steps aside detection. We stay for the hypothesis that in the close vicinity of these critical points, there exists a state point (stop or go). Taking into account a length of stride (~75 cm) we define a range of 1m around the state point. Critical points in this range must be ignored and excluded from the construction of the 3D polyline.

- Obstacle avoidance

Many objects can be considered as obstacles on the way: furniture, materials, other people, etc. In order to avoid an obstacle the user deviates from his way for a short time and quickly returns in the right direction. To develop a technique for detection of obstacle we are based on the typical example on Fig. 3.11.

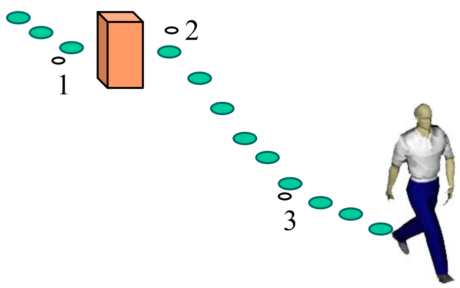



Figure 3.11: Typical example of obstacle avoidance

Following the turn detection technique discussed above, three turns are detected on the figure thus defining three critical points (1, 2 and 3). We use the geometry of that part of user's trajectory to define the so called *triangle of deviation (1-2-3)*. This triangle is necessarily obtuse in point 2 and reflects the manoeuvres performed to avoid the obstacle (Fig. 3.12).

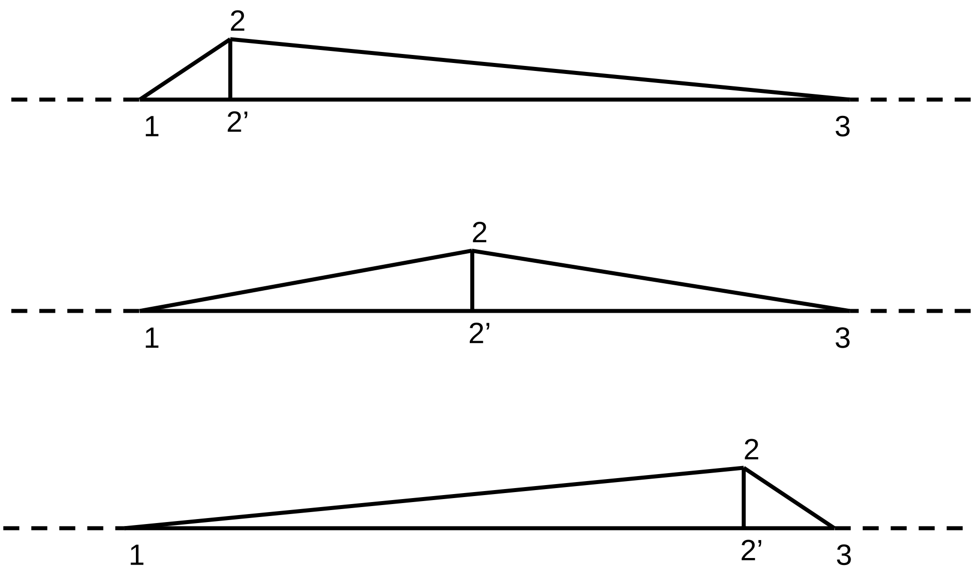



Figure 3.12: Triangles of deviation based on the manoeuvres for obstacle avoidance

In this study we consider only the case of static obstacles. In case of dynamic obstacles (e.g. walking people) more sophisticated figures can be assigned. The aim is to distinguish the manoeuvres from the normal turns and to construct the polyline ignoring the three critical points. For decide whether a set of manoeuvres corresponds to obstacle avoidance we use the following two rules based on computations in the triangle of deviation:

a) The height [2-2'] is relatively short. For the indoor navigation we have chosen [2-2'] $\leq w$, where $w$ corresponds to the width of the corridor in the building.

b) The ratio Surface/Perimeter (of the triangle) is very small, i.e. Surface/Perimeter $\rightarrow 0$.

If both rules are satisfied, then we can consider the manoeuvres as obstacle avoidance. Thus the turns in point 1, 2 and 3 are ignored. Their corresponding critical points are not taken into account in the construction of the polyline.

- Half-turn

With this movement the person takes rapidly the opposite direction of walk. In some cases it is supported with stop and go states. In other cases it is performed in several steps without stopping. The first case has been discussed on the previous page. In the second case the total change of direction is subdivided in two, so as to be considered as two consecutive turns in the same direction (Fig. 3.13). Thus the half-turn is defined with two critical points.

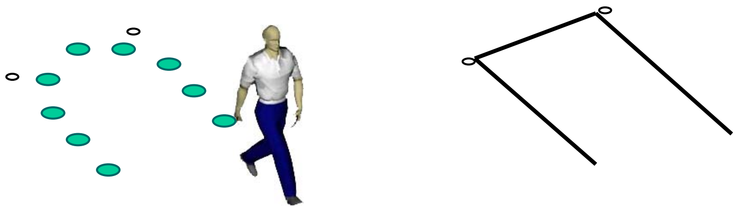
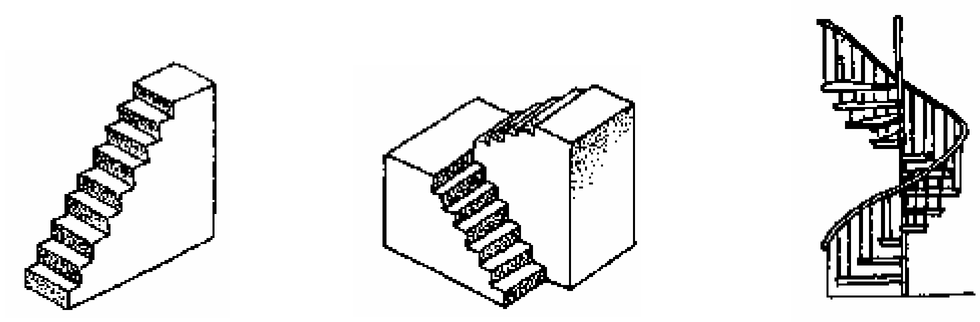


Figure 3.13: Generalization of the points in the case of half-turn

- Stair case

The representation of change-of-floor by a vertical segment reflects more realistically the case when person is taking the elevator. In that case the vertical movement is clearly defined by state points on the corresponding floor. The vertical segment represents truly the geometry of the displacement.

In the staircases the situation is different. In the database applied in our approach (Chapter 2.1.6.) all the staircases are represented by simple vertical edges. There is a big variety of staircases.



a) b) c)

Figure 3.14: Different types of staircases

In some of them the person can step from a floor to another without changing the direction of walk (Fig. 3.14a), in others the person makes a half-turn in the middle (Fig. 3.14b), or to go round in the staircase (Fig. 3.14c).The critical movements in the staircase are detected and critical points are defined. In order to consider the displacement as a change-of-floor the critical points between both state points are ignored.

**The adequate input**

The definition of the segments of the 3D polyline is the final step in the pre-processing which allows creating the adequate input to the initial localization process. That input consists in the relative parameters of the segments: length of segment $l_t$, horizontal angle $\alpha_{t,t-1}$, between two consecutive segments and elevation $\delta_t$ of segment at moment $t$ (Fig. 3.15).

These parameters constitute the polyline as a sequence of segments. Every time a new segment is fixed, a new set of relative parameters is computed. That progressive formation of the 3D polyline is the basis to define the time discretization of the process of initial localization. It is different from the time discretization of the raw measurements acquisition, where every moment $t$ fixes the step event. In the polyline representation of trajectory the moment $t$ corresponds to the determination of a new set of relative parameters ($l_t$, $\alpha_{t,t-1}$, $\delta_t$).

Special attention must be paid to the elevation $\delta_t$. It can have three alternative values: 0, 1 and -1. As mentioned this elevation indicates whether a vertical movement has occurred or not. These values correspond to: take one floor up ($\delta_t = 1$), take one floor down ($\delta_t = -1$), stay on the same floor ($\delta_t = 0$). In the first two cases ($\delta_t \neq 0$), the values of $l_t$ and $\alpha_{t,t-1}$ are not taken into account. Only the information for $\delta_t$ is meaningful when deciding whether the person has changed the floor. In the case when $\delta_t = 0$ the values of $l_t$ and $\alpha_{t,t-1}$ are computed using the coordinates of their adjacent vertexes.

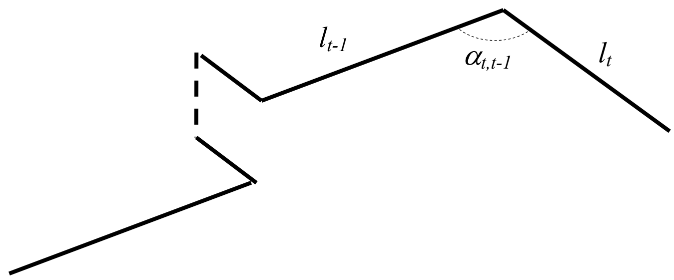



Figure 3.15: Relative parameters of the 3D polyline at moment $t$. The dashed line represents a vertical movement.

A flowchart of the pre-processing step is shown in Fig. 3.16. The input of the raw measurements is made on every user's step; on the other hand the output is made only when a critical movement is detected.

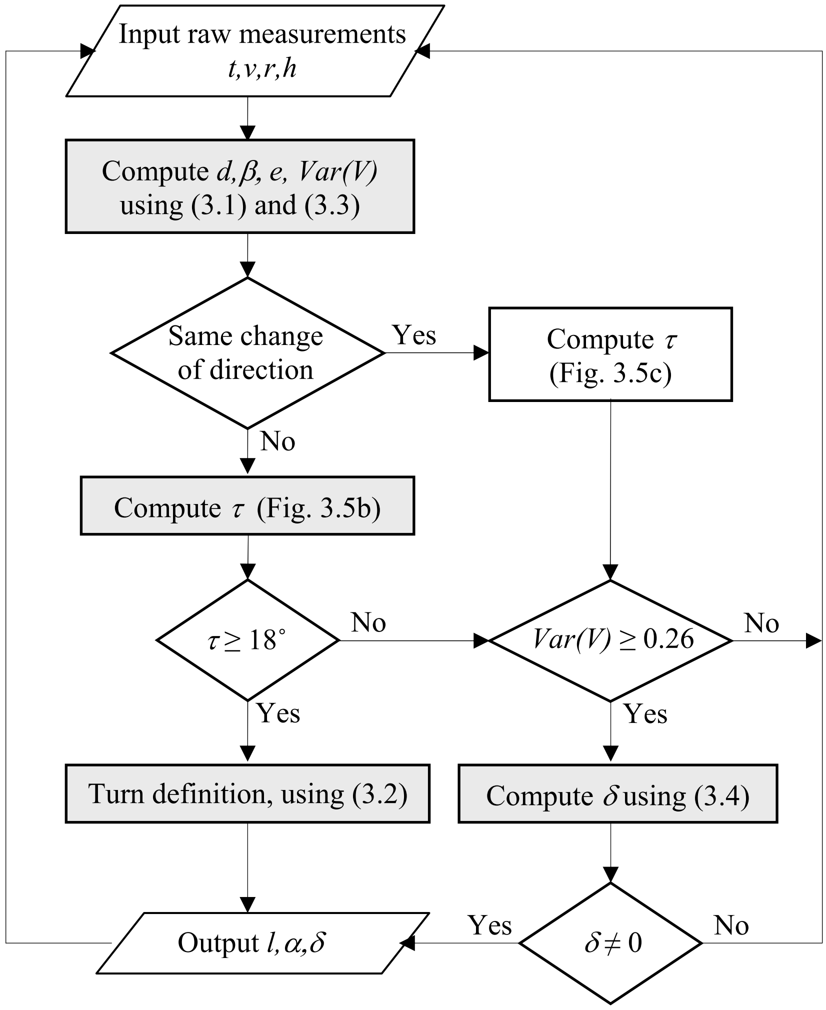



Figure 3.16: Flowchart of the pre-processing step.

## 3.2.2. Problematic of the initial localization

The initial localization is to find the edge of the graph occupied by the person and the person's orientation on that edge.

We assume that since the user walks in the building, his trajectory passes through the corridors, stairs, elevators, etc. Thus the 3D polyline that reflects the trajectory (Fig. 3.17a) covers certain part of the graph of the building (Fig. 3.171b).

The method in this approach is based on the association of similar geometric forms and topological information from the graph and the trajectory. This process is known in the literature as map-matching.

Consider the polyline as the history of movement and the last segment corresponding to user's actual location. We can determine user's location in the graph if we find the edge of the graph which corresponds to the last segment of the polyline. This is possible if we consider the history of movements, i.e. whole polyline, and find its placement in the graph.

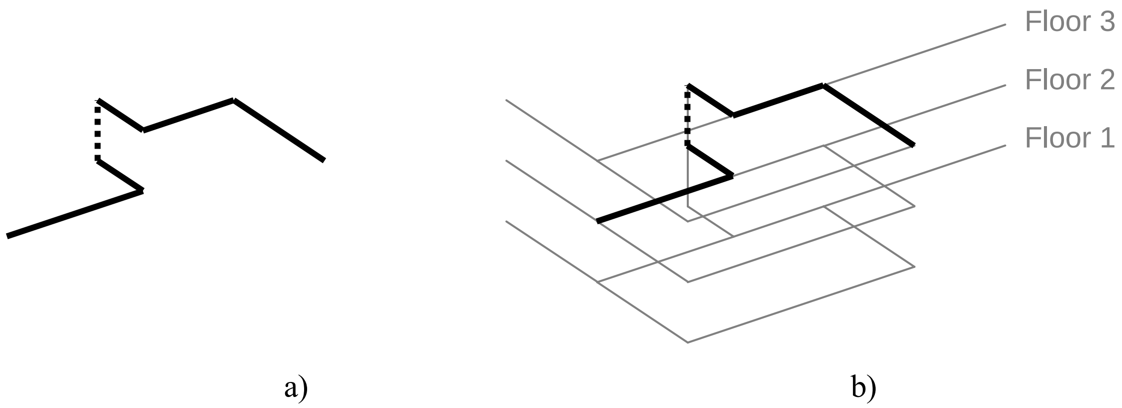



Figure 3.17: The 3D polyline as part of the graph

The aim is to find in the graph the set of successive edges that fits best the form of the polyline at moment *t*. Every time a new critical point is detected, a new segment is added to the polyline and the matching process repeats. Depending on the building the graph can have a symmetric structure with repetitive elements. Thus in certain moment *t* the best match of the polyline can be found in several places in the graph (Fig. 3.18). Later, with the acquisition of new measurements there will be a moment when the polyline will hold enough information. That will allow finding the unique placement of the polyline in the graph and we will determine the edge occupied by the user, named for simplicity *location edge*. The definition of user's orientation is based on the hypothesis that the person performs a normal walk. Knowing the location edge and the edge occupied before, we can identify in what direction the person goes. Thus, in the moment of determination of the location edge user's orientation is defined to be equal to the orientation of the edge in the direction of walk.

The process of initial localization depends on the acquisition of information on the trajectory based on inertial measurements. That means the person can be localized after he has started his trajectory. His location will be determined in the moment when sufficient information on his displacement in the building is acquired. The polyline is constructed from erroneous measurements, so it is impossible to find a perfect match of the polyline in the graph. Instead the best match could be estimated by applying probabilistic approach.

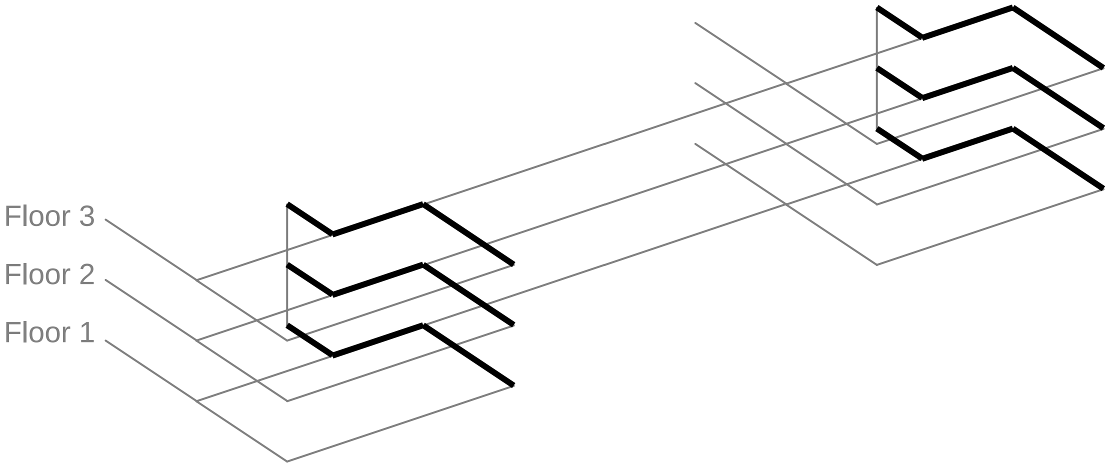



Figure 3.18: Six possible placements for the polyline in the graph.

While the graph has a finite number of elements, the polyline is updated with new data periodically. Every time the polyline is updated an estimation of the user's location will be performed until the unique placement of the polyline is found on the graph. The estimation relies on prior information (the trajectory, actual measurements and map database) that could be used to compute a posterior estimation of location via the Bayesian inference.

### 3.2.3. Bayesian formulation

Bayesian theory provides a framework to achieve an optimal strategy in inference and decision making. It makes use of a prior probability distribution, standing for the likelihood that certain hypotheses are true, based on observations and acquired knowledge [Althaus et al., 1999]. By observing present events Bayes theorem allows one to update this probability distribution. The name "Bayesian" comes from the frequent use of Bayes' theorem in the inference process. Bayes' theorem relates the conditional and marginal probabilities of stochastic events A and B as follows:

$$p(A|B) = \frac{p(B|A)\,p(A)}{p(B)} \tag{3.5}$$

where

*p(A|B)* is the *conditional probability* of A, given B, also called the *posterior probability*;

*p(B|A)* is the conditional probability of B, given A;

*p(A)* is the *prior probability* of A;

*p(B)* is the prior probability of B, acting as a normalizing constant;

If we define *L(A|B)* as the *likelihood*, the Bayes' theorem may be paraphrased as:

$$p(A|B) \propto L(A|B)\,p(A) \tag{3.6}$$

In words the posterior probability is proportional to the product of the prior probability and the likelihood.

Bayesian inference is statistical inference in which observations are used to update or to infer the probability that certain hypothesis may be true. Bayesian inference uses a numerical estimate of the degree of belief in a hypothesis before observation has been made and calculates a numerical estimate of the degree of belief in the hypothesis after observation has been made. With the accumulation of information from the observations, the degree of belief in the hypothesis changes. With enough observations, that degree becomes very high or very low. Thus a hypothesis with very high degree of belief should be accepted as true.

How the Bayesian inference is applied in our approach. The process of initial localization aims at determination of the user's location on the map, based on the history of movement and actual measurements.

Thus we need to compute the probability that the user occupies certain edge of the graph at moment $t$. At every moment $t$ we acquire new information on the user's trajectory reflecting the evolution of the polyline. Accumulating this information we will evaluate for each edge the degree of belief in the hypothesis that the person is on that edge. So the problem of localization of the person is transformed into localization of a polyline segment in the contents of the graph.

The walking person is considered as a dynamic system, whose trajectory is presented as 3D polyline. The evolution of that dynamic system is reflected by the addition of new segment to the polyline at each moment $t$, which is defined by the following state-space model:

$$x_t = f\left(x_{t-1}, u_t\right) \tag{3.7a}$$

$$y_t = h\left(e^{(i)}, e^{(i+1)}\right) + z_t \tag{3.7b}$$

with the following elements:

| | |
|---|---|
| $x_t$ | - state vector, representing an edge at moment t |
| $u_t$ | - motion input |
| $y_t$ | - measurement vector |
| $h(e^{(i)},e^{(i+1)})$ | - dimensions of $x_t$ and $x_{t-1}$ according to the map database |
| $z_t$ | - measurement error |

In the state equation (3.7a) the state vector $x_t$ represents the edge in moment $t$. The motion input $u_t$ characterizes the evolution of the process, i.e. the user will be on $x_t$ after performing a movement $u_t$ from $x_{t-1}$. The measurement vector $y_t = (l_t, \alpha_t, \delta_t)^T$ in the measurement equation (3.7b) includes the distance of the polyline segment, the horizontal angle with the past polyline segment and the elevation. These are the parameters computed in the pre-processing phase. The function $h$ contains the same relative information for a pair of edges of the graph considering data from the map database. The history of all states up to moment $t$ is defined by $X_t=\{x_0, x_1, \ldots, x_t\}$, respectively $Y_t=\{y_1, y_2, \ldots, y_t\}$ defines the history of the input data up to moment $t$. The problem to solve is using the set of all available measurements $Y_t$, to estimate the probability of given edge $x_t$ to be occupied by the user's. Estimation is made every time the new measurements $y_t$ are available. The process of acquisition of input data $(l_t, \alpha_t, \delta_t)$ is discretized considering the definition of new segment. Therefore for simplicity we denote each segment with $t$, which corresponds to the moment $t$. Note that in 3.7a we don't introduce an element of error. That is because the evolution of the process is dictated of the graph representation of the building, which is considered as perfect, i.e. we consider that the map database is constructed without any error.

From Bayesian viewpoint this sequential estimation problem demands the computation of the posterior density $p(X_t|Y_t)$. We assume that the state follows a first order Markov process:

$$p(x_t|x_{t-1}, x_{t-2}, \ldots, x_0) = p(x_t|x_{t-1}), \quad \text{and} \quad p(x_0|x_{-1}) = p(x_0) \tag{3.8}$$

So if we compute the marginal of the posterior density $p(x_t|Y_t)$, also known as filtering density, there is no need to keep the complete history of the states [Doucet et al., 2001].

$$
\begin{aligned}
p(x_t|Y_t) &= \frac{p(Y_t|x_t)p(x_t)}{p(Y_t)} \\
&= \frac{p(y_t, Y_{t-1}|x_t)p(x_t)}{p(y_t, Y_{t-1})} \\
&= \frac{p(y_t|Y_{t-1}, x_t)p(Y_{t-1}|x_t)p(x_t)}{p(y_t|Y_{t-1})p(Y_{t-1})} \\
&= \frac{p(y_t|Y_{t-1}, x_t)p(x_t|Y_{t-1})\cancel{p(Y_{t-1})}\cancel{p(x_t)}}{p(y_t|Y_{t-1})\cancel{p(Y_{t-1})}\cancel{p(x_t)}} \\
&= \frac{p(y_t|Y_{t-1}, x_t)p(x_t|Y_{t-1})}{p(y_t|Y_{t-1})} \qquad (3.9)
\end{aligned}
$$

Here $p(x_t|Y_{t-1})$ is the prior of the state at moment $t$, $p(y_t|Y_{t-1},x_t)$ is the likelihood function, the evidence $p(y_t|Y_{t-1})$ acts as a normalizing constant. The repetitive acquisition of new data on the trajectory provides new input to the computation at every moment $t$. Thus $p(x_t|Y_t)$ can be computed recursively in two stages: *prediction* and *update*.

The update step is used to compute the likelihood function. We determine a specific weight $w_t^{(i)}$ for each edge $e^{(i)}$ in the graph where $i=1...n_e$ is the number of the edges. That weight reflects the probability for an edge to be occupied by the person. It is composed by two sub weights: $w_m^{(i)}$, using the actual input data $y_t = (l_t, \alpha_t, \delta_t)^T$ and $w_h^{(i)}$, using data history $Y_{t-1}$.

For the first sub weight we compare the input data $y_t = (l_t, \alpha_t, \delta_t)^T$ with the characteristics of each edge in the graph, i.e. the length $L(e^{(i)})$, the angle with the previous edge $B(e^{(i)}, e^{(i+1)})$ and the elevation $\Delta( e^{(i)})$. We denote:

$$\Delta l^{(i)} = l_t - L(e^{(i)}) \qquad (3.10a)$$

$$\Delta \alpha^{(i)} = \alpha_t - B(e^{(i)}, e^{(i+1)}) \qquad (3.10b)$$

where $\Delta l^{(i)}$ is the residual between the lengths of the segment $t$ and the edge $e^{(i)}$. Respectively, $\Delta\alpha^{(i)}$ is the residual between the horizontal angles $\alpha_t$ and $B(e^{(i)}, e^{(i+1)})$. These residuals are used to compute:

$$w_l^{(i)} = 1 - \frac{\Delta l^{(i)}}{\sum_{i=1}^{n_e} \Delta l^{(i)}} \qquad (3.11a)$$

$$w_\alpha^{(i)} = 1 - \frac{\Delta \alpha^{(i)}}{\sum_{i=1}^{n_e} \Delta \alpha^{(i)}} \qquad (3.11b)$$

And then:

$$w_m^{(i)} = w_l^{(i)} \cdot w_\alpha^{(i)} \tag{3.12}$$

The sub weight $w_m^{(i)}$ characterizes the resemblance between the data input and each edge $e^{(i)}$ of the network. It is evident that smaller residuals $\Delta l^{(i)}$ and $\Delta\alpha^{(i)}$ lead to bigger $w_m^{(i)}$.

Here we show the computation of $w_m^{(i)}$ by treating the horizontal angle $\alpha_t$. That is the case where no vertical movement is detected and $\delta_t$ is not taken into account. Respectively, in the case of vertical movement we treat only $\delta_t$ without taking into account $\alpha_t$. We consider both cases as mutually exclusive. The reason is that the vertical connections (elevators and staircases) are presented in the network as simple vertical edges $\Delta( e^{(i)})$. On the other hand the vertical segments in the polyline are characterized by $\delta_t = \{0, 1, -1\}$ representing the vertical movements simply as change of the floor ignoring possible changes in direction of walk.

In the graph all vertical edges have the same length $L(e)$. So it will not be reasonable to compute the residuals $\Delta l^{(i)}$ or $w_l^{(i)}$. Thus in the case of vertical movement the first sub weight $w_m^{(i)}$ will depend only on $\delta_t$, which will indicate an elevation or descending. We write:

$$w_m^{(i)} = \begin{cases} 1, & if\, \delta_t = \Delta(e^{(i)}) \\ 0, & otherwise \end{cases} \tag{3.13}$$

For the second sub weight $w_h^{(i)}$ we take into account the data history $Y_{t-1}$ assuming that it covers a part of the graph. That is the person has passed that part of the network before to arrive to the occupied edge. So there exists a sequence of segments that corresponds to sequence of successive edges in the graph.

We write:

$$w_h^{(i)} = \prod_{j=2}^{T-1} q^{(j)} \tag{3.14}$$

where

$$q^{(j)} = \begin{cases} 1, & if\, e^{(j-1)} \to e^{(j)} \quad and \quad p(y_j \middle| Y_{j-1}, x_j, x_{j-1}) \to 1 \\ 0, & otherwise \end{cases} \tag{3.15}$$

The sub weight $w_h^{(i)}$ indicates the presence of the passed polyline $Y_{t-1}$ in the graph. Here $q^{(j)}$ is Boolean variable that checks the topological connectivity of the graph elements and compares the input data on each segment $j$ with these graph elements.

Thus the total weight for each edge $e^{(i)}$, $i=1...n_e$, in the network is computed as product of both sub weights:

$$w_t^{(i)} = w_m^{(i)} \cdot w_h^{(i)} \tag{3.16}$$

Finally, for the likelihood we write:

$$p(y_t | Y_{t-1}, x_t, x_{t-1}) = p(y_t | x_t, x_{t-1}) p(x_{t-1} | Y_{t-1}) \tag{3.17}$$

Following the concept of the Bayes' theorem we compute the posterior probability by multiplying the likelihood by the prior (3.6).

The prediction step is used to compute the prior as follows:

$$p(x_t | Y_{t-1}) = \sum_{x_{t-1}} p(x_t | x_{t-1}) p(x_{t-1} | Y_{t-1}) \tag{3.18}$$

The quantity $p(x_{t-1}|Y_{t-1})$ is available from the computation of the posterior probability for the last segment, and the model $p(x_t|x_{t-1})$ simply characterizes the topology of the graph. Consider $x_{t-1}$ as estimation at moment $t$. Thus, $p(x_t|x_{t-1}) = 1$ if $x_t$ is a possible successor of $x_{t-1}$ in the graph, respectively $p(x_t|x_{t-1}) = 0$ if $x_t$ is not a possible successor of $x_{t-1}$ in the graph. The simplest possible model is to assign equal probability to each feasible successor of $x_{t-1}$, but more sophisticated characterizations of the topology can be used in this framework.

### 3.2.4. Algorithm for initial localization

The computation of the posterior probability can be regarded as process of repetitive computation of the specific weights $w_t^{(i)}$ for each edge in the graph. This computation is implemented in an algorithm that aims at the localization of the person on the map.

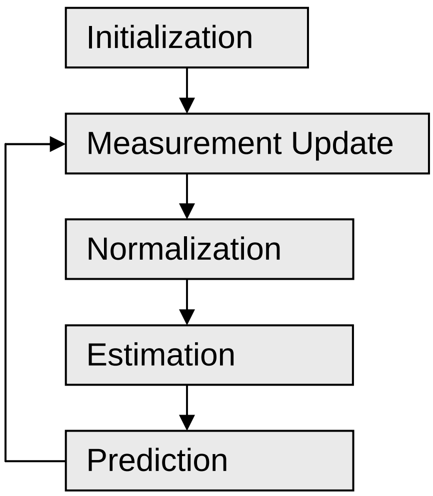



Figure 3.19: Principal phases of the algorithm for initial localization

There are two sources of input data: the map database and the polyline parameters. An iteration of the algorithm is performed every time a new segment is added to the polyline. The phases of the algorithm are illustrated as Fig. 3.19 and are discussed in details further.

The *initialization* is the first phase in the algorithm. At this stage ($t$ = 0) there is no available information on the user's trajectory. So we can not evaluate the probability distribution of user's location. Instead, we can define it as uniform distribution by giving equal weights to all of the edges of the graph. This definition corresponds to the assumption that at moment $t = 0$ the person can be anywhere in the building (Fig. 3.22a). It can be written as follows:

$$w_0^{(i)} = \frac{1}{n_e}, \quad i = 1, ..., n_e \tag{3.19}$$

where $w_0^{(i)}$ is the weight of the edge $i$ and $n_e$ is the number of edges in the graph.

The *measurement update* is the phase where the likelihood is computed (Fig 3.22b). It consists in the update of the weights of the edges at moment $t$, using the available set of input data $(l_t, \alpha_t, \delta_t)^T$ and the history of input data $Y_{t-1}$ as shown in (3.10). In order to estimate the posterior probability we multiply the likelihood by the prior. That is the weight of each edge is multiplied by its prior weight (1 or 0).

The *normalization* phase reflects the inference in (3.6). That means the updated weights are normalized as follows:

$$\overline{w}_t^{(i)} = \frac{w_t^{(i)}}{\sum_{i=1}^{n_e} w_t^{(i)}}, \quad i = 1, ..., n_e \tag{3.20}$$

After the normalization we estimate the location ($\hat{x}_t$) of the person on the graph by choosing the edges regarding their weight. It is possible that at certain moment $t$ the estimation consists in several edges. As mentioned above, that means the unique placement of the polyline is not found yet. At this stage of the algorithm we need to define what weight must have an edge in order to be estimated as user's location.

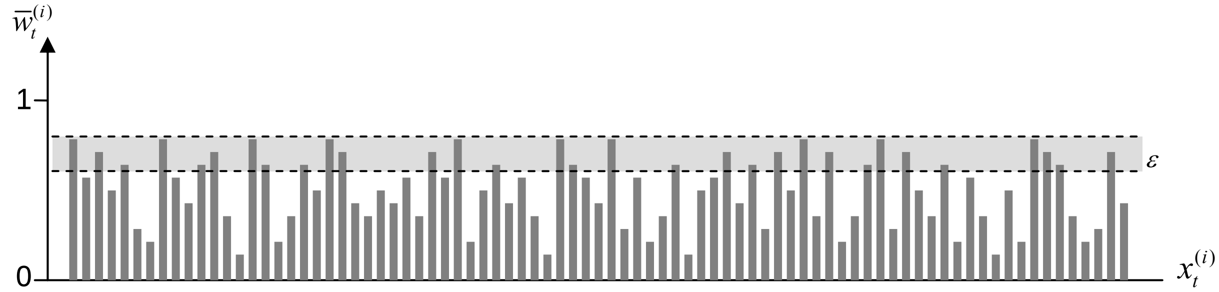



Figure 3.20: Definition of the parameter $\varepsilon$. On this figure $\varepsilon$ = 25% of the maximal weight $w_{(t)}^{(MAX)}$ at moment $t$.

If in the estimation phase of the algorithm we consider only the edges with maximal weight as user's location, we ignore the rest of the edges. At moment $t$ it is possible that the location edge has no maximal weight, but a "near-maximal" weight and we may not estimate the correct location of the user. To avoid such faults we need to define a threshold for the weights considered in the estimation phase. That threshold is represented by the parameter $\varepsilon$ as a percentage of the maximal weight (Fig. 3.20). Thus all edges with a weight larger than $\varepsilon$ are eligible as the location of the user. We write:

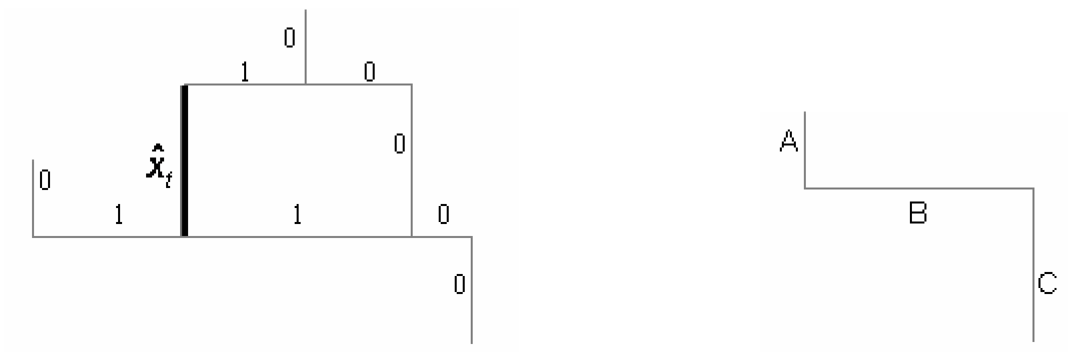
$$\hat{x}_t = x_t^{(i)} \Big| \overline{w}_t^{(i)} \geq \left( \overline{w}_t^{(MAX)} - \varepsilon \right), \quad i = 1, \ldots, n_e \tag{3.21}$$

The optimal value of $\varepsilon$ is evaluated after the tests, discussed later in chapter 3.2.5.

The *prediction* phase aims at the computation of the prior. That corresponds to the determination of the next probable locations on the graph. This determination depends on the last estimation. We write:

$$w_{t+1}^{(i)} = \begin{cases} 1 & ,\hat{x}_t \ \ is \ \ neighbor \ \ of \ \ x_{t+1}^{(i)} \\ 0 & \text{,otherwise} \end{cases}, \quad i = 1, \ldots, n_e \tag{3.22}$$

Here $w_{t+1}^{(i)}$ is the prior weight of the edge $i$. The predicted edges are supposed to be the neighbours of the last estimated edges (Fig. 3.21a). This assumption springs from topological point of view, illustrated on Fig. 3.21b. The person could not step directly from edge A to edge C without passing through edge B. Thus in the prediction phase the predicted edges will have weight 1 and the others - weight 0 (Fig. 3.22d).




a) b)

Figure 3.21: Prediction phase chooses the neighbour edges as probable locations of the next moment and gives them weight 1.

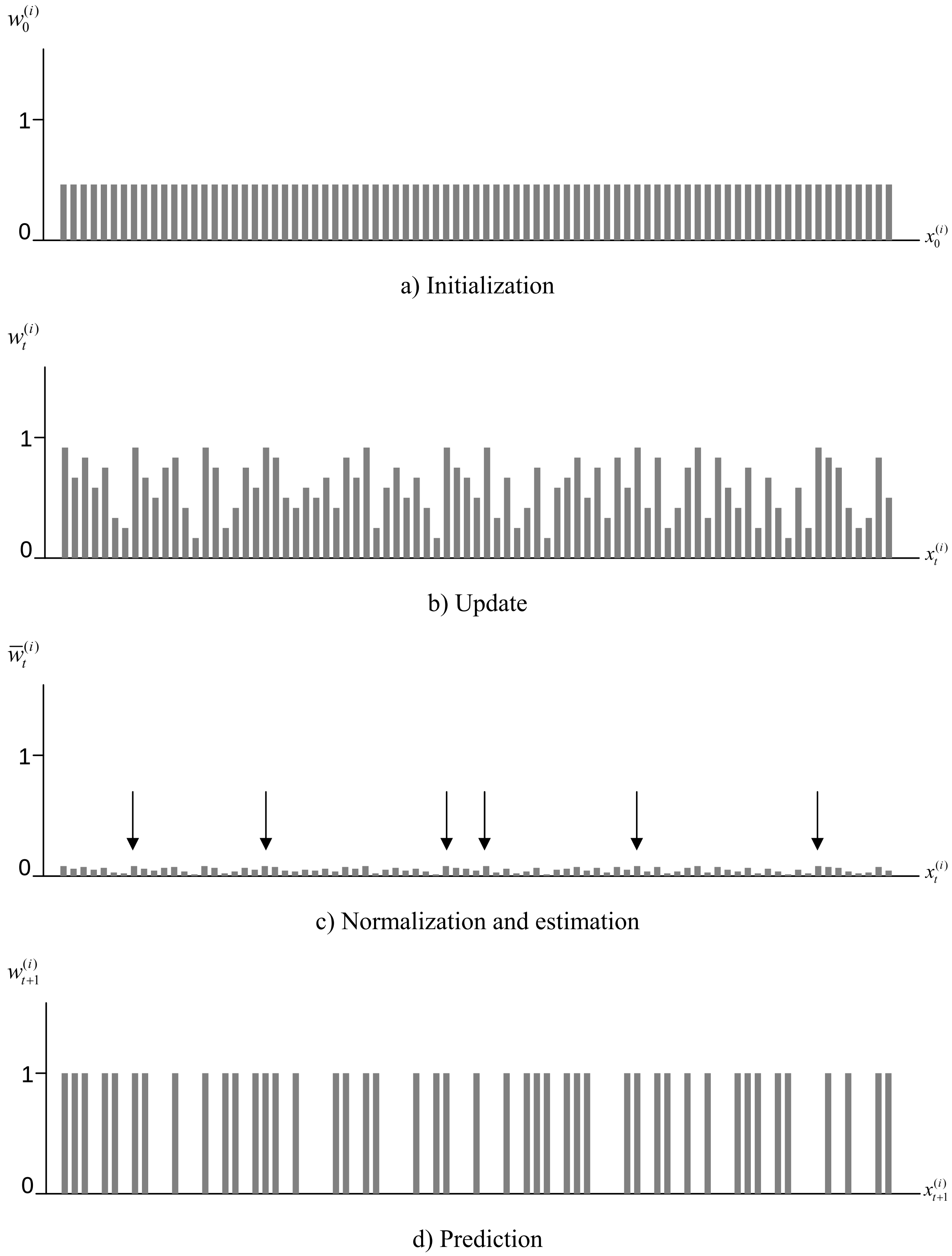



Figure 3.22: Principal phases of the algorithm for initial localization. The edges of the graph are settled on the abscissa. The ordinate reflects the weights of the edges.

At moment $t$ the weights computed for each edge of the graph define the probability distribution as discrete multimodal distribution. Those edges that possess the highest weight are estimated as best match to the last segment of the 3D polyline (Fig. 3.23).

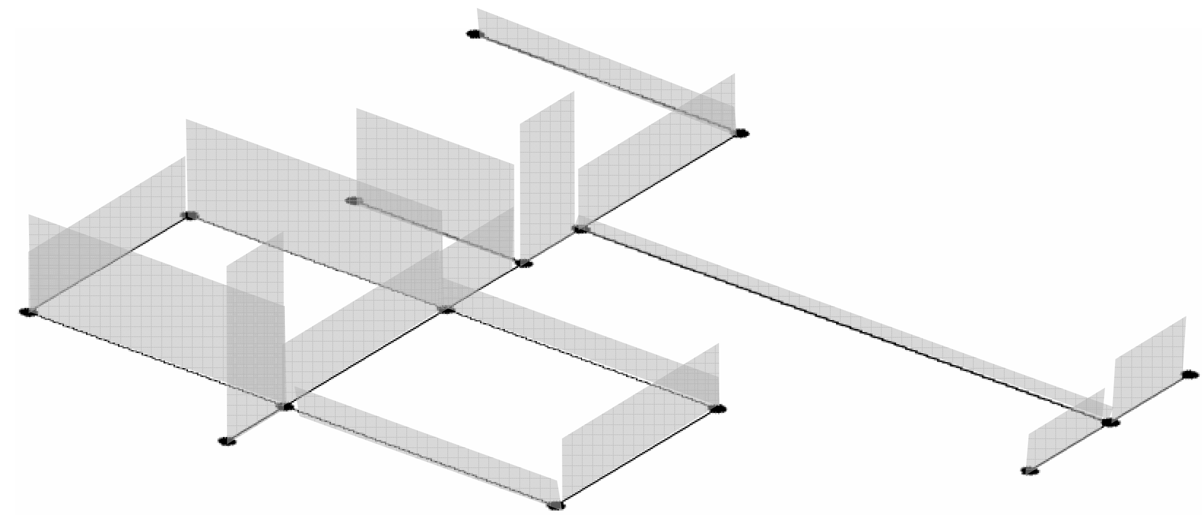


Figure 3.23: Illustration of the probability distribution at moment $t$ for a part of the graph. The gray verticals are proportional to the weights of the edges.

With the accumulation of information on the polyline that distribution changes in the time. When user's location is found, i.e. the unique placement of the polyline is determined, only one edge will have the highest weight of 1 and the other edges will have weight 0. In that moment the distribution becomes unimodal distribution (Fig. 3.24).

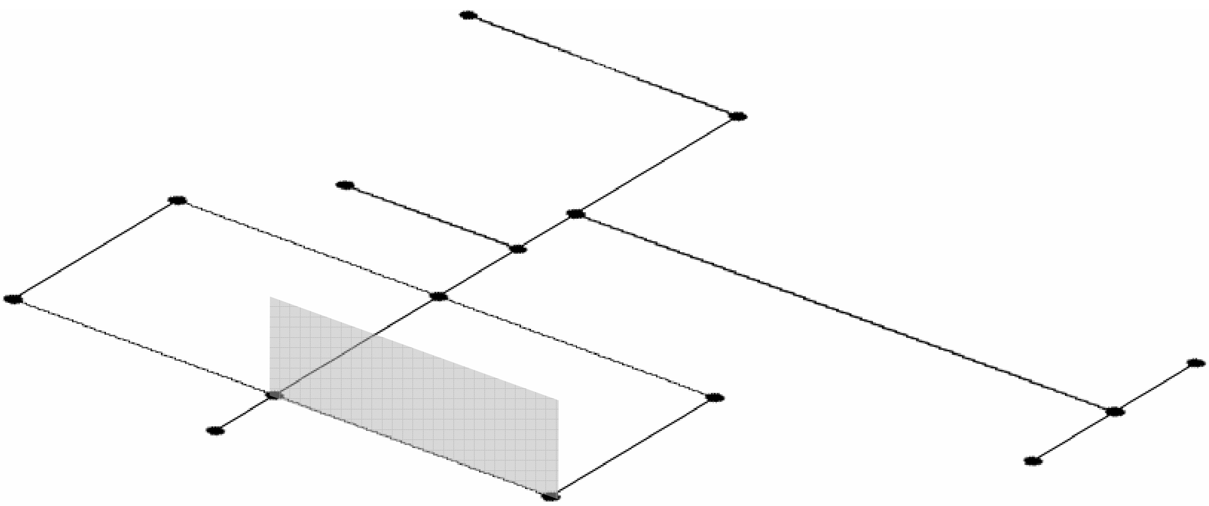


Figure 3.24: Illustration of the probability distribution when user's initial location is found.

The algorithm of initial localization is presented with flowcharts for every phase as follows.

Fig. 3.25 gives a general view of the algorithm of initial localization. The gray cages present the main phases which are explicitly discussed further in separate flowcharts (Fig. 3.26, Fig. 3.27 and Fig. 3.28).

The phase of *history update* aims at completing the history of input data $(l_t, \alpha_t, \delta_t)$. This history contains information on the polyline from the beginning of the trajectory up to moment $t$-$1$. At moment $t$ the new set of input data (the last segment) is added to the history in order to be used in the next iteration.

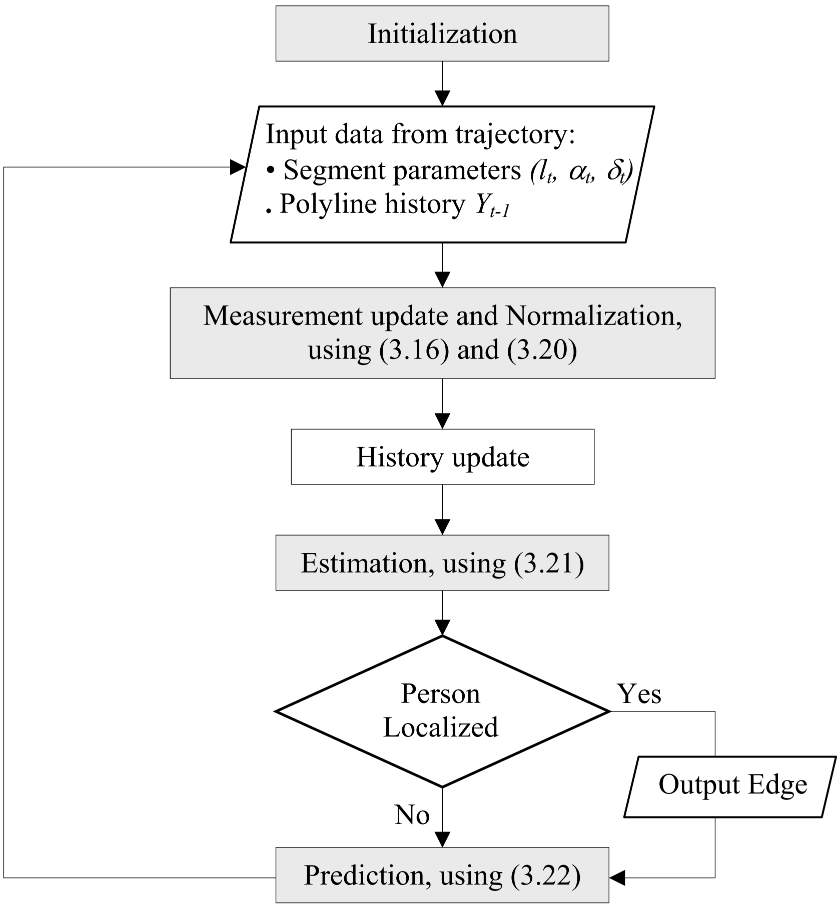



Figure 3.25: Flowchart of the initial location.

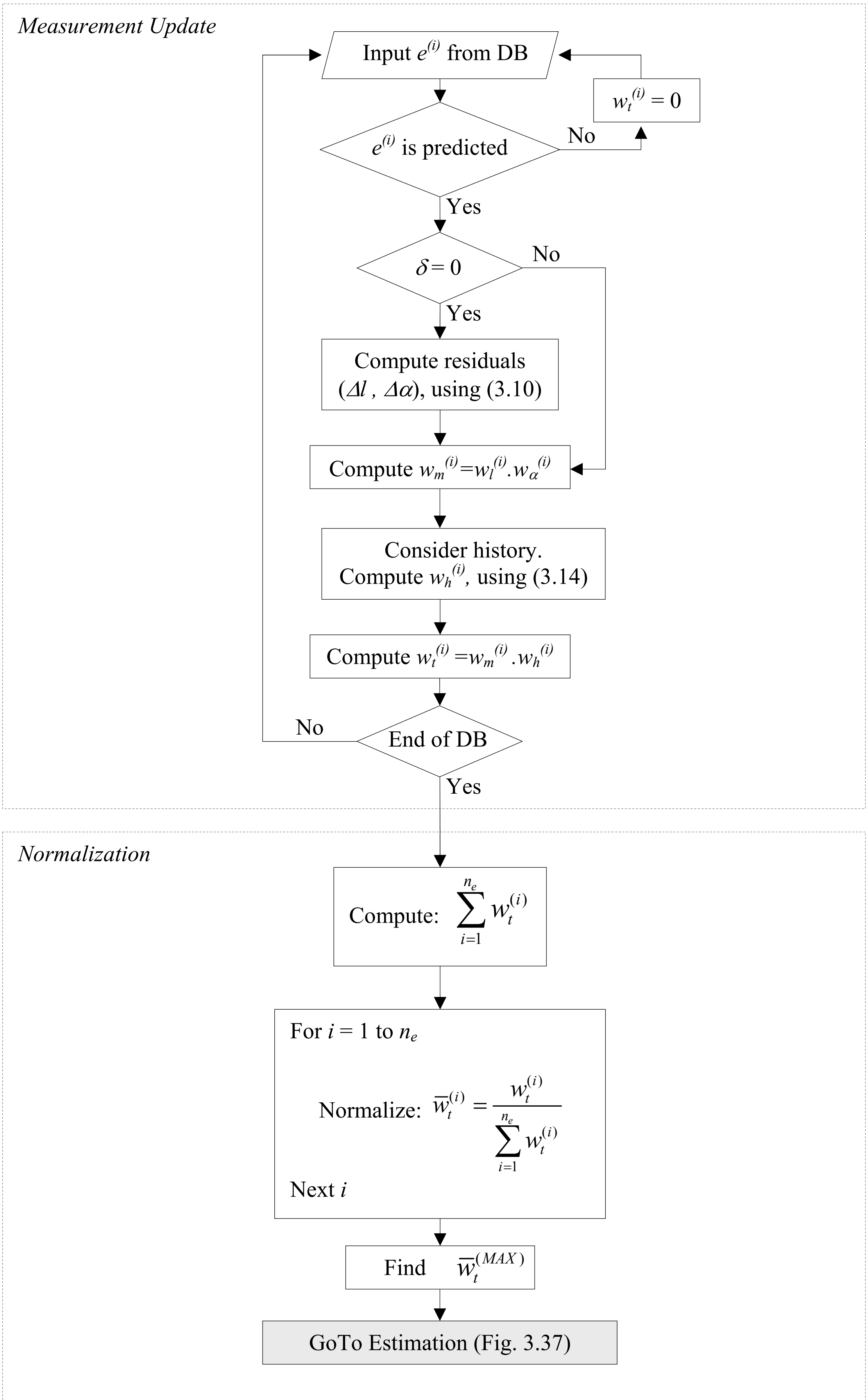



Figure 3.26: Flowchart of the phases of measurement update and normalization

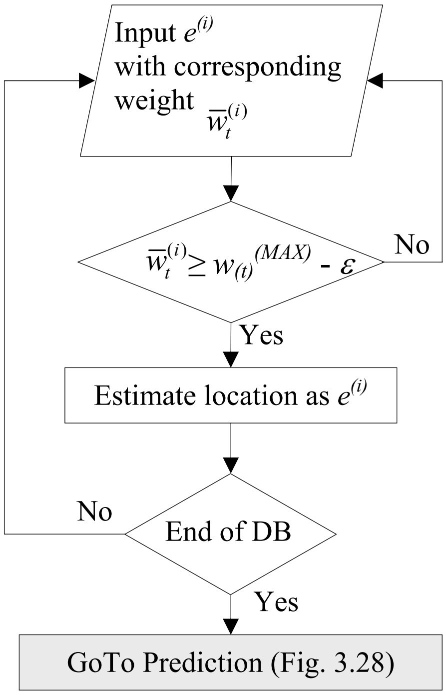



Figure 3.27: Flowchart of the estimation phase.

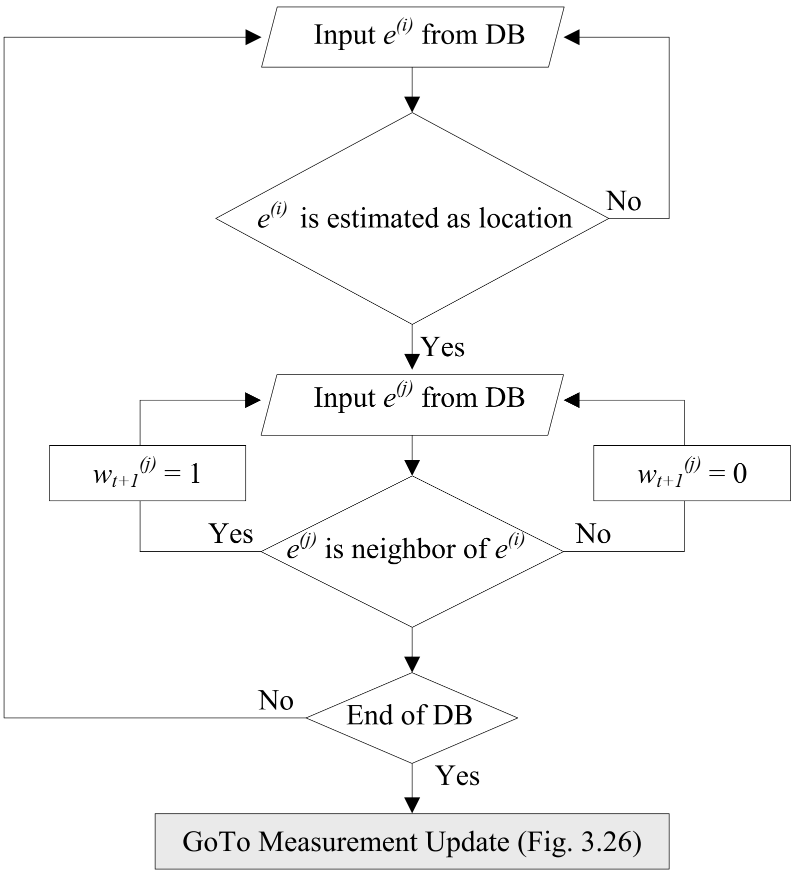



Figure 3.28: Flowchart of the prediction phase.

### 3.2.5. Tests, Results and analysis

Several scenarios were made to test the robustness and the efficiency of the algorithm for initial localization. They include normal walk, taking elevator, taking stairs, stop-go, avoiding obstacle, entering and leaving a room. In the tests four persons with different height were involved.

The trajectories were performed using the PNM (ref. Chapter 3.1.). Following the factory instructions a calibration of the module is performed for each person. That calibration is necessary for two reasons:

- To eliminate the magnetic disturbances caused by metallic objects on the person.
- To define the user's length of step and the misalignment between the user's line of sight and the PNM line of sight.

Once the calibration is made the trajectories are performed. The raw measurements of each trajectory are saved in separate file on the user's PDA. Later these files are transferred to a PC and treated in post-treatment mode. As play ground for the tests we used one of the buildings of the EPFL, consisting of three floors.

Data collection is designed to challenge the algorithm and identify the instances when it fails. The performance of the algorithm is tested applying different values of the following parameters:

- Threshold for turn detection during the pre-processing phase. This parameter is discussed in chapter 3.2.1. During the walk, changes of direction whose value is below this threshold are not considered as turns. Values used during the tests range from 1° to 60° with increment of 1°.
- Threshold for speed variance during the pre-processing phase. The use of this parameter helps to detect the state points and vertical movements (chapter 3.2.1). Values used during the tests range from 0.05 to 0.40 with increment of 0.01.
- Number of edges considered as history of movement. Defines how many segments of the polyline (trajectory) must be considered in order to compute the sub weights, discussed in chapter 3.2.3. Values used during the tests are range from 2 to 6 with increment of 1.
- Threshold ($\varepsilon$) for the weights of the edges in the estimation phase. This parameter defines what weight an edge must have in order to be considered as the user's location, discussed in chapter 3.2.4. Values used during the tests range from 0% to 100% with increment of 1%.

The test results are presented in Table 3.1. The tests are made for the same ranges of values for the four parameters. In the first 12 cases the person is localized and the number of iterations is shown. The localization is effective for a specific range of values for each parameter, which is shown graphically later on. In the other cases the person is not localized for various reasons, shown as remarks and discussed further.

| Trajectory No. | Turn detection threshold (deg) | Speed variance threshold | History (edges) | $\varepsilon$ (%) | Localization on iteration | Remarks |
|---|---|---|---|---|---|---|
| 1 | 1 – 60 | 0.05 – 0.40 | 2 – 6 | 0 – 100 | **5** | |
| 2 | 1 – 60 | 0.05 – 0.40 | 2 – 6 | 0 – 100 | **7** | |
| 3 | 1 – 60 | 0.05 – 0.40 | 2 – 6 | 0 – 100 | **6** | |
| 4 | 1 – 60 | 0.05 – 0.40 | 2 – 6 | 0 – 100 | **6** | |
| 5 | 1 – 60 | 0.05 – 0.40 | 2 – 6 | 0 – 100 | **9** | |
| 6 | 1 – 60 | 0.05 – 0.40 | 2 – 6 | 0 – 100 | **8** | |
| 7 | 1 – 60 | 0.05 – 0.40 | 2 – 6 | 0 – 100 | **7** | |
| 8 | 1 – 60 | 0.05 – 0.40 | 2 – 6 | 0 – 100 | **7** | |
| 9 | 1 – 60 | 0.05 – 0.40 | 2 – 6 | 0 – 100 | **6** | |
| 10 | 1 – 60 | 0.05 – 0.40 | 2 – 6 | 0 – 100 | **6** | |
| 11 | 1 – 60 | 0.05 – 0.40 | 2 – 6 | 0 – 100 | **6** | |
| 12 | 1 – 60 | 0.05 – 0.40 | 2 – 6 | 0 – 100 | **6** | |
| 13 | 1 – 60 | 0.05 – 0.40 | 2 – 6 | 0 – 100 | - | Incorrect PNM ware |
| 14 | 1 – 60 | 0.05 – 0.40 | 2 – 6 | 0 – 100 | - | Incorrect PNM ware |
| 15 | 1 – 60 | 0.05 – 0.40 | 2 – 6 | 0 – 100 | - | Incorrect PNM ware |
| 16 | 1 – 60 | 0.05 – 0.40 | 2 – 6 | 0 – 100 | - | Exit staircase |
| 17 | 1 – 60 | 0.05 – 0.40 | 2 – 6 | 0 – 100 | - | Exit staircase |
| 18 | 1 – 60 | 0.05 – 0.40 | 2 – 6 | 0 – 100 | - | Exit staircase |
| 19 | 1 – 60 | 0.05 – 0.40 | 2 – 6 | 0 – 100 | - | Exit staircase |
| 20 | 1 – 60 | 0.05 – 0.40 | 2 – 6 | 0 – 100 | - | Neighbor corridor |

Table 3.1: Test results and the optimal ranges of the parameters

The results are not statistically meaningful and 40% occurrence of failure cannot be generalized. Indeed, the instances have been explicitly designed to challenge the method.

The performance of the algorithm is tested and analyzed applying different values of the parameters. The most important criteria are the efficiency and the complexity of the algorithm. The first criterion reflects how fast the initial location is determined, i.e. the number of iterations of the algorithm before localization. The second criterion reflects the memory usage, i.e. what amount of data is treated. Finally, we define optimal ranges for the four parameters, which provide the best performance of the algorithm.

### Threshold for turn detection

The detection of turns plays a major role in our methodology. The introduction of this threshold can be considered as a way to distinguish the turns from the basic changes of direction, discussed in chapter 3.2.1. However, this parameter must be chosen carefully, in order to detect all the turns in the trajectory. After the tests, for each trajectory we can define a range of values for this threshold which allows detecting the turns (Fig 3.29). Thus we can estimate an optimal range of 18° - 29° which is valid for most of the trajectories.

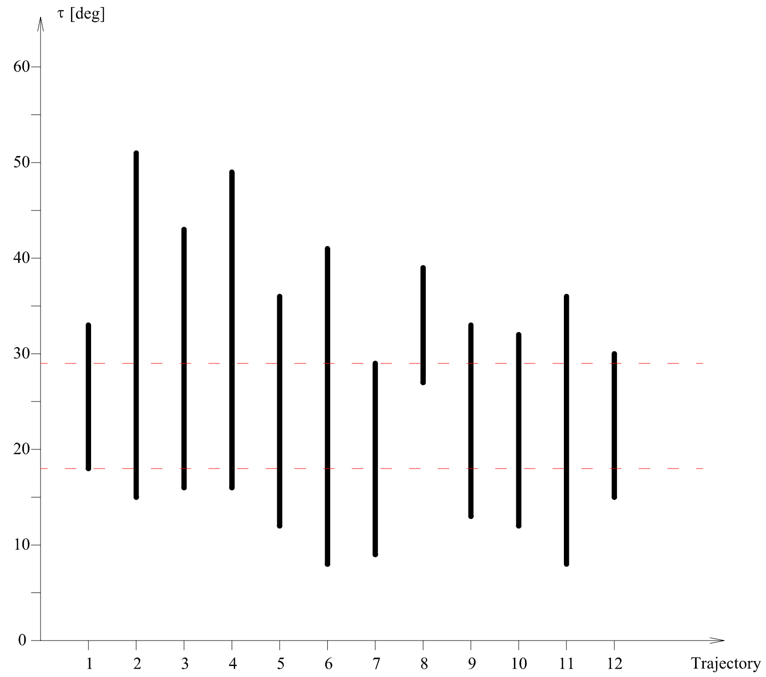
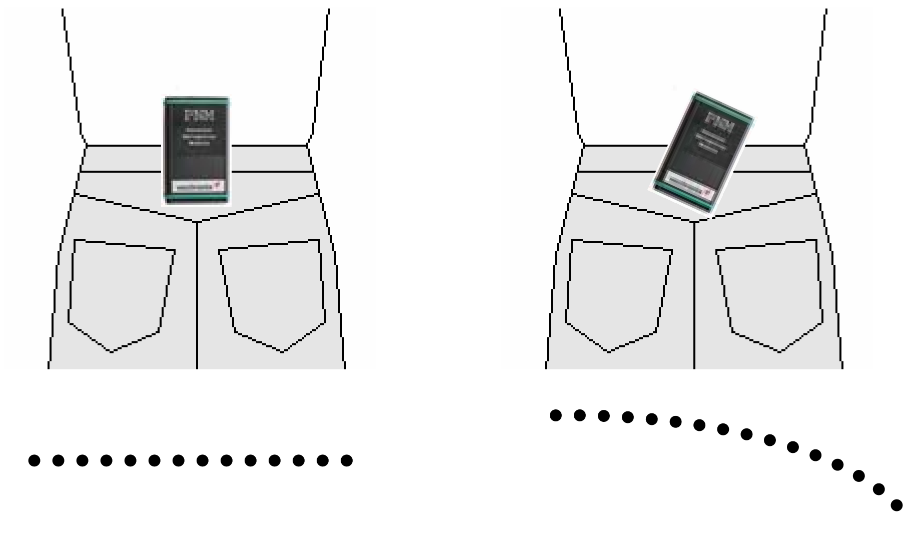



Figure 3.29: Optimal range (18° - 29°) marked with the dashed lines

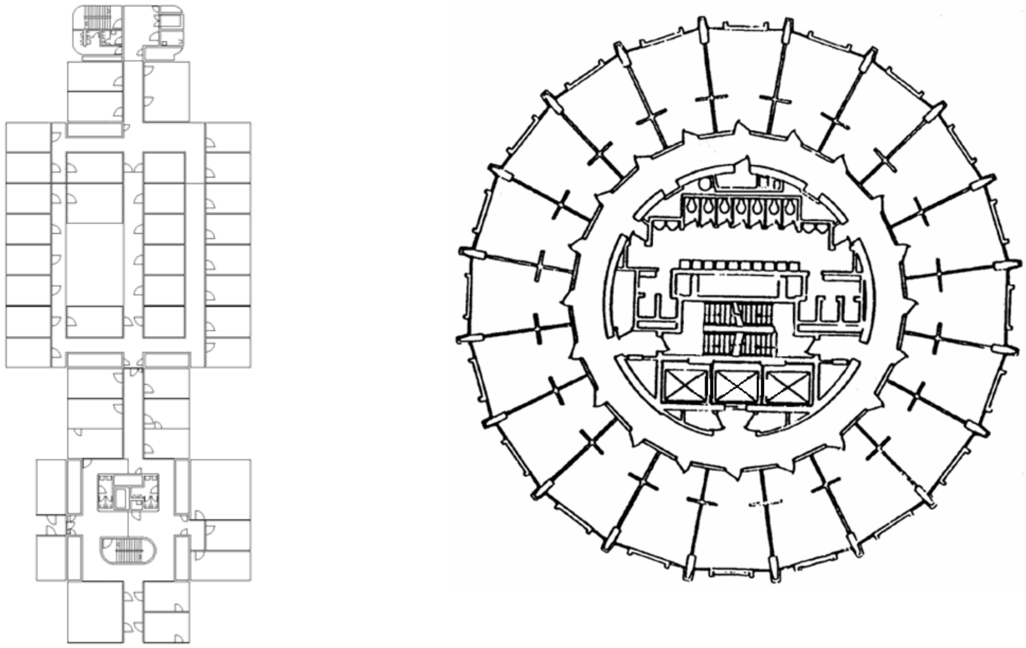
On Fig 3.29 one can notice that the values of $\tau$ for trajectory 8 make an exception from the optimal range. This trajectory needs rather higher threshold values to detect the turns. The reason is that the PNM was incorrectly worn by the person during the walk. That introduces an important drift in the measurements and we need to define a bigger threshold to avoid this influence and to detect the turns (Fig. 3.30).




a) b)

Figure 3.30: Straight walk performed with correctly worn PNM (a) and with incorrectly worn PNM (b)

The turns in a trajectory depend as well on the structure of the building. Most of the corridors in our playground, the EPFL, form normal crossings of 90°, which obliges the person to make right turns and facilitates the task of turn detection (Fig. 3.31a).



a) b)

Figure 3.31: Floor plans of buildings with different structures

However, we can see many buildings with a non standard structure (Fig. 3.31b), where one can not clearly define where the person makes a turn during the walk. In that context two main problems can arise: Is it possible to keep the same map representation of the building (link-node model) and do we need to make changes in the methodology? Some ideas on that problematic are presented in chapter 4.

**Threshold for speed variance**

The detection of vertical movements relies entirely on the determination of the state points where we use the speed variance. Even if the theoretical concept of that technique is reasonable, the tests show that it represents a very delicate point of the methodology. Fig. 3.32 shows the ranges of the speed variance for each test trajectory. We refer to the initial hypothesis that the person performs a normal walk. Nevertheless, it is not the case in trajectory 1 on Fig. 3.32, because the person has run while descending the stairs.

Several times when the user takes a staircase the state points are not detected, while in other cases state points are detected incorrectly. We can explain this again with the incorrect attachment of the PNM on the user's body. It must be tightly attached on the user's belt; otherwise the module swings during the walk and gives very erroneous measurements. Another error is the incorrect calibration before the test. It must be made in accordance with the factory instructions.

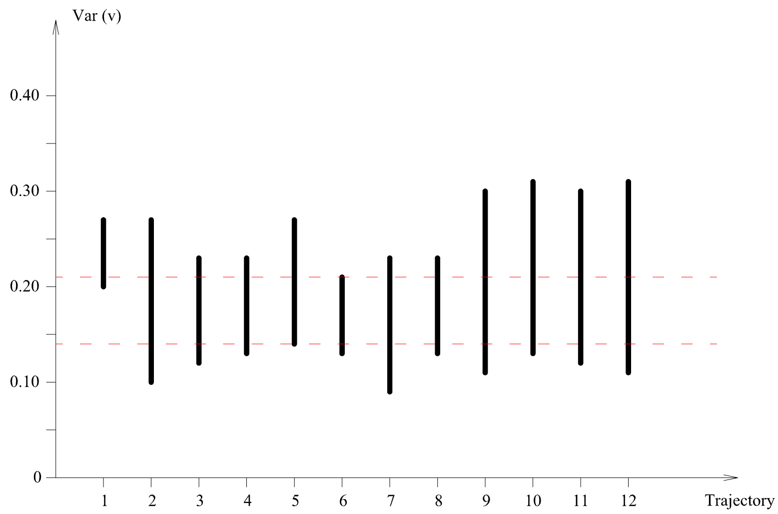



Figure 3.32: Optimal range of the speed variance (0.14 – 0.21) marked with the dashed lines

However, another major problem comes from methodological point of view. This technique to detect vertical movements works well in the case of elevators. That is, the detection of state points where the user stops and goes is easier than state points where user slows down or accelerates.

Taking a stair-case differs, because the user can change the speed anytime in the stair-case, typically by running. On the other hand, most of the staircases at EPFL (our playground) have doors. In order to access or leave the staircase the user must perform some sophisticated movements, including step back and turn around, while opening the door. At this stage of the research and with this reduced set of input information, such a delicate point of the methodology is not surprising. Considering the test results, we can see that the best value for speed variance is in the range of 0.14 to 0.21.

### Estimation threshold $\varepsilon$

This parameter affects the complexity and the efficiency of the algorithm. We can make the prior hypothesis that increasing $\varepsilon$ the complexity and the efficiency shall increase as well. So, we need to define an optimal value of $\varepsilon$ that provides minimal complexity and maximal efficiency of the algorithm. The complexity and the efficiency were evaluated for a range of $\varepsilon$ from 0% to 100%.

Fig. 3.33 shows an example that confirms the hypothesis made above. The augmentation of $\varepsilon$ means that more edges are involved in the computation, thus increasing the complexity. On Fig. 3.33 we can distinguish some parts of the graph where the complexity increases drastically (for $\varepsilon$ = 20 to 40 %) and other parts (for $\varepsilon$ = 40 to 70 %) where the complexity increases lightly. The reason is that the complexity depends on the number of edges treated in the update phase, but also on the number of neighbor edges treated in the prediction phase, which can differ from iteration to iteration.

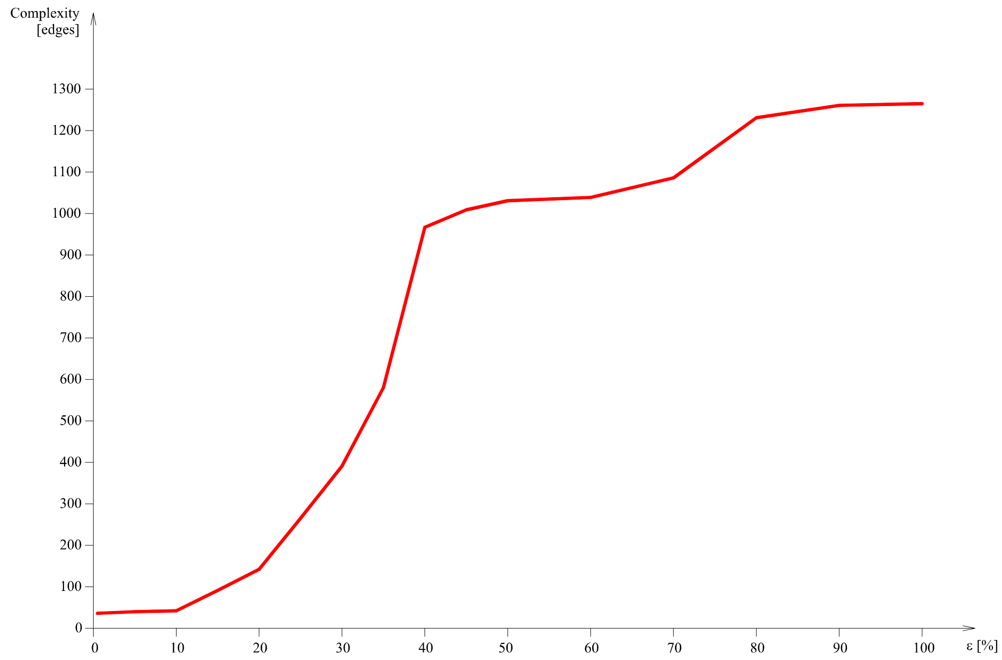



Figure 3.33: Increase of the complexity by augmenting $\varepsilon$. The complexity is presented with the number of treated edges.

The efficiency is expressed by the number of iterations, needed to localize the user. Augmenting $\varepsilon$, some of the tests show a slight increase of the efficiency (Fig. 3.34), and other tests show a constant efficiency (Fig. 3.35) up to a certain value of $\varepsilon$.

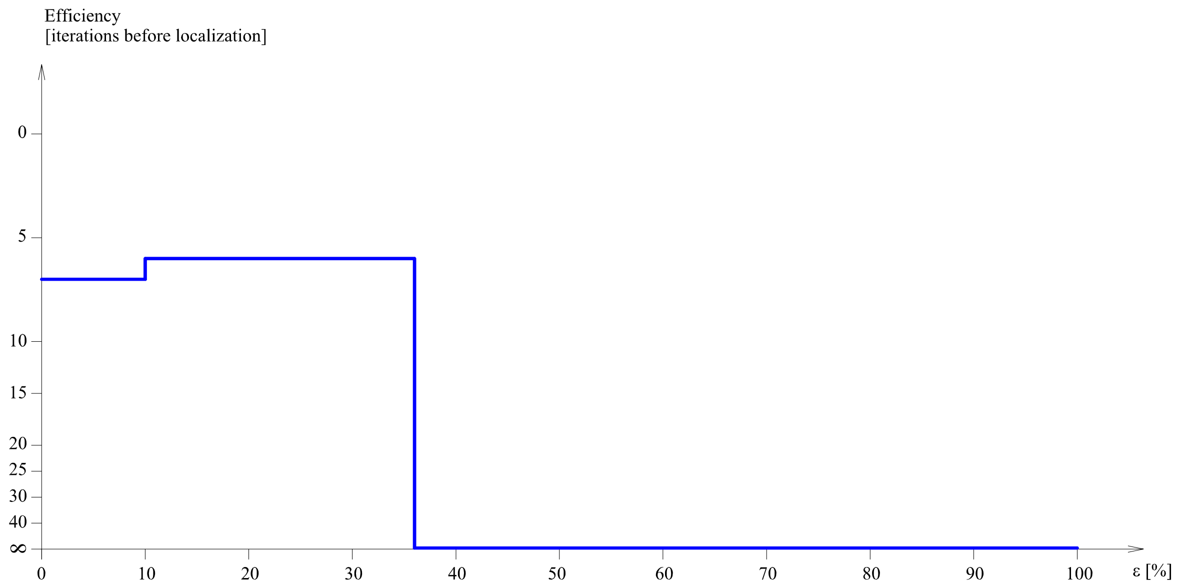



Figure 3.34: Slight increase of the efficiency when augmenting $\varepsilon$

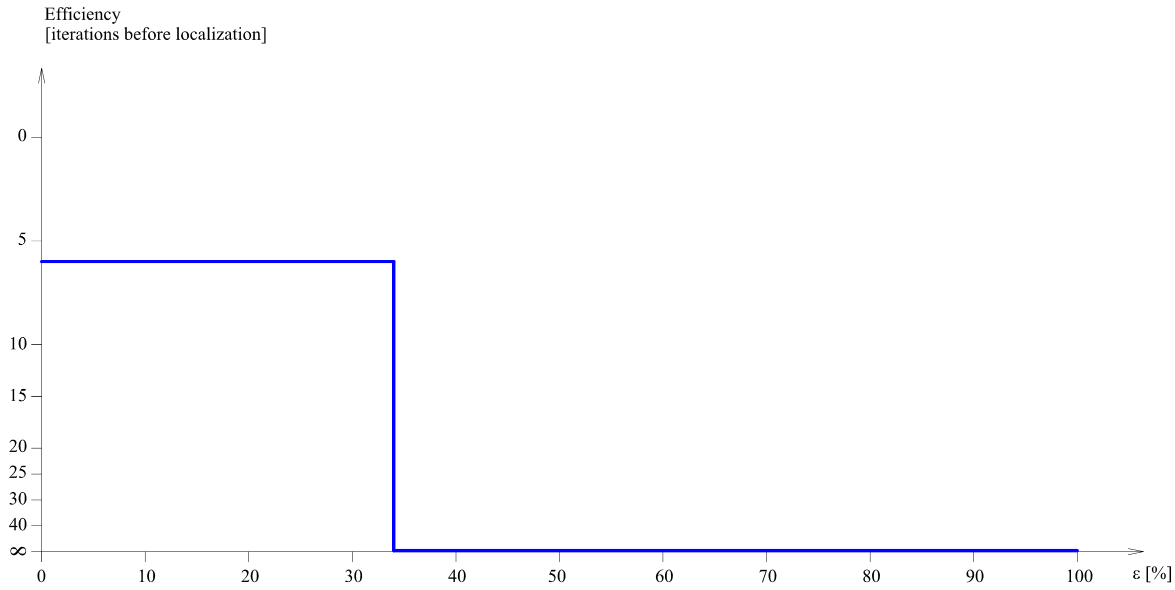



Figure 3.35: Constant efficiency when augmenting $\varepsilon$

However, we can remark a drastic drop of the efficiency after $\varepsilon \approx 35$ %. Basically, the algorithm fails when these values are used. In that range of $\varepsilon$, (35 % - 100 %), we can see that the algorithm is inefficient, i.e. there is no localization and the number of iterations grows to infinity. In that case, due to the relatively big value of $\varepsilon$, additional edges are considered in the estimation phase on each iteration (Fig. 3.36).

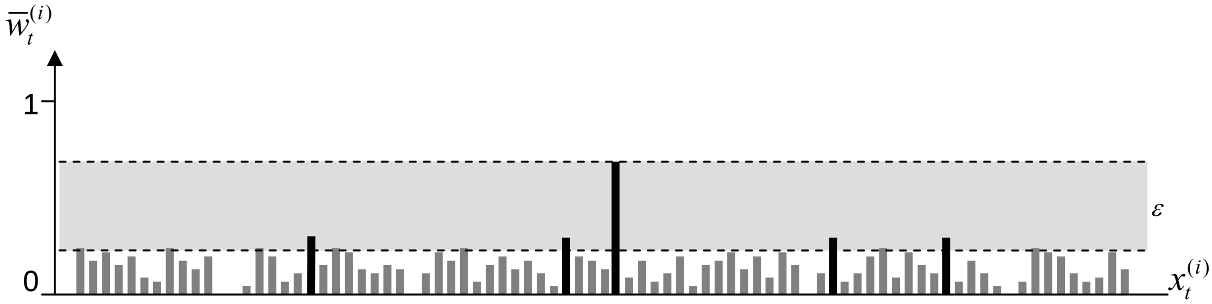



Figure 3.36: Augmenting $\varepsilon$ involves more edges in the computations

Thus the location edge can not be determined even if it possesses a much bigger weight than the other candidates. These additional edges implicate their neighbor edges in the next prediction phase. Thus the algorithm turns to infinity.

Another important reason for the drop of efficiency is that on each iteration the multimodal distribution from the update phase is replaced by a distribution of [0, 1] in the prediction phase (Fig. 3.37). Thus, certain edges will have weight 0 after the next update phase and we may reject the potentially correct edge in the estimation.

This approach is imperfect because changing the distribution we qualify all the estimated edges as user's location without keeping count on the information from the last update. That information can be crucial for the estimation. However, it will show efficiency when huge databases (large building complexes) are treated. Considering the tests, we can qualify the algorithm as efficient for $\varepsilon < 35$ %.

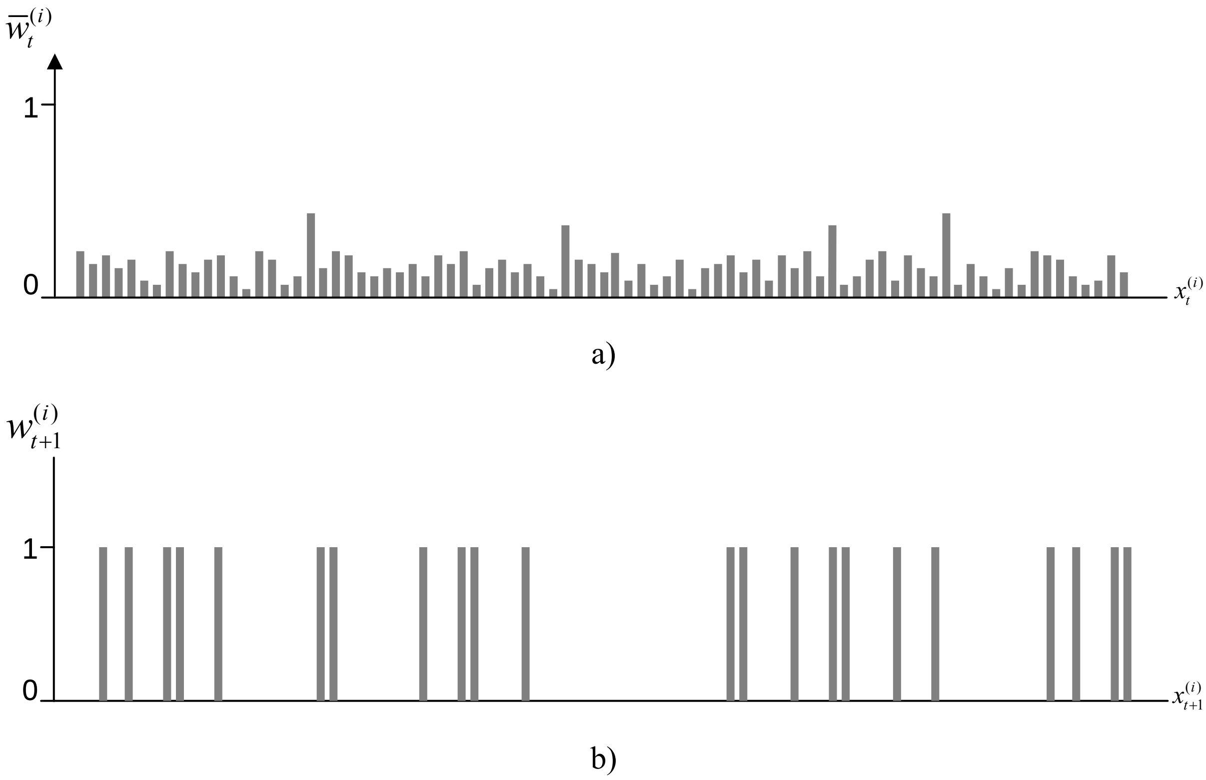



Figure 3.37: Change of the multimodal distribution (a) with distribution [0, 1] (b)

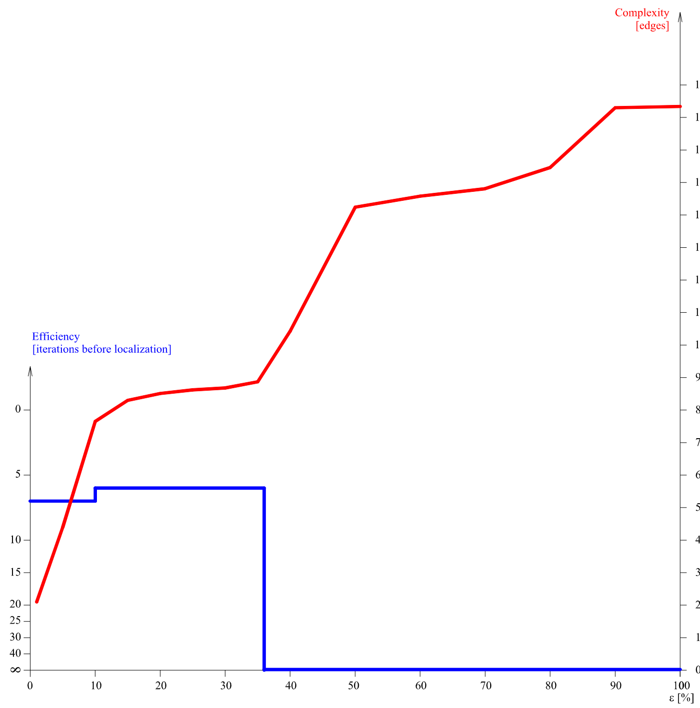



Figure 3.38: Definition of the optimal value for $\varepsilon$. In this example the algorithm is efficient for $\varepsilon < 36$ %.

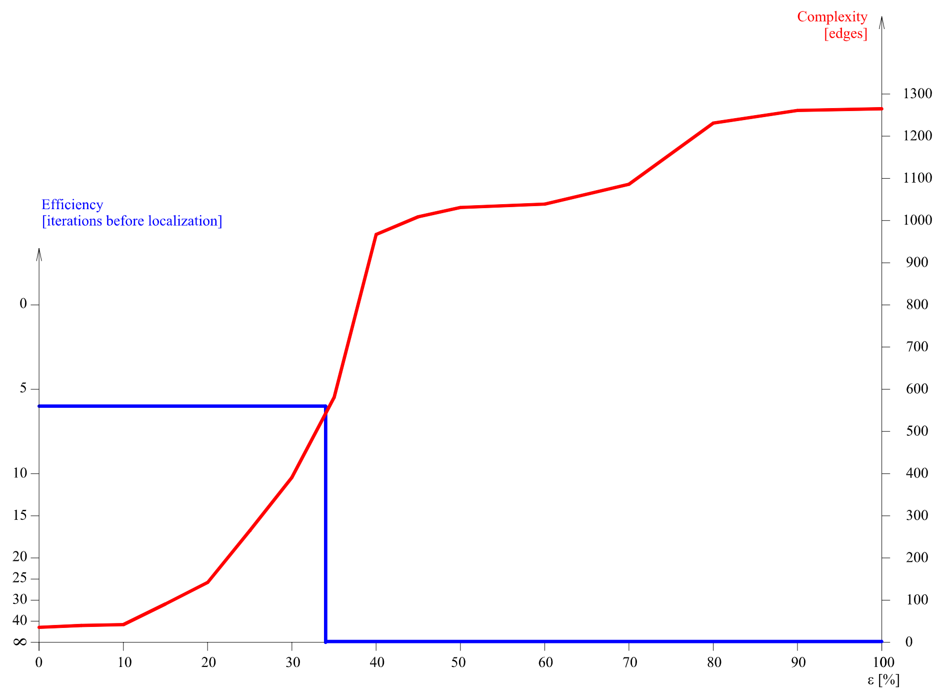



Figure 3.39: Definition of the optimal value for $\varepsilon$. In this example the algorithm is efficient for $\varepsilon < 34$ %.

For the definition of an optimal value of $\varepsilon$ we have superposed the graphs for the efficiency and the complexity (Fig. 3.38 and Fig. 3.39). If we consider only the range of $\varepsilon$ for which the algorithm is efficient, we can take as optimal value of $\varepsilon$ the one which gives the smaller complexity.

In the case of Fig. 3.38 the algorithm shows best performance for $\varepsilon = 0\%$. That corresponds to the case where we estimate only the edges with maximal weight as user's location. In the case of Fig. 3.39 the algorithm shows best performance for $\varepsilon = 0\%$ to 10% with a slide increase of the complexity until $\varepsilon = 20\%$. The main reason for the algorithm to show its best performance for small $\varepsilon$ is the use of [0, 1] distribution in the prediction phase, as discussed above.

### History of the movement

The history of movement contains crucial information for the estimation of the user's location. The history is represented by the number of segments of the polyline (trajectory) taken into account during the computation. As discussed in chapter 3.2.2, we need to find the placement of the polyline in the contents of the graph.

If we consider all segments of the polyline (trajectory) the estimation at moment $t$ shall be most robust. However, it will cost a lot of memory usage, i.e. the complexity will increase. Therefore, during the computation we consider only a part of the polyline. Besides the last segment we treat $n$ segments back, which represent the history.

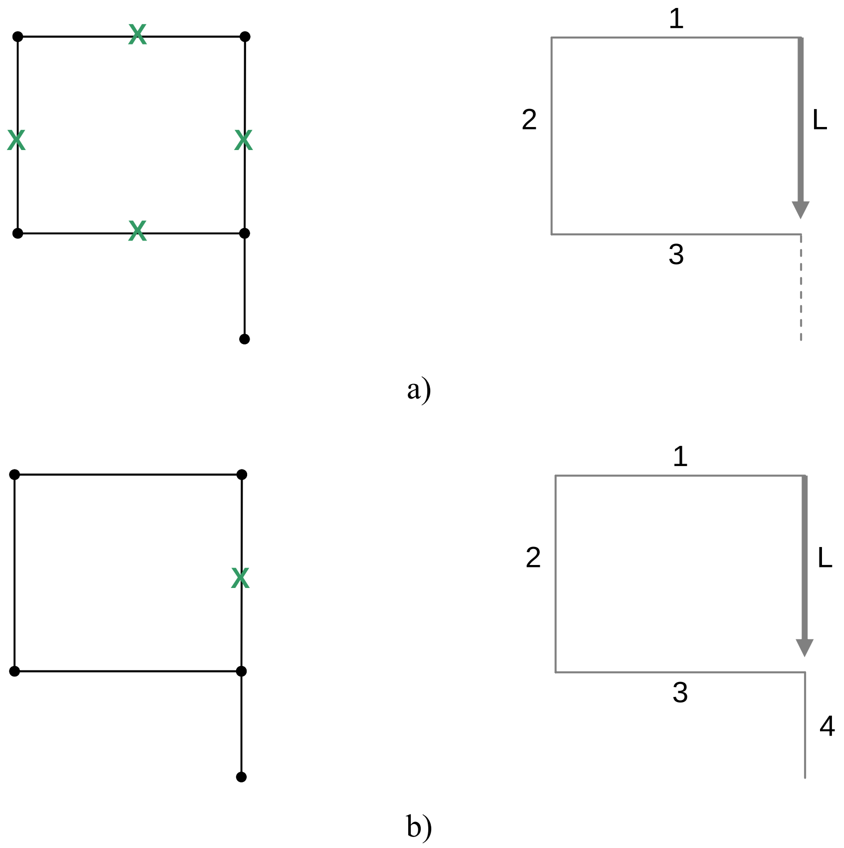



Figure 3.40: Finding the placement of the segment "L" of the polyline (right) in the contents of the graph (left), marked with green "X".

Fig. 3.40 shows an example of finding the placement of the segment "L" of a polyline in the contents of the graph, taking into account different number of segments as history. A history of 3 segments (Fig. 3.40a) is not enough for finding the unique place of "L" on the graph. With a history of 4 segments (Fig. 3.40b) the place of "L" is found. Our tests show that a history of at least six segments needs to be taken in order to localize the person.

**Distance scale correction**

A very important element of the algorithm is the correction of the measured distance. Several preliminary tests show that the distances computed from the raw measurements are influenced by a scale factor of about 1.06. Empirically, we have determined a correction constant and we have made a scale correction on every iteration of the algorithm.

This correction shows its importance in the last test trajectory (Table 3.1). The trajectory was treated without performing a distance scale correction. As a result, the location of the person is estimated in the corridor neighboring the correct one.

**Sophisticated movements**

One of the main tasks in our approach is the detection of the critical movements like turns and change of the floor. The performance of the algorithm can be hampered by sophisticated movements like stop to talk with someone, entry in a room to take something, etc.

The technique to detect obstacle avoidance is efficient when there is one obstacle on the way. However, avoiding several obstacles at the same time is not detected, typically when crossing many people in a corridor.

We can not restrict the freedom of movement of the person. However, we need to mention as a general remark to the methodology, that it is less robust when “sophisticated” (unusual) trajectories are performed, because they do not comply with the methodological assumptions.

### 3.2.6. Conclusions

The method of initial localization is based on the association of geometric and topologic information from both data sources (trajectory and map database). It is different from the classical map-matching techniques where the initial (preceding) position of the user is known. Another difference is that the association criteria are based on the likelihood of the geometry and not on the proximity of the trajectory to the elements of the graph. This is explicited by the fact that the graph and the trajectory are defined in different coordinate systems.

Following the concept for autonomous localization the process uses inertial measurements only and information from the map database. There are two main assumptions for the process of localization: the person walks normally and the trajectory is made in an area covered by the map database (the graph). That is, the person must not leave the area represented by the graph. The graph is not a realistic representation, since it presents a constraint to the movement of the person. Another representation could be the definition of zones with different probabilities for the passage of people.

The efficiency of the algorithm depends as well on the volume of the map database. A big database can contain information about several buildings, e.g. a university campus. In order to facilitate the localization process additional information about the user’s location can be given. For example, knowing that the person is in the “Architecture” building only the corresponding part of the database can be taken.

The Bayesian inference is chosen in this research because it is very effective in the treatment of multimodal non-Gaussian distributions, which is relevant for the initial localization presented here. Before estimating the location edge, the output of the inference is a multimodal distribution, which is important property of that approach. This corresponds to the fact that at a certain moment *t* several edges can be estimated as user’s location. We need to keep count on all possible estimated edges, i.e. to maintain the information of the multimodal distribution.

Another advantage of the Bayesian approach is the possibility to combine the input data. Different types of measurements (distances, angles) and different types of information (geometric and topologic) can be used as input to the inference. The core of the Bayesian approach is the computation of the likelihood, reflected by the specific weights of the edges. The measurement update is the phase where the input data from different sources are treated. This advantage allows the implication of additional data, i.e. measurements from other positioning systems like GPS and WiFi. Although the method discussed here is developed as an autonomous technique for localization, the statistical approach allows the use of external information when it is available.

## 3.3. Continuous localization

As mentioned above the initial localization aim at finding the edge in the graph occupied by the person, called the *location edge*, and person's orientation on that edge. As result of that operation we consider the transformation of the polyline (the trajectory) from the user's coordinate system to the coordinate system of the graph.

Now we need to determine where exactly on the location edge is the person. Contrary to the initial localization in the continuous localization user's location is not presented as edge but as a point, named for simplicity *location point*. That point is considered as a part of the edge and will be estimated using measurements on each step.

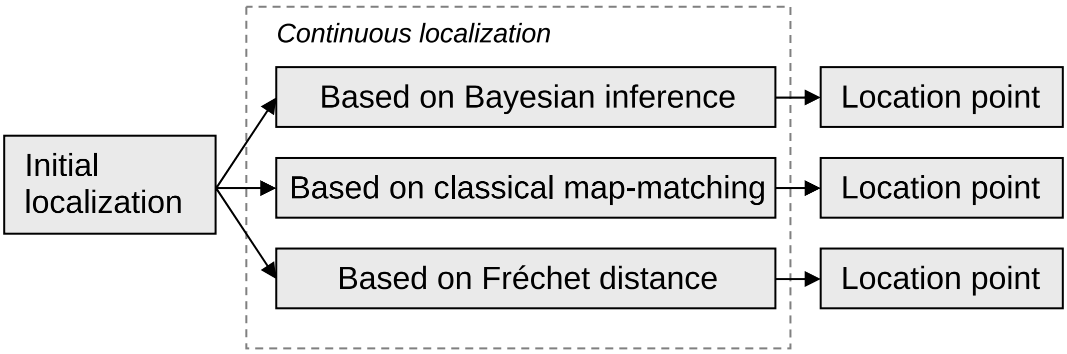



Figure 3.41: Continuance of the localization process after the initial localization

In this chapter (3.3.) three different methods for continuous localization are discussed. Each method is based on the assumption that the location edge is known, as well as the orientation of the person of that edge. The three methods are independent and can be applied individually, taking into account the output information from the initial localization method (Fig. 3.41). Nevertheless, it is possible to combine some of those three methods in order to assure additional control on the positioning.

### 3.3.1. Continuous localization based on Bayesian inference

**Theoretical formulation**

In order to assure a continuous localization process we need to have the location edge determined at every moment. Knowing the location edge the location point at moment *t* is fixed on it. Thus the problem can be subdivided in two parts: determine the location edge and estimate the location point.

The determination of the correct edge is based on the Bayesian inference, but the methodology is different from that in the initial localization. First, the time discretization of the process reflects the acquisition of new measurements on every step (Fig. 3.42) and not on the definition of every polyline segment. Second, the likelihood computation is based on the distance and heading of every stride instead of the polyline parameters.

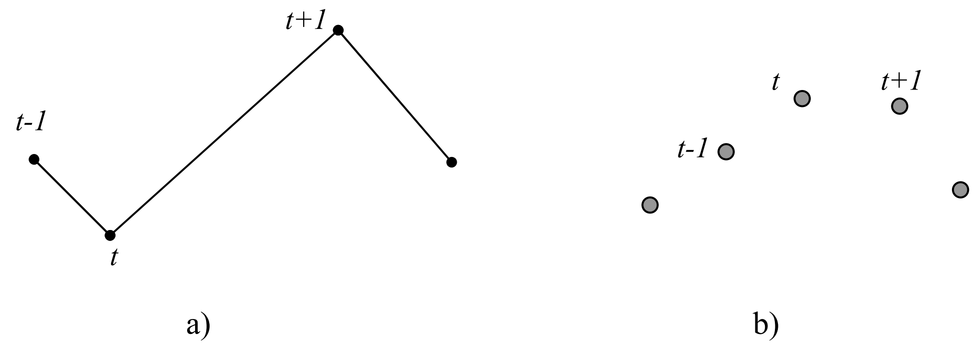



Figure 3.42: Time discretization of the initial localization (a), and continuous localization (b)

However a problem arises when step from one edge to another and on the crossroads, where a choice must be made between several candidate edges. Fig. 3.43 shows three neighbour edges and their probabilities to be the location edge depending on the passed distance.

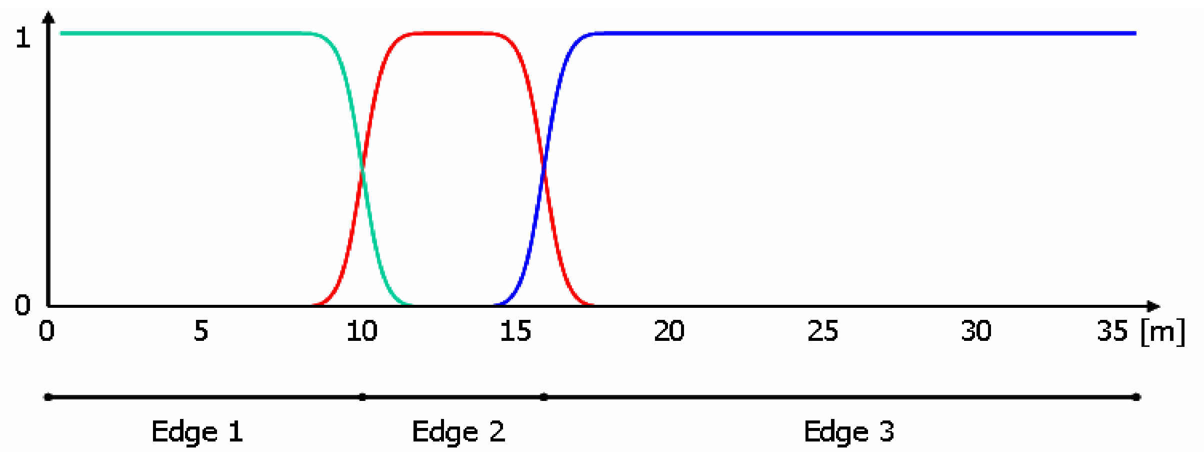



Figure 3.43: Probability of 3 neighbour edges

The moment of stepping from one edge to another is of great importance in this approach. At this moment we start to accumulate the distance of the strides from the beginning of the edge.

$$D_k = \sum d_k \tag{3.23}$$

where $k$ is used to count the steps on the edge.

The accumulated distance $D_k$ is used to estimate the location point. Thus at moment $t$ user's location is defined by a point fixed on the location edge on distance corresponding to the accumulated distance (Fig. 3.44).

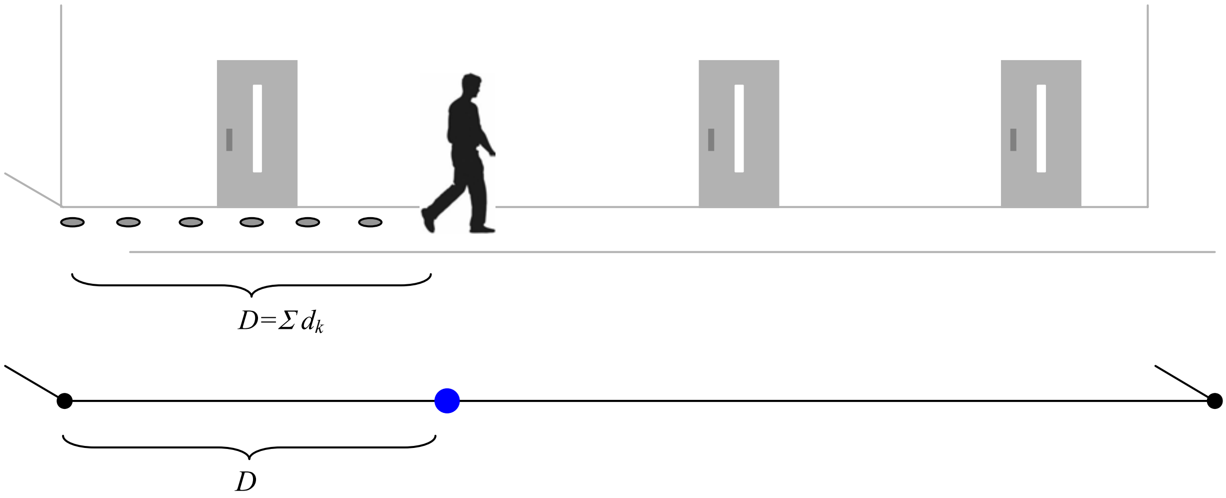



Figure 3.44: Estimation of the location point on distance $D$ from the beginning of the edge

When the user steps on the next edge the accumulated distance is set to zero and the counter $k$ is restarted ($k = 0$).

**Algorithm**

The main task then is to determine the edge on which the location point will be fixed. Applying the Bayesian approach the problem of continuous localization is solved in two phases: prediction and update.

The update phase consists in the computation of specific weight $w_t^{(j)}$ for each candidate edge $e^{(j)}$ where $j$ is the number of the candidate edges at moment $t$. That weight reflects the probability for the candidate edge to be the location edge. It is based on the computation on the following residual:

$$\gamma_t^{(j)} = A(e^{(j)}) - \sum \left( \hat{r}_t - \hat{r}_{t-1} \right) \tag{3.24}$$

Here $\hat{r}_t$ is the stride heading after the transformation of the trajectory in the coordinate system of the graph, and $A(e^{(j)})$ is the azimuth of the candidate edge.

Then the weight of each candidate edge is computed as:

$$w_t^{(j)} = 1 - \frac{\gamma_t^{(j)}}{\sum_{j=1}^{G} \gamma_t^{(j)}} \tag{3.25}$$

where *G* is the set of the candidate edges.

The update phase is performed on every user's step. Note that the raw measurements are treated so as the trajectory is transformed to the coordinate system of the graph. We estimate the location edge at moment *t* as:

$$\hat{e}_t = e_t^{(j)} \left| w_t^{(j)} \geq \left( w_t^{(MAX)} - \varepsilon \right), \quad j = 1, \ldots, G \right. \tag{3.26}$$

where $\hat{e}_t$ is the estimated location edge and $\varepsilon$ has the same meaning as discussed in chapter 3.2.4. This phase is followed by the estimation of the location point (Fig. 3.44).

The prediction step consists in the choice of candidate edges. An essential difference from the methodology of the initial localization is that the prediction step is not repeated every time after the update step. As for the initial localization, the neighbour edges are considered as candidates. However, the question is when to perform the prediction. For the time when the user is walking in the middle of the location edge we can easily estimate the location and to fix the point. The idea is that the prediction must be made when the person is approaching a junction (e.g. crossroad).

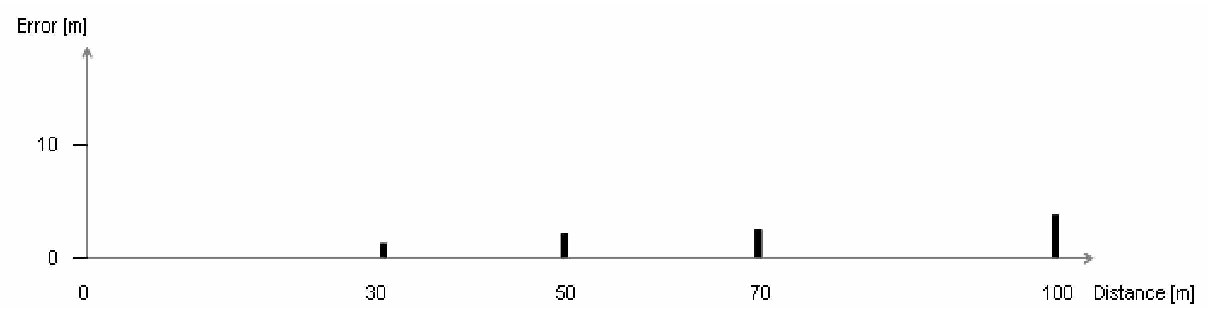



Figure 3.45: Positioning error of the system (4% of the passed distance).

Regarding the passed distance on the edge we can decide when the user approaches the end of the edge. In that moment the distance *D* comes near to the length of the edge. That means the estimated location will be close to the end of the edge. For that we use the technical characteristics of the PNM. According to the user's manual of the module the positioning error of the system is 4% of the passed distance (Fig. 3.45). The criterion of proximity to the end of the location edge is based on that error. Thus we consider the person is approaching the end of the edge if:

$$D_k - L(e^{(j)}) \le 4\%\left(D_k\right) \tag{3.27}$$

At that moment there can be several candidate edges $e^{(j)}$, $j=1,\dots,G$. The location edge is estimated after the update phase, discussed above. The flowcharts of the update phase and the estimation phase of the algorithm are shown on Fig. 3.46 and Fig. 3.47.

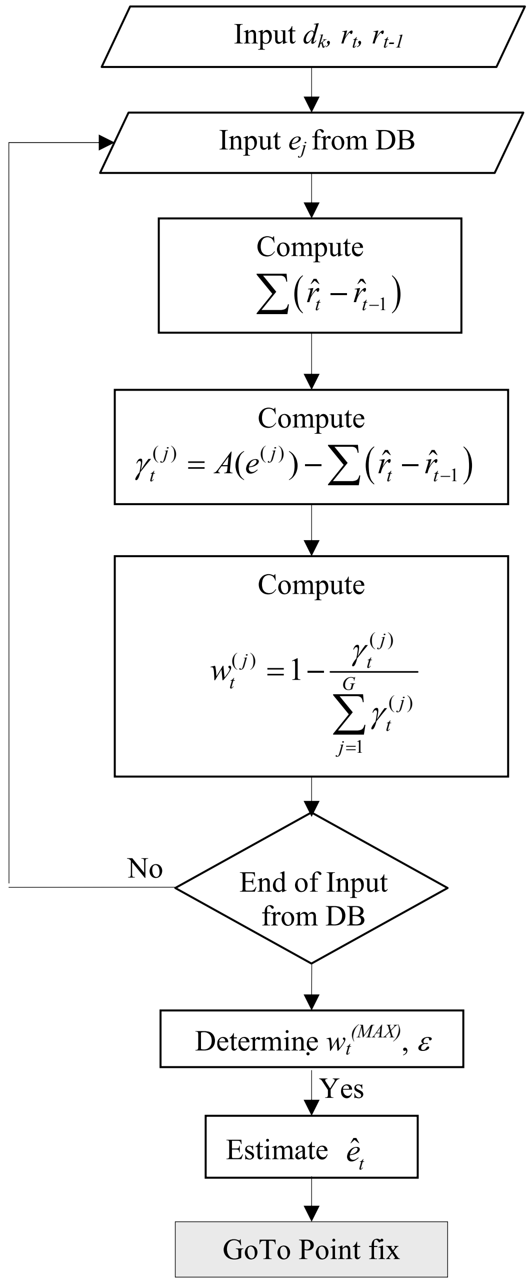



Figure 3.46: Update phase of the continuous localization

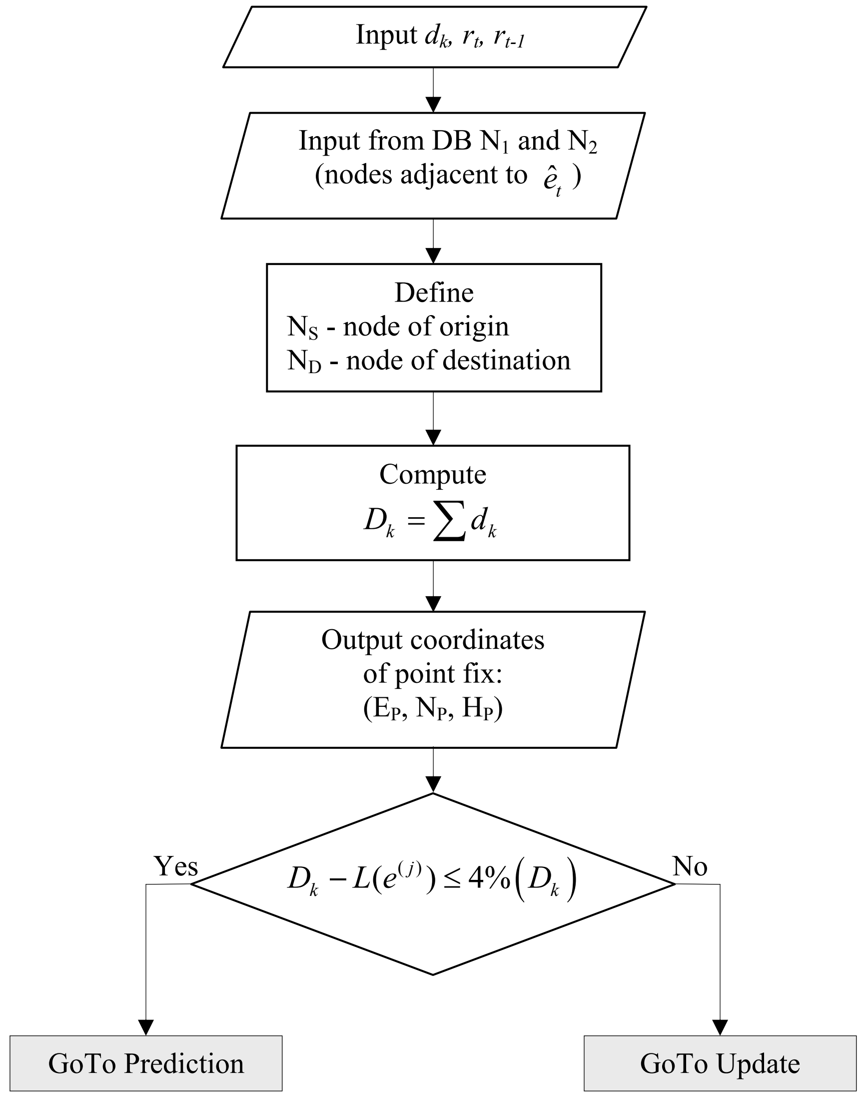



Figure 3.47: Estimation of the point fix with its coordinates ($E_P$, $N_P$, $H_P$), where $H_P$ refers to the floor of the building

**Tests, results and analysis**

This algorithm for continuous localization is tested with many trajectories and the analysed aspects are the accuracy of the localization, robustness and continuity of the process.

The precision of the location point fix depends directly on the precision of the estimated distance *D* (3.23). In order to test that precision we compare the length of the location edge with *D* in the moment when the person is at the end of the edge. Table 3.2 shows the results of this comparison for some of the edges passed in the test trajectories.

From the results in Table 3.2 we can easily distinguish the biggest error of -1.19m. However we can not say that a bigger error corresponds to bigger distance.

| Length of edge (m) | Estimated *D* (m) | Error (m) |
|---|---|---|
| 14.96 | 15.00 | 0.04 |
| 3.60 | 3.90 | 0.30 |
| 37.02 | 35.83 | -1.19 |
| 6.93 | 7.20 | 0.27 |
| 3.63 | 3.90 | 0.27 |
| 16.08 | 15.98 | -0.10 |
| 14.23 | 14.40 | 0.17 |
| 8.69 | 8.90 | 0.21 |
| 16.10 | 15.80 | -0.30 |

Table 3.2: Comparison of the length of the edge and the corresponding estimated distance *D*

The estimation of the distance *D* depends on the moment when we start to accumulate it. In other words it depends on the first user's step on the edge. The point fix of that step is estimated in the first update after the prediction phase.

In fact it is possible that the point fix does not correspond to the physical position of the person. The reason is that we perform the prediction earlier then the user has arrived on the junction.

The explanation is given regarding the criterion in (3.27). Normally, that criterion is fulfilled before the user has arrived on the junction, when $D_k < L(e^{(j)})$ but their difference is less then 4%. Thus the prediction is performed in a moment when the person is almost on the junction, but still on the previous edge. The next step is taken into account in the update phase, even if it is made before to step on the new edge. Thus that user's step is included in the computation of the distance *D* which causes the error in Table 3.2. That problem mostly arrives in the crossroads, where the user turns in another direction. In that case more than one step can be included in the computation of *D*, without being performed on the location edge.

Another question is if the localization accuracy is sufficient. Nowadays the efforts in the domain of pedestrian navigation are pointed to achieve a localization accuracy of 3 meters [Abwerzger et al. 2004] or even 1 meter [Usui et al. 2005]. Taking into account the errors in Table 3.2 we can say that the method provides a sufficient accuracy, since the distance *D* is directly referred to the location estimation.

The robustness of the continuous localization process depends on the performance of the algorithm in critical situations [Quddus et al. 2006], [Pyo et al. 2001]. Typical example for such situation is the pass through a junction. There a decision must be taken between several candidate edges and a reliable estimation of user's location is expected.

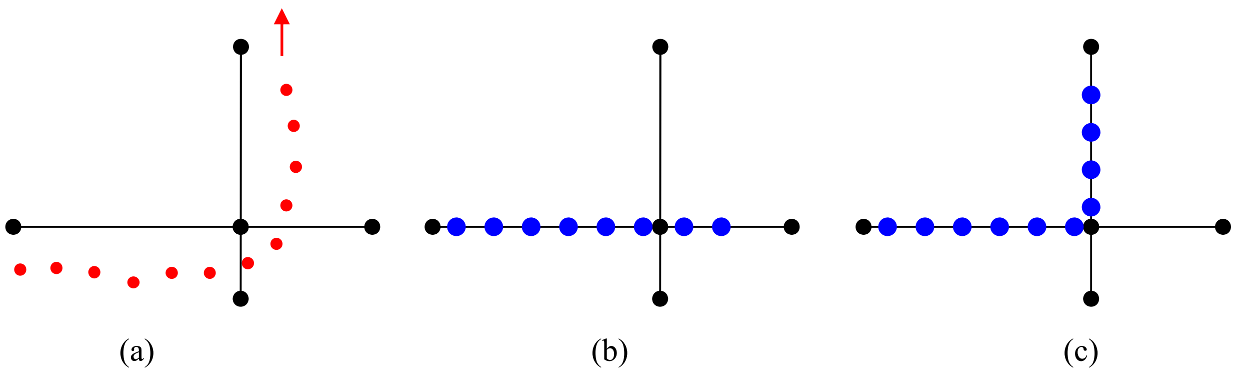



Figure 3.48: Testing the robustness on the junctions

A part of trajectory passing through a crossroad is shown on Fig. 3.48a. Even if on that crossroad the direction of walk is changed the first two point fixes after the junction are estimated on the edge in straight direction (Fig. 3.48b). After acquisition of information on the next steps the edge on left is defined as location edge and the point fixes are correctly estimated (Fig. 3.48c).

The pass through a junction is not the only critical situation for the localization process. Another point to check the robustness of the algorithm is its performance when the measurement input possesses a gross error. Indoors, there exist many disturbing factors for the inertial sensors (metallic constructions, electric installations, etc.). Their influence on the inertial measurements can cause a gross error particularly in the heading measurement [Ladetto et al. 2002].

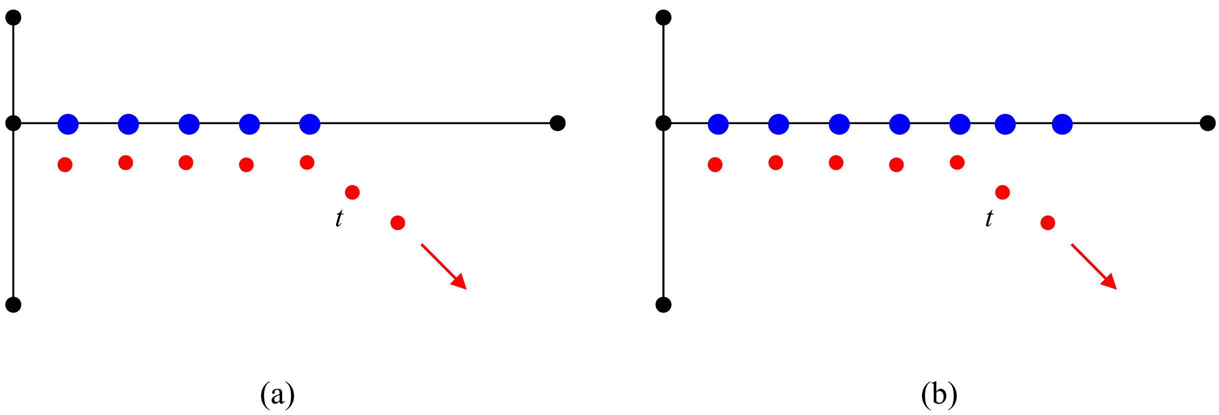



Figure 3.49: Testing the robustness in case of gross error in the heading measurement

In order to check the robustness in that case we have introduced a gross error (25°) in the heading in the middle of a straight walk (Fig. 3.49a). The algorithm shows robustness even in that case. Taking into account the topological information and the last point fix estimation we can decide that at moment $t$ there is only one candidate edge and the location point is fixed on it (Fig. 3.49b).

The continuity of the entire localization process depends on the passage from initial to continuous localization. The latter is activated right after the location edge ($\hat{e}_t$) is found.

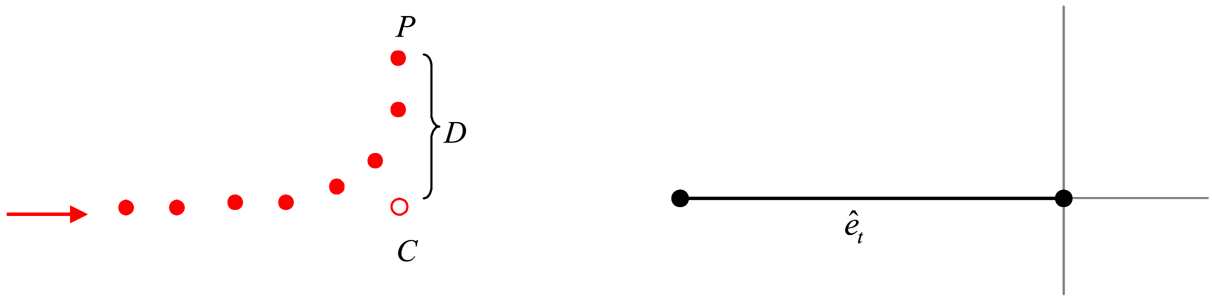



Figure 3.50: Illustration on the activation of the continuous localization process

In fact, the location edge $\hat{e}_t$ is determined after the person has physically left it. On the example on Fig. 3.50 that determination depends on the moment of definition of the last critical point *C* (refer to chapter 3.2.2.). At this moment the person is in point *P* and has already walked certain distance on the next edge. Thus the continuous localization process will start with small delay. So we define:

$$
\begin{aligned}
D &= |CP| \\
\gamma_t^{(j)} &= A(\hat{e}_t) + \theta
\end{aligned}
\tag{3.28}
$$

where *j* is the number of the candidate edges, $A(\hat{e}_t)$ is the azimuth of the location edge and $\theta$ is the accumulated change of direction. The activation of the continuous localization starts at this moment using the defined values for *D* and $\gamma$. As candidate edges, $e^{(j)}$, the neighbours of the location edge are chosen.

In terms of continuance, the processes of initial localization and continuous localization will both continue working in parallel, thus keeping count on the history of walk which will assure a control for the location edge estimation.

## 3.3.2. Continuous localization based on classical matching techniques

One of the tasks of this research is to analyse the performance of the existing map-matching techniques when they are applied in the context of the pedestrian navigation. We must consider the liberty of movement of the person and the capability to perform vertical displacements. We apply a weighting system based on the application of the matching techniques reviewed in chapter 2.3. That system is pointed at the known problem of determination of the location edge among several candidates, and at the estimation of user's location on it.

**Theoretical formulation**

Here we make reference to two existing matching techniques, point-to-point and point-to-edge. We discuss the choice of one of these techniques depending on the situation.

Consider the location point $P_{t-1}$ at moment $t-1$ is known ($E_{t-1}$, $N_{t-1}$, $H_{t-1}$), where $H_{t-1}$ states for the floor (Fig. 3.51). At moment $t$ we receive the measurements which define the user's position as $P_t^{'}$, called for simplicity *raw position* The task is to determine the location point $P_t$ at moment $t$, by matching $P_t^{'}$ to an element of the graph. Generally we consider the case of matching the raw position to an edge. However, in some cases matching to node must be applied, which is discussed further. [Büchel 2004].

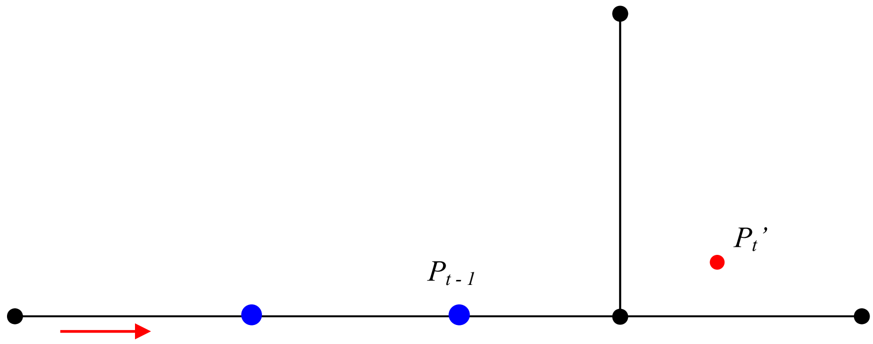



Figure 3.51: Illustration on the activation of the continuous localization process

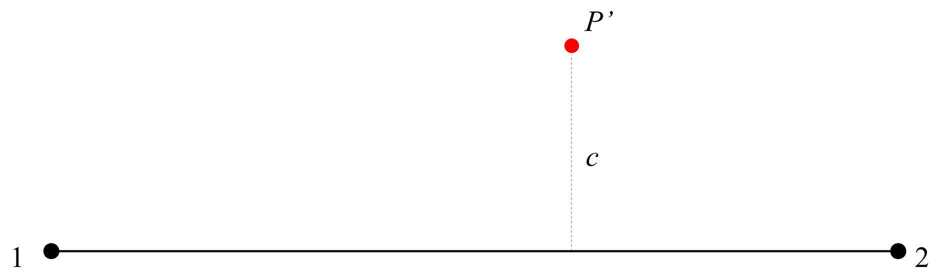
The problem consists in the determination of the edge of the graph to which $P_t^{'}$ must be associated. This problem becomes most delicate in the junctions, where the decision must be made among several candidate edges $e^{(j)}$, where $j$ is the number of the candidate edges at moment $t$. Regarding the raw measurements at moment $t$ and the topological information we can evaluate the probability of each candidate to be the location edge.

In this approach that probability is evaluated considering different criteria. The fulfilment of each criterion is reflected by a specific weight. Thus we apply a weighting system in order to classify the candidate edges and to decide which one to valorize [Gilliéron et al. 2004].

The criteria that an edge must meet are the following: criterion for proximity; criterion for orientation and criterion for topological connectivity.

- criterion for proximity

The proximity of a point to an edge is reflected by the perpendicular distance from $P^{'}$ to the edge (Fig. 3.52).




$$c = \sqrt{\frac{\left((y_1 - y_2)^2 x_p + (x_2 - x_1)^2 y_p + (x_1 y_2 - x_2 y_1)\right)^2}{(y_1 - y_2)^2 + (x_2 - x_1)^2}}$$

Figure 3.52: Evaluating the proximity of the raw point $P^{'}$ to an edge

Therefore the weight for proximity to the edge is defined as:

$$w_P^{(j)} = \frac{1}{c^{(j)}} \tag{3.29}$$

Where $c^{(j)}$ is the distance between $P^{'}$ and the edge $e^{(j)}$, and $w_P^{(j)}$ is the weight for proximity. As the perpendicular distance $c$ decreases the proximity increases.

- criterion for orientation

The second criterion controls whether the candidate edge has a similar orientation with the direction of the last stride $\overrightarrow{P_{t-1}P_t^{'}}$ (Fig. 3.53). For that we take into account the difference between the azimuth of $\overrightarrow{P_{t-1}P_t^{'}}$ and the azimuth of the candidate edge, marked as angle $\omega$.

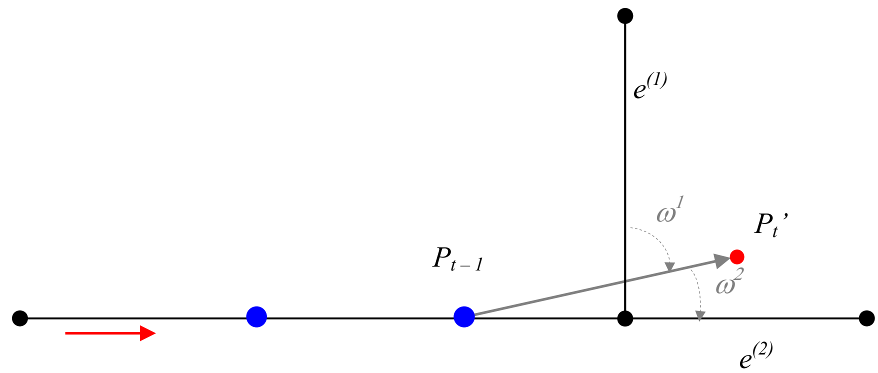



Figure 3.53: Compare the orientation of the last stride and the candidate edge

We decide that the orientation of the stride is more important for the estimation of the location edge than the proximity. Therefore we introduce a parameter $K > 1$, in the computation of the weight for orientation. We write:

$$w_O^{(j)} = K\cos\omega \tag{3.30}$$

Where $w_O^{(j)}$ is the weight for orientation. The parameter $K$ is the introduced parameter. The choice of such parameter depends on the topology of the graph and on the precision of the map database [Quddus et al., 2003].

- criterion for topological connectivity

That criterion verifies whether the candidate edge is neighbour of the location edge at moment $t$. Fig. 3.54 shows an example where decision must be made between three edges. The edges $e^{(1)}$ and $e^{(2)}$ have the same weights for proximity and orientation. However, there is no topological connection between $e^{(2)}$ and the location edge with point $P_{t-1}$. Thus the candidate $e^{(1)}$ is excluded from the estimation.

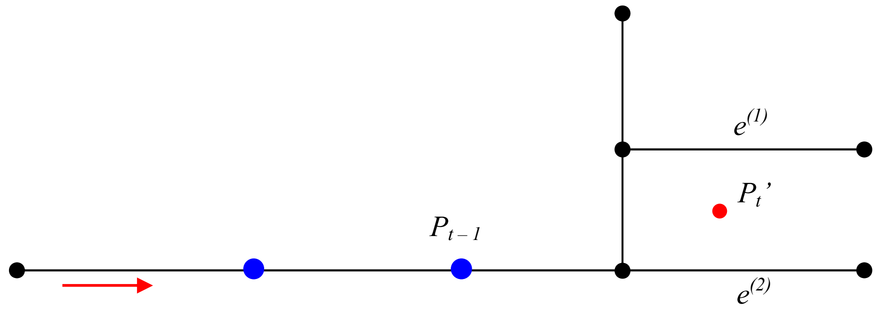



Figure 3.54: Control for topological connectivity for $e^{(1)}$ and $e^{(2)}$

For the weight of topological connectivity we write:

$$w_C^{(j)} = \begin{cases} 1 & ,\hat{e}_t \quad is \quad neighbor \quad of \quad e_{t+1}^{(j)} \\ 0 & ,\text{otherwise} \end{cases} \tag{3.31}$$

Finally the estimation of the total weight for each candidate edge is made by multiplying the three weights:

$$w_T^{(j)} = w_P^{(j)} \cdot w_O^{(j)} \cdot w_C^{(j)} \tag{3.32}$$

The edge with highest total weight is considered as location edge at moment $t$. Then, the location point at moment $t$ is fixed by projecting the raw position $P_t^{'}$ on the location edge.

Besides point-to-edge matching there are situations where we can apply point-to-point matching. These are the cases when the user enters or leaves the elevator or staircase, defined in the graph as vertical edges. Since the vertical displacements are considered as change of the floor all intermediate points during the passage from one floor to another can be matched to the nodes of the vertical edges. The point-to-point matching consists in the computation of the distance between $P_t^{'}$ and the candidate nodes. It is very unstable technique, because it simply associates the raw position to the closest node. However, if we consider only the nodes of vertical edges, the estimation is sufficiently reliable.

**Algorithm**

The application of the classical matching techniques in that methodology leads to very simple computations. Moreover, only measurements of the last stride are used. Thanks to this the developed algorithm (Fig 3.55) runs very fast.

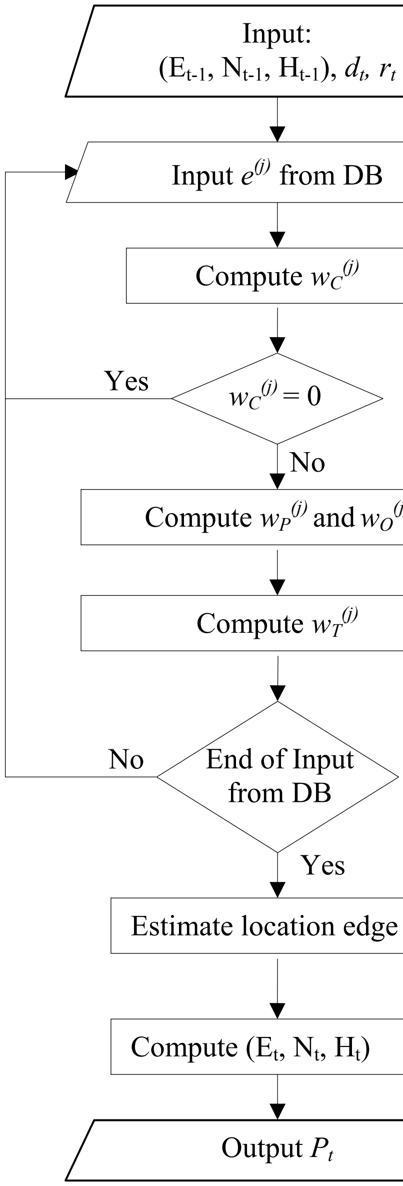
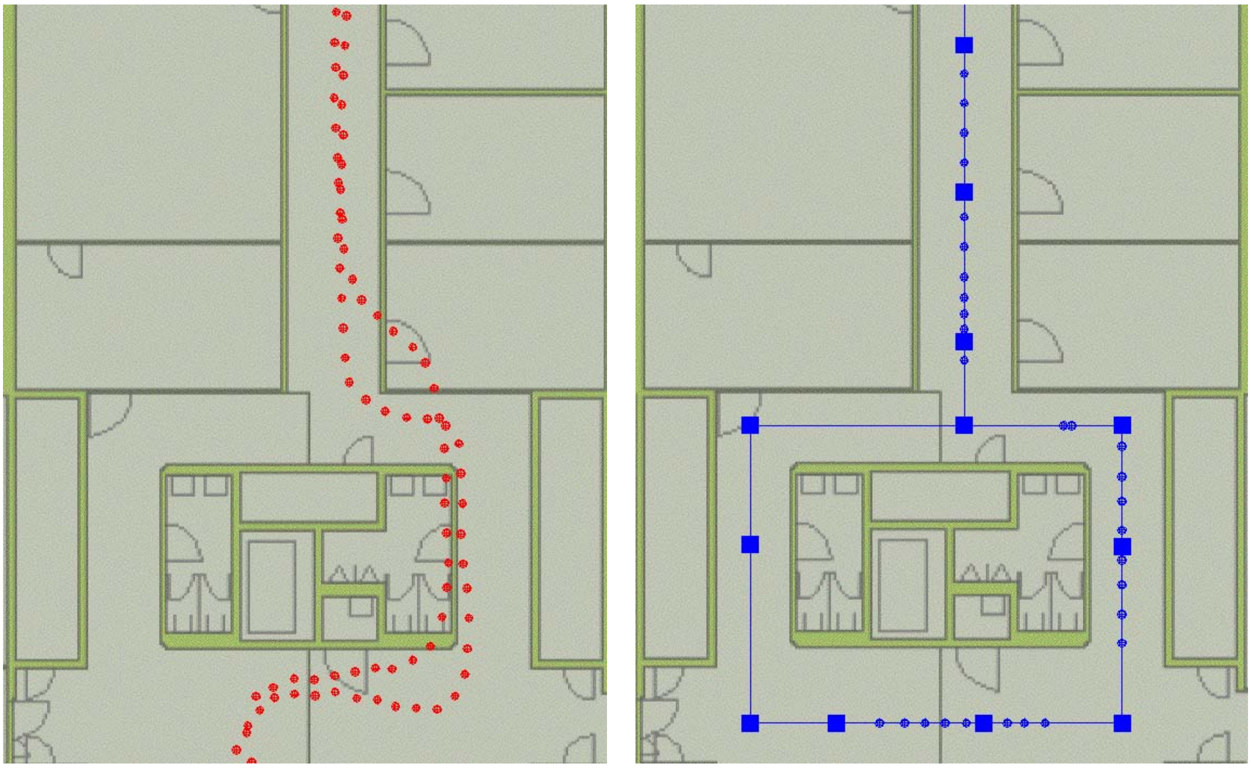



Figure 3.55: Estimation of $P_t$ applying the weighting system.

**Tests, results and analysis**

This algorithm for continuous localization is tested with many trajectories and the analysed aspects are the of the localization accuracy, robustness and continuity of the process.

The MM algorithm is performed as a post-treatment of the raw data. At each user's step a pair of consecutive points is analysed. The azimuth, the horizontal distance and the elevation between both points is calculated. The application of point-to-curve matching method showed excellent results. The performed trajectory includes all kinds of movement – passing through a corridor (Fig. 3.56), changing the floor using a staircase (Fig. 3.57) and elevator (Fig. 3.58)



a) b)

Figure 3.56: Test trajectory passing through a corridor, measured trajectory (a) and matched points (b)

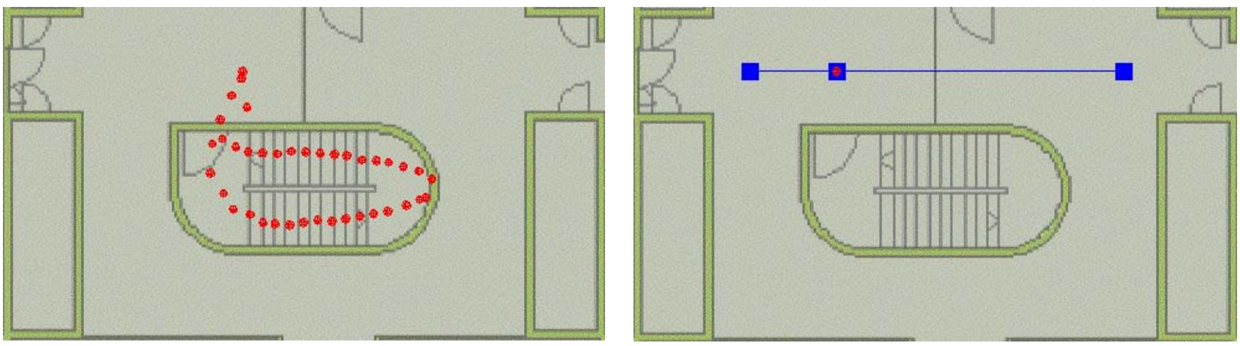


Figure 3.57: Test trajectory passing through a corridor and taking staircase

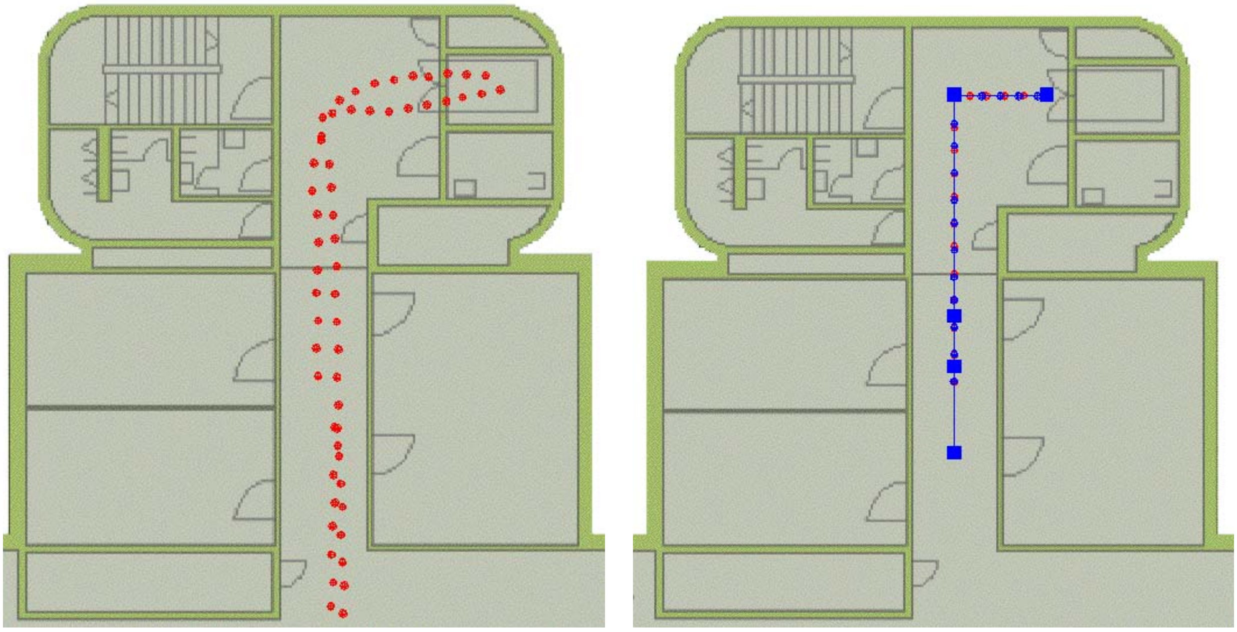


Figure 3.58: Test trajectory passing through a corridor and taking elevator

We can remark on Fig. 3.56 that the matched points are relatively dense on the edges, however on the turns around the junctions projections are missing. The reason is that on the turns the person passes in the internal part of the corridor. Thus the raw trajectory does not follow exactly the axis of the corridor and the matched points in the turns are not projected in the zones of the junctions. The continuance in such cases depends entirely on the performance of the measurement unit (PNM). That will present a problem for the localization in case of big errors in the measurements, e.g. when passing through a zone with magnetic disturbance.

On Fig. 3.57b the projected points are not very good visible, because they are associated to the vertical node. All these matched points correspond to the steps in the staircase. On Fig. 3.58 we have the same case with trajectory taking elevator. The vertical edges in both examples are designed next to the door of the staircase and the elevator.

We have evaluated the positioning accuracy of that method as percentage of the correct edge identification, which is 85%. As shown in the tests, the measured trajectory matches always with the spatial position of the corridor. There are no jumps of positions and the average distance to the middle of the corridor is acceptable.

### 3.3.3. Continuous localization based on Fréchet distance

The methods of continuous localization discussed so far estimate user's location through computation of specific weights considering separate elements of the trajectory. However, if we take a more global view on the user's trajectory we would be able to match a part of it in the contents of the graph in more robust way. Moreover, treating more elements of the trajectory in the same time, will assure additional control to the estimation of the other two methods. This method is based on the evaluation of the similarity between planar shapes. Here we will tackle this problem applying the *Fréchet distance* as a measure of the resemblance between two planar shapes.

**Theoretical formulation**

These days the solution of many technical problems (e.g. speech analysis and computer vision) relies on the shape recognition. Often two-dimensional shapes are defined as planar curves and the task is to determine how much these curves resemble each other. Logically the notion of resemblance needs to be reflected by an underlying metric, e.g. the Euclidean distance.

In the theory of spaces the Fréchet distance is defined as a measure of the similarity between two curves in a metric space. It takes into account the location and the order of the points along the curves [Alt et al. 1995].

Let $f : [a, b] \rightarrow V$ and $g : [a', b'] \rightarrow V$ be two curves, defined in the metric space $(V, d)$, where $a, b \in \Re$ and $a \leq b$. The Fréchet distance between $f$ and $g$ is determined as:

$$\delta_F(f,g) = \inf_{\alpha,\beta} \max_{t\in[0,1]} d(f(\alpha(t)), g(\beta(t))) \tag{3.33}$$

where $\alpha$ (respectively $\beta$) is an arbitrary continuous non-decreasing function from [0, 1] onto $[a, b]$ (respectively $[a', b']$). The most popular illustration of the Fréchet distance between two curves is the example with the man walking his dog (Annex B).

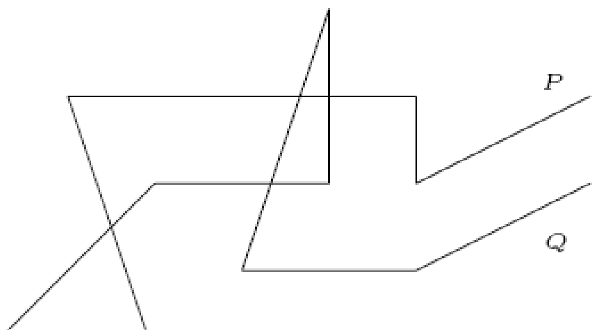



Figure 3.59 A pair of polygons *P* and *Q*

An intuitive variation of the problem is the computation of the Fréchet distance for a pair of polygons (Fig. 3.59). We consider the polygon $P$:[0, $N$] as a sequence of $N$ connected segments. The parameterization of that polygon consists in the definition of a parameter $a$ such that $P(a)$ refers to a given position on the curve, where $P(0)$ refers to the first vertex of the polygon curve and $P(N)$ refers to its last vertex [Pelletier 2002].

Suppose that $P$ and $Q$ are two polygons with lengths respectively $N$ and $M$. Using the continuous non-decreasing functions $\alpha$ and $\beta$ every point of the polygons $P$ and $Q$ can be modelized as a function of the time $t$ [Wenk et al. 2001]. If we define $\alpha(0) = 0$, $\alpha(1) = N$ and $\beta(0) = 0$, $\beta(1) = M$, the point of $P$ is given by $P(\alpha(t))$ and the point of $Q$ by $Q(\beta(t))$. Then for the Fréchet distance between two polygons we have:

$$\delta_F(P, Q) = \min_{\substack{\alpha[0,1]\to[0,N] \\ \beta[0,1]\to[0,M]}} \{\max_{t\in[0,1]} d(\,P(\,\alpha(t)\,), Q(\,\beta(t)\,)\,)\} \tag{3.34}$$

The computation of the Fréchet distance passes through the solution of the following decision problem [Venkatasubramanian 1999]:

Given polygonal curves $P$ and $Q$ and some real value $\varepsilon \geq 0$,
Decide whether $\delta_F(P,Q) \leq \varepsilon$. (3.35)

The geometric meaning of $\varepsilon$ is shown on Fig. 3.60. It corresponds to the distance between a pair of points, one from $P$ and another from $Q$. These distances are called *critical values* of $\varepsilon$.

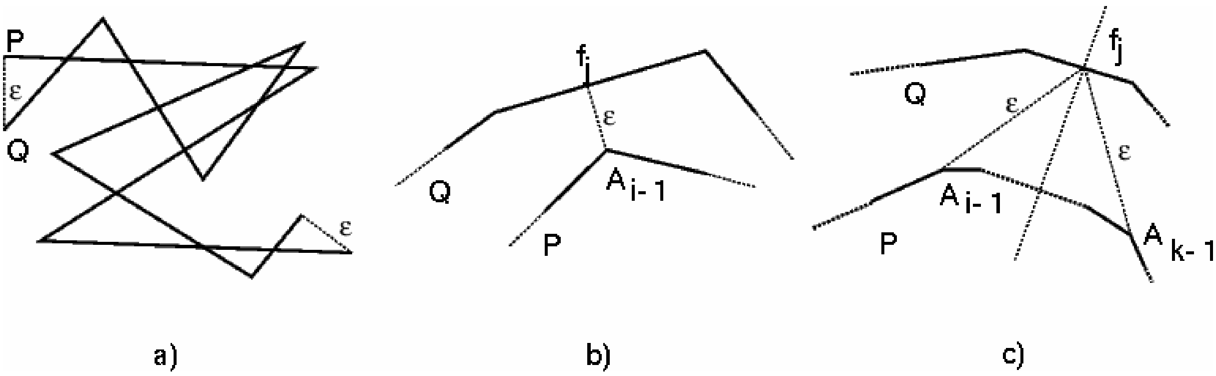



Fig 3.60. The geometric meaning of $\varepsilon$

Complete explanation of the solution of the decision problem is given by [Pelletier 2002] and [Buchin et al. 2006]. The main steps of the computation of the Fréchet distance are:

1. Determine all critical values of $\varepsilon$
2. Sort them
3. Do a binary search on the sorted sequence in each search step solving the decision problem.

After the computation the Fréchet distance takes the first value of $\varepsilon$ for which the decision is positive. This value shows how much both polygons are similar to each other.

One of the properties of the Fréchet distance, which is of great importance for our approach, is that $\delta_F(P,Q)$ can be computed for polygons with different parameterizations. That means the polygons can be composed of different number of segments. Another property is that the Fréchet distance is a measure for similarity of two polygons, and not for their proximity to each other. Thus the computation does not depend on the displacement nor on the orientation of the polygons.

Here we discuss how we apply the computation of the Fréchet distance in the problematic of continuous localization. We keep the idea that this method will serve mostly to control the localization solution of the other two methods for continuous localization. That is, computation will be made periodically taking into account relatively small parts of the trajectory. Thus the estimation of user's location will be made in reasonable intervals of time.

Let's take the example with the two polygons *P* and *Q* and change the notation of *polygon* to *polyline*. In our approach we define the polyline (*Q*) as part of the graph, composed of two consecutive edges. We can take more edges to construct a larger polyline, but that will extend the intervals of time between the estimations, mentioned above. The polyline (*P*) is defined as part of the user's trajectory with length close to the length of *Q*, where every vertex is referred to a step of the person. (Fig. 3.61). The construction of *P* and *Q* is discussed further.

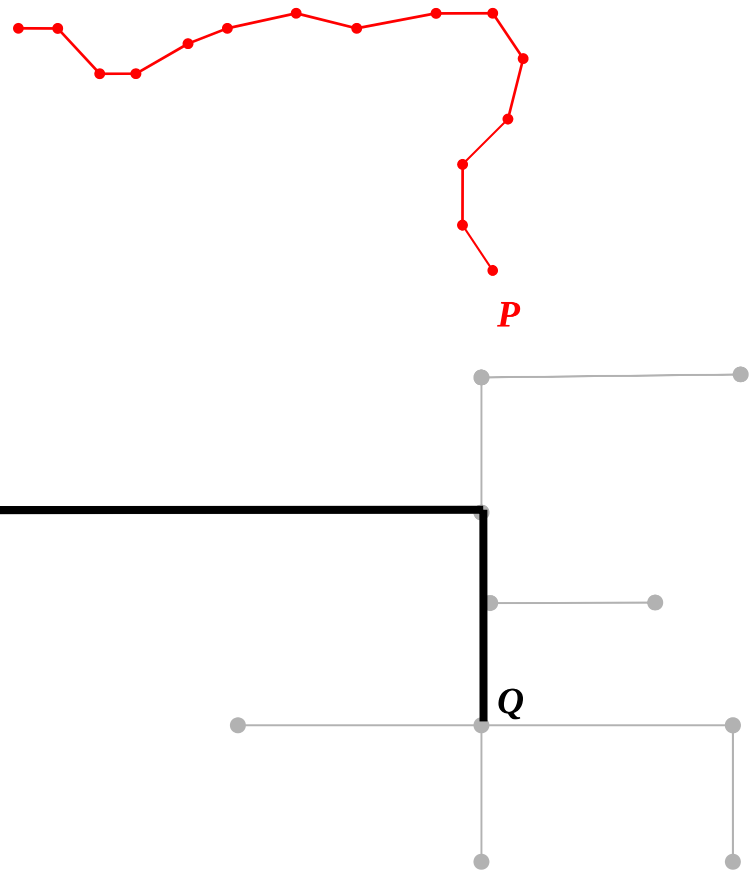



Figure 3.61 Part of trajectory and edges of the graph, forming the polylines *P* and *Q*

Suppose we know user's location and orientation at moment *t-k* and we need to determine the location at moment *t*, after the user has performed *k* steps (Fig. 3.62). These *k* steps form the first polyline for the comparison, named for simplicity *P*.

The definition of the second polyline (*Q*) takes into account the location edge $\hat{e}_{t-k}$ at moment *t-k*. This edge can be regarded as the first segment of *Q*. As second segment of *Q* we can take any one of the neighbour edges of $\hat{e}_{t-k}$, in the direction of walk, i.e. $e^{(j)}$, $j=1,\ldots,G$. For the example on Fig. 3.62 we define three polylines: $Q_1(\hat{e}_{t-k}, e^{(1)})$, $Q_2(\hat{e}_{t-k}, e^{(2)})$, $Q_3(\hat{e}_{t-k}, e^{(3)})$.

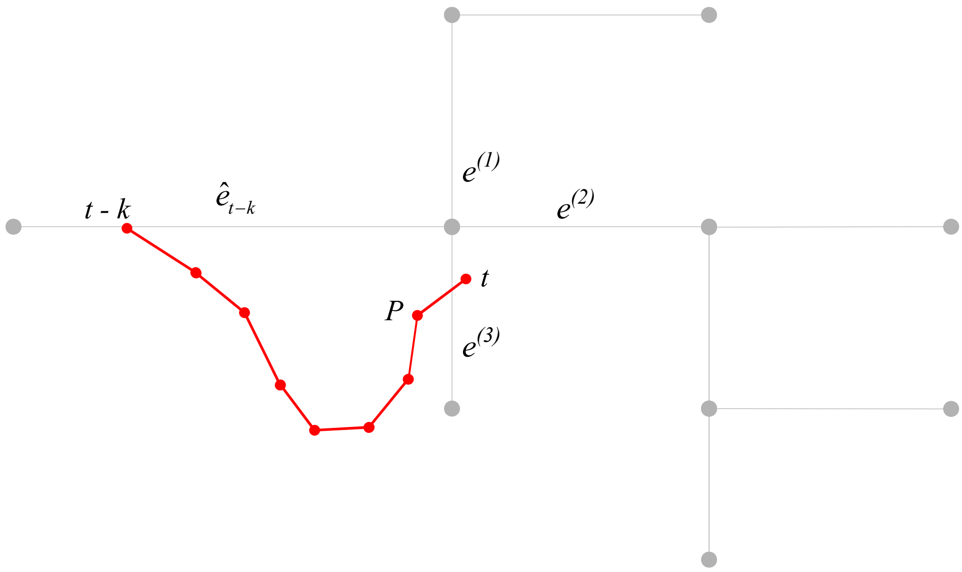



Figure 3.62 Trajectory from moment $t$ to $t$-$k$

The aim is to determine the location edge among the neighbour edges $e^{(j)}$, and to estimate user's location on it at moment $t$. This is possible if we compute the Fréchet distance $\delta_F(P,Q_j)$ for each pair of polylines $(P, Q_j)$, $j=1,\ldots,G$. The polyline $Q_j$ which gives smallest $\delta_F(P,Q_j)$ is considered as the best match of $P$ and $e^{(j)}$ is the location edge.

In order to estimate the location point at moment $t$ on the location edge we consider the length of $P$, and fix a point on the location edge considering that distance. The fixed location point defines the first vertex of the new polyline $P$ for the next computation. Knowing the direction of walk (coming from the orientation of the person) we can determine the next set of neighbour edges. Here we refer to the common assumption that the person performs a normal walk.

The trajectory evaluates permanently in time. An essential question is what part of the trajectory to take for the computation, i.e. at which moment to make the estimation. For decide that we take into account the distance of each candidate polyline $Q_j$. From moment $t$-$k$ we start to accumulate the length of the strides, thus computing the length of $P$ at each step. The moment of estimation is set when the length of $P$ overcomes the distance of the shortest polyline $Q_j$. This choice assures that the initial vertex for the next estimation will be fixed on one of the neighbour edges.

**Algorithm**

The following algorithm uses information from the PNM and from the map database. The polylines are defined in files containing the coordinates of their vertexes in the coordinate system of the graph. The coordinates of the vertexes of $P$ (the user's steps) are computed using the distance and direction measurements ($d_t$, $r_t$) starting from the initial position noted with ($E_0$, $N_0$) (Fig. 3.63).

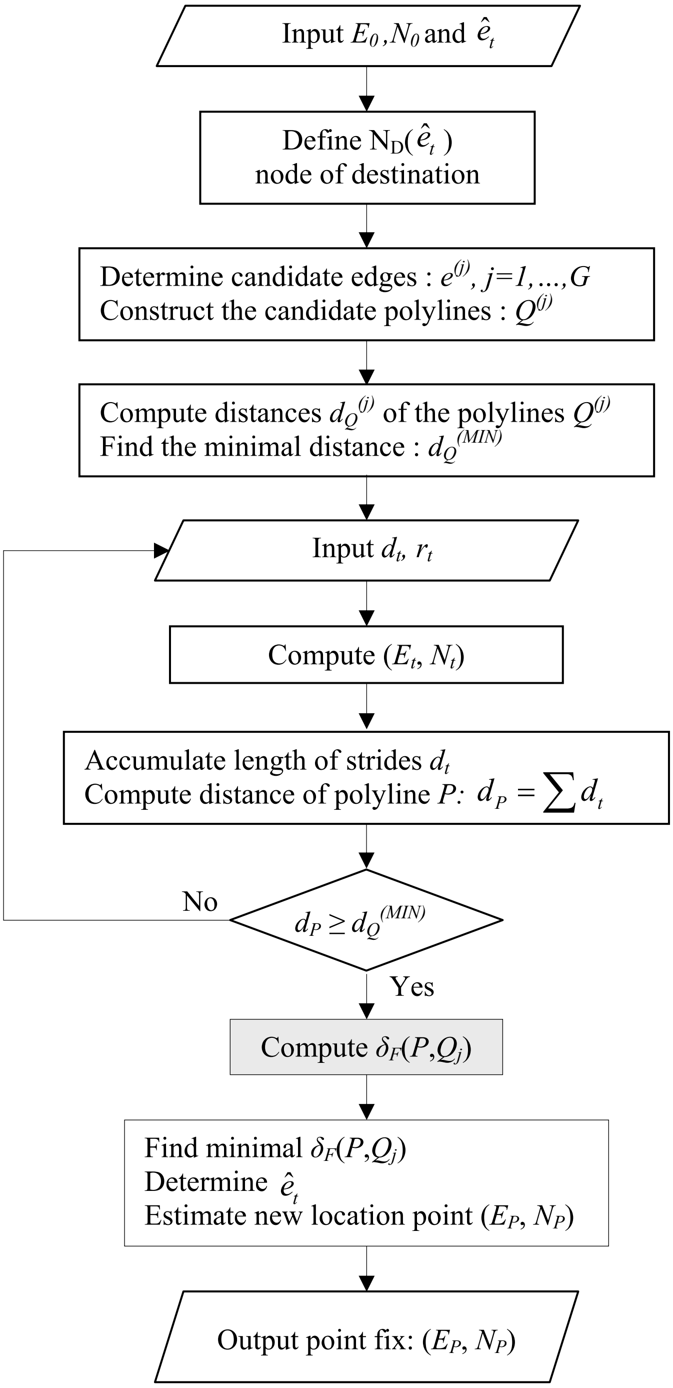



Figure 3.63 Estimation of the location after computing the Fréchet distance

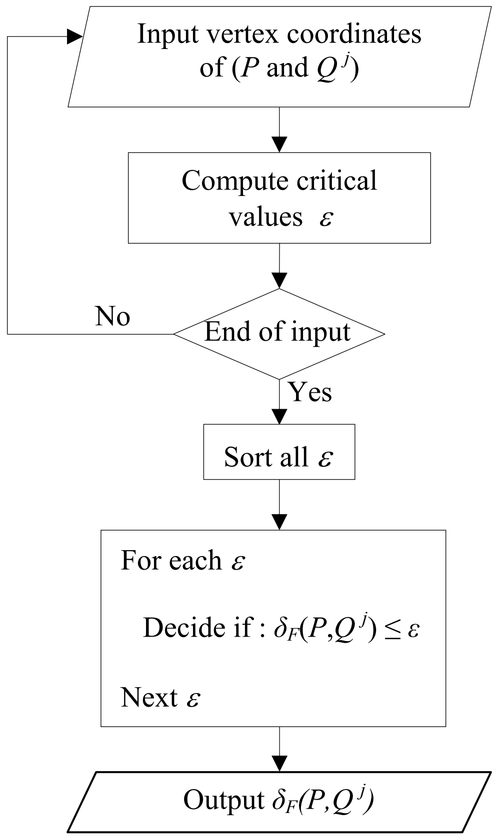



Figure 3.64 Computation of the Fréchet distance for pair of polylines

The flowchart on Fig. 3.64 describes the typical algorithm of computation of the Fréchet distance. Such algorithm is elaborated in the Laboratory of Geomatics in 2006 using MATLAB [Constantin and Wasser 2006].

**Tests, results and analysis**

The performance of the algorithm is tested on the campus of EPFL. We observe and analyse several aspects including presence of bias in the initial direction of trajectory and scale factor in the distance measurements. First we start with a normal trajectory (Fig. 3.65). Estimation is made once per several steps as discussed above.

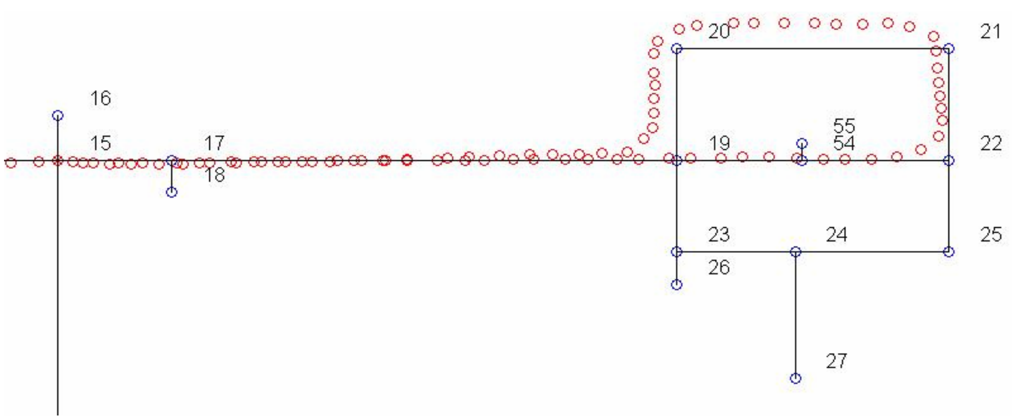
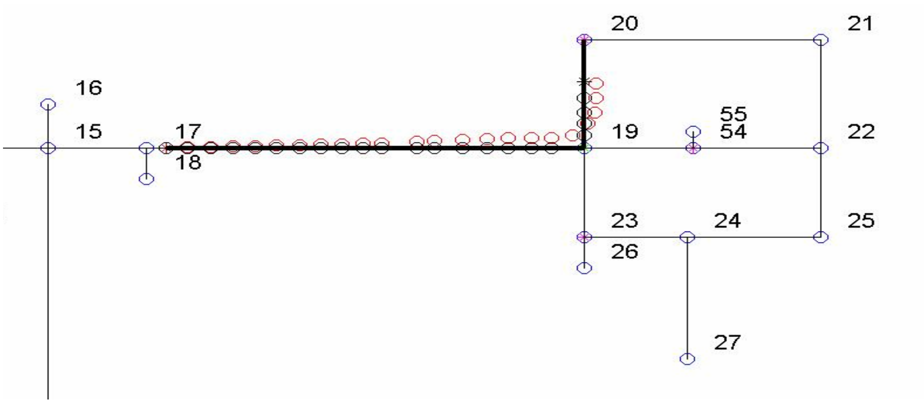
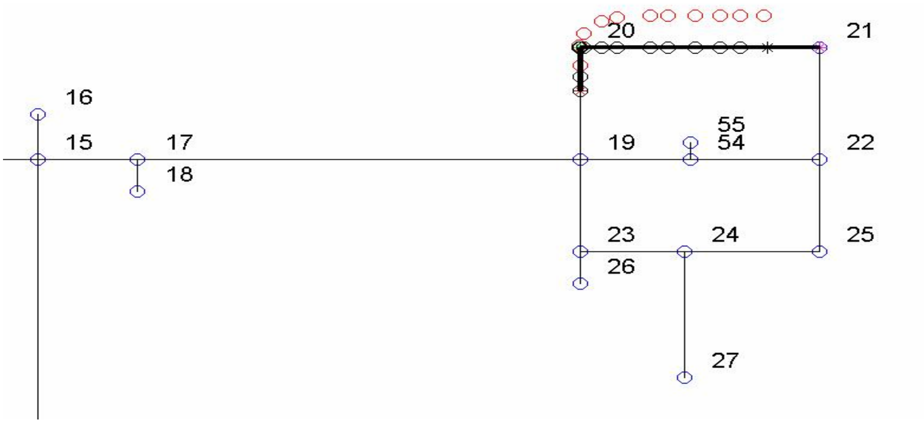


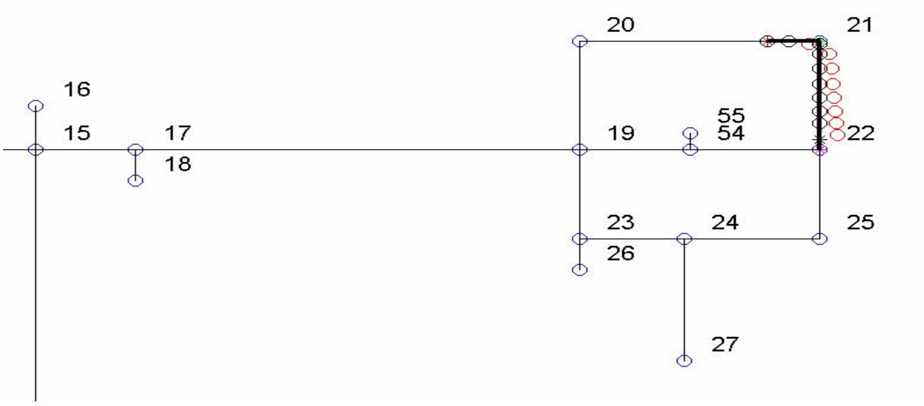

Figure 3.65 Repetitive location estimation of normal trajectory



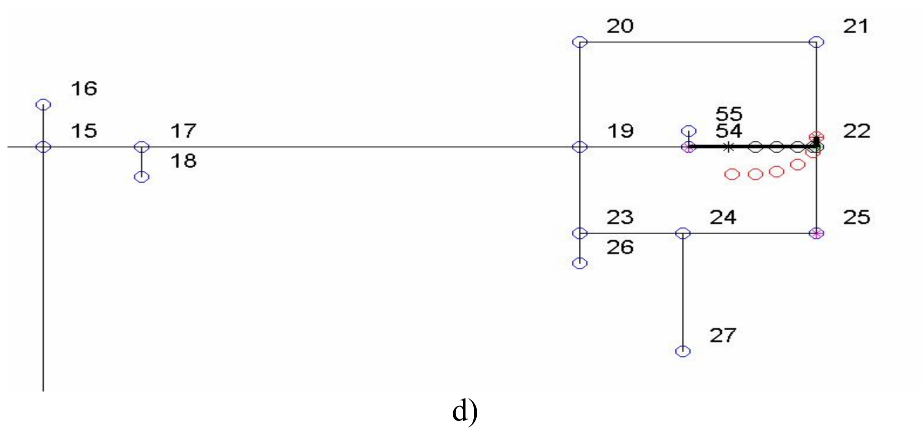

a)




b)




c)

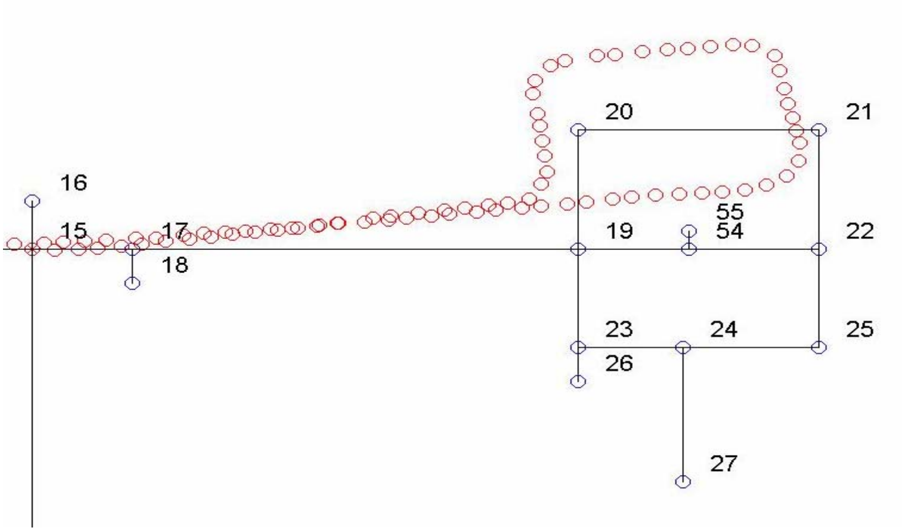



d)

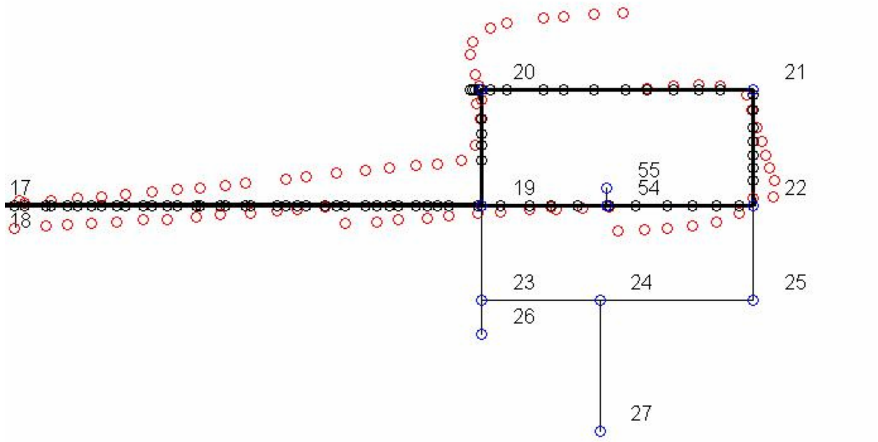
Figure 3.66. Repetitive location estimation. Point fix is marked with ∗ on the location edge

However, for more clearness the intermediate steps are projected on the corresponding edges. The measured positions are shown in red and the estimations in black (Fig. 3.66). In every section (a, b, c, d) a new repetition of the algorithm from Fig. 3.63 is performed, estimating the location of the last step on the graph.

For that trajectory the algorithm shows excellent results. We can remark that the trajectory follows relatively well the elements of the graph. This is very rare indoors, considering the inertial sensors, which are very sensitive from different factors. In the next tests we use trajectories that contain errors in the inertial measurements.




a)

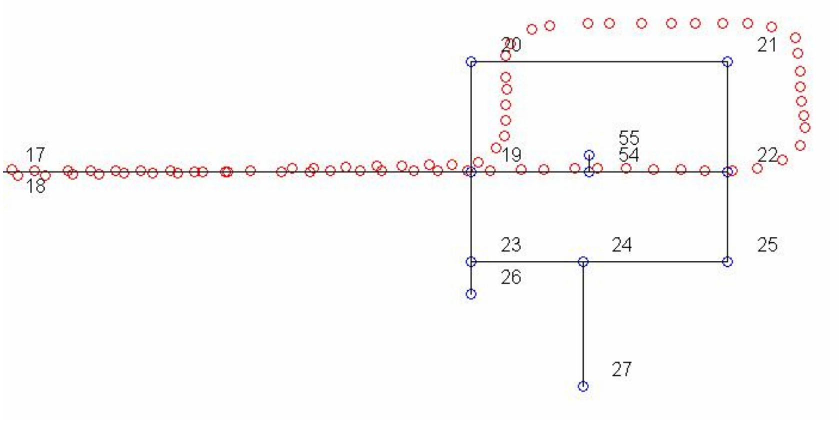



b)

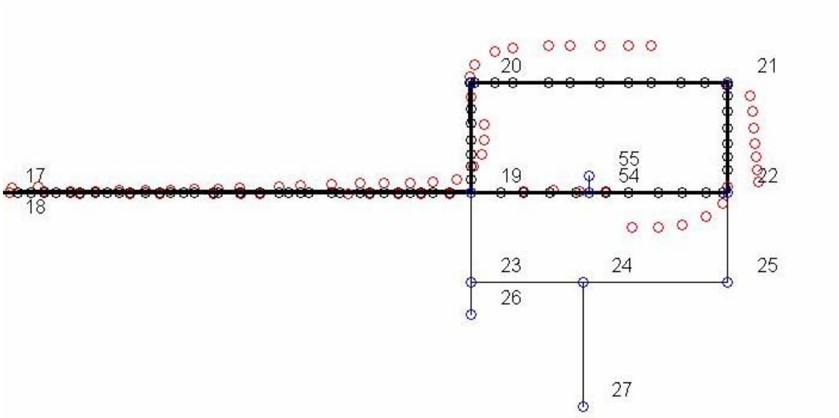
Figure 3.67 Repetitive location estimation of biased trajectory

On Fig. 3.67a we have a trajectory with error of 7 degrees in the initial orientation. The repetitive estimations are shown on Fig. 3.67b. One can distinguish clearly the repetitions of the algorithm. At each repetition the new estimated point fix is taken as initial location for the next part of the trajectory. Therefore on Fig. 3.67b the subsequent parts of the trajectory are transformed and each part starts from the corresponding edge. The algorithm shows robustness and performs reliable location estimation.

For the next test we consider a trajectory stretched by a scale factor. That corresponds to the existence of systematic error in the measurements. We consider an error of 10% for the length of the trajectory (polyline *P*). The whole raw trajectory is shown on Fig. 3.68a and the repetitive estimations – on Fig. 3.68b.




a)




b)

Figure 3.68 Repetitive location estimation of biased trajectory

Even in the presence of a scale factor in the trajectory the performance of the algorithm is excellent. That shows that the comparison of both polylines takes into account their geometry and not their lengths.

The results of these tests show that the application of the Fréchet distance is excellent solution for the problematic of map-matching. It can be applied for longer polylines, allowing for deeper analysis of the user's displacement.

The repetitive estimation of the new location refers to the computation of translation parameters for the next steps of the trajectory. Using these parameters corrections to the inertial sensors can be computed.

### 3.3.4. Conclusions

In this study we discuss three different methods for continuous localization. The output of these methods is user's location represented by a point fix on the graph. The methods are independent from each other and can be applied as continuance of the initial localization separately.

The first method (chapter 3.3.1.) shows another application of the Bayesian inference different from the method of initial localization (chapter 3.2.). The computation is based on the distance and heading of every stride instead of the polyline parameters, which is in accordance with the time discretization of the process.

The second method (chapter 3.3.2.), based on classical matching techniques, has already proven its efficacy in the vehicle navigation (e.g. rail transport). However, the assumption that the vehicle is on the road, can not be made for the person. Thus the method shows sensibility to the liberty of movement of the user, which is illustrated in the example with the passage on the turns (Fig. 3.56).

Comparing the first two methods, we can say that in terms of localization accuracy they are very similar. However, the first method is preferable because it shows better performance on the junctions of the graph. Another advantage is that in the case of more sophisticated graphs it will be more robust than the second one.

The third method (chapter 3.3.3.) refers to a new application of the Fréchet distance. While the first two methods can provide location estimation in real time, this method treats parts of the trajectory in post treatment. It can be regarded as a complement to the other methods assuring control of the estimation. The post treatment mode allows treating even the whole trajectory. For certain applications that method can be applied without having reference to the real time location estimation. An example can be given with the analysis of the user's trajectory in a supermarket.

| Localization methods | Estimation | Edge control |
|---|---|---|
| Initial localization + Bayesian inference | Real time | Yes |
| Initial localization + Classical techniques | Real time | Yes |
| Initial localization + Fréchet distance | Post treatment | Yes |
| Bayesian inference + Fréchet distance | Real time | Yes |
| Classical techniques + Fréchet distance | Real time | Yes |

Table 3.3: Variants of localization methods

If we need to construct a methodology for the entire navigational process based on the methods discussed here, we have to consider the following variants (Table 3.3):

The localization accuracy depends mainly on the precision of the graph. The classical matching techniques applied in the second method (chapter 3.3.2.) are sensitive to gross errors in the measurements. Furthermore, for sophisticated graphs the application of the weighting system can give very inaccurate results.

In the other two methods (chapters 3.3.1. and 3.3.3.), the accuracy depends directly on the accumulated distance. Taking into account the technical characteristics of the PNM we saw that the method, based on the Bayesian inference is precise enough (Table 3.2). The method, based on the Fréchet distance shows robust performance, even in the case with the trajectory stretched by a scale factor. Both methods assure an accuracy in the range of 1 meter, which is sufficient in the domain of pedestrian localization. Table 3.4 gives for comparison some pedestrian localization methods developed recently.

| Authors | Year | Equipment | Localization Accuracy [m] | Area |
|---|---|---|---|---|
| Mezentsev O. (*Canada*) | 2005 | HSGPS<br>DR | 42 | Outdoors |
| Usui et al. (*Japan*) | 2005 | GPS<br>Imagery | 10 | Outdoors |
| Oh et al. (*USA*) | 2004 | GPS<br>Map information | 2 – 4 | Outdoors |
| Kourogi et al. (*Japan*) | 2006 | GPS<br>DR<br>RFID | 2 – 6 | Indoors |

Table 3.4: Comparison with some pedestrian localization methods developed recently

A major factor in these methods is the usage of measurements for absolute positioning (GPS). Moreover in the case of indoor navigation additional equipment is used which overloads the user. The main advantages in the methods developed in this study are the usage of IMU and map information which assures the autonomy of the process.

## 3.4. Pedestrian guidance

The first fundamental task in the context of the pedestrian navigation is to localize the person discussed in the previous chapters. The second task is to guide the person to his/her destination, i.e. pedestrian guidance. It consists in sending instructions from the navigation system to the user. The guidance can be performed only if the location of the person is known.

That process takes into account a predefined optimal path (e.g. the shortest path) between the user and the destination, called for simplicity *itinerary*. The algorithms for computation of the path have been discussed in 2.1.5. In this chapter we discuss the different types of guidance and the particular problem of path finding.

### 3.4.1. Types of guidance

Depending on the liberty of movement of the person we distinguish three types of guidance [Büchel 2004]:

- *Free guidance*: That type of guidance does not refer to any itinerary. The person is localized on the graph. The navigation system does not impose any restrictions and does not transmit guiding instructions to the person. However, information like the accessibility to certain zones can be sent to the user.
- *Semi-constrained guidance*: This type of guidance refers to an itinerary. According to that itinerary instructions are sent to the user. However, (s)he is not forced to follow it and can leave the itinerary in any moment. Then, considering that deviation another optimal path is computed from user's new position to the destination and instructions are sent according to the new itinerary.
- *Constrained guidance*: This type of guidance refers to an itinerary as well. Contrary to the previous type the user is forced to follow the itinerary. In case of deviation, instructions are sent to the user in order to redirect him/her to the itinerary.

The three types of guidance have different applications. In the first type the notation of guidance can be argued, since no itinerary is taken into account. However, information on the objects of interest in the vicinity of the person can be useful to define a destination in certain moment. The third type of guidance can be applied in some extreme cases when the person must be directed rapidly to the destination. As example we can give the evacuation of the person from the building.

The second type has the most practical application. It is discussed in details further, considering the case of path finding.

### 3.4.2. Path finding

The problem of path finding concerns the case when the person has lost his itinerary. The aim is to instruct the person in order to put him in the right direction to the destination. There are several phases of the problem: to detect the deviation of the person; to re-compute the itinerary; to give instructions to the user [Rey 2006]. Thus the continuity of the navigation process is assured.

**Theoretical formulation**

The discussion of the problematic is based on the following example (Fig. 3.69). The itinerary from point $A$ to point $B$ is computed. Following the instructions of the navigation system the person has arrived at point *P*. Assume that in the next moment the person takes a wrong direction, which causes a deviation from the itinerary. The first phase of the problem is to localize that deviation.

The solution is based on the association of the trajectory to the graph discussed in chapter 3.3.1. For detect a deviation we estimate user's location on every step and verify whether that location is on an edge from the itinerary. For the position fixes from *A* to *P* that verification is positive. However, point fix *C* is not on the itinerary. That deviation is detected and localized. For instance, in the case of constrained guidance the system would send a warning message to the user that (s)he has left the itinerary and would give instruction to step back on the itinerary. In the case of semi-constrained guidance the user is not forced to go back on the itinerary. (S)he can continue in the new direction ignoring the warning message from the system.

The second phase of the path finding is the computation of a new itinerary. The question here is when to re-compute the itinerary. In the first several steps after the deviation it is almost sure that the optimal path is steel the original one.

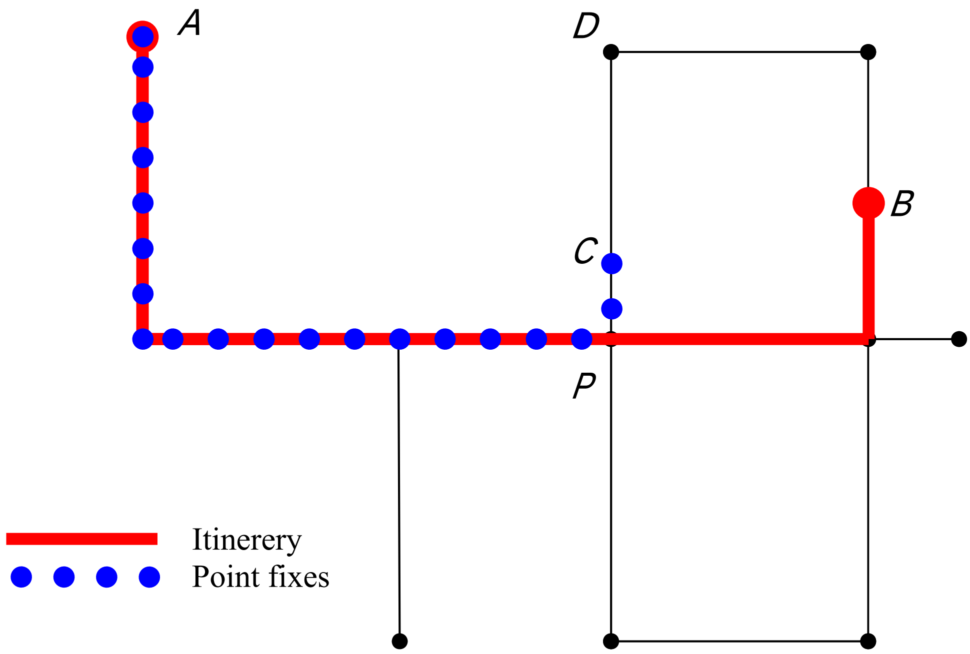



Figure 3.69: Illustration of path finding problematic

We can start computing the optimal path to the destination on each user's step, taking the step as origin of the itinerary. However, this will represent computational burden for the process. Instead we can compute the optimal path once the person has arrived at the junction (point *D* on Fig. 3.69). Thus the node of the junction will be taken as origin of the itinerary. After the computation of the new itinerary instructions are given to the user.

**Algorithm**

The developed algorithm (Fig. 3.70) applies as well the continuous localization discussed above. All points are defined by their coordinates (E, N, H).

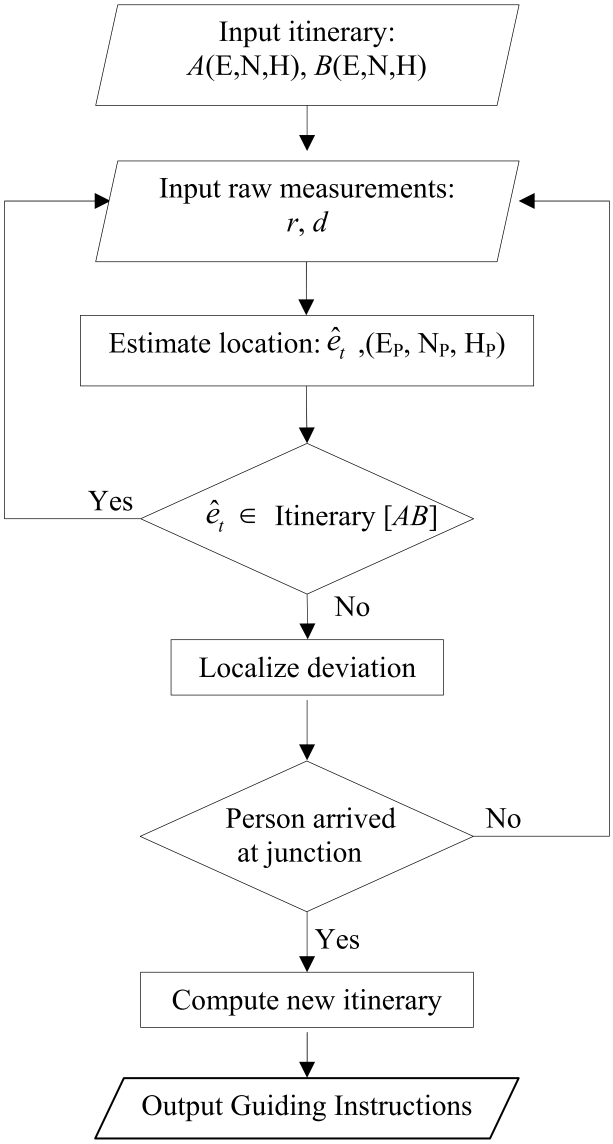



Figure 3.70: Flowchart of the guidance algorithm

**Tests, results and analysis**

The algorithm is tested with several scenarios. For the example on Fig. 3.71 we consider a person whose attention has been captured by, for example, poster on the wall. So, (s)he has deviated from the itinerary. It is evident, that the new itinerary is shorter then the original one. Thus the person can continue to the destination. For the tests we use the computation of the shortest path developed or the navigation on the campus of EPFL (http://plan.epfl.ch).

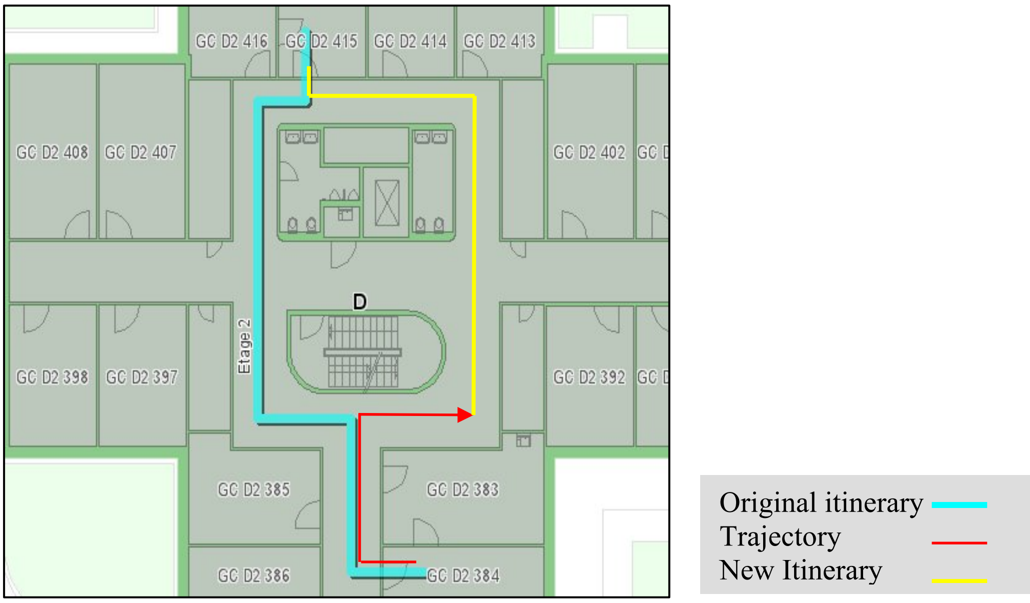



Figure 3.71: Deviation from the itinerary. The new itinerary is shorter.

For the next example we take the same scenario. This time the new itinerary is longer then the original one and the user is instructed to take the original itinerary (Fig. 3.72).

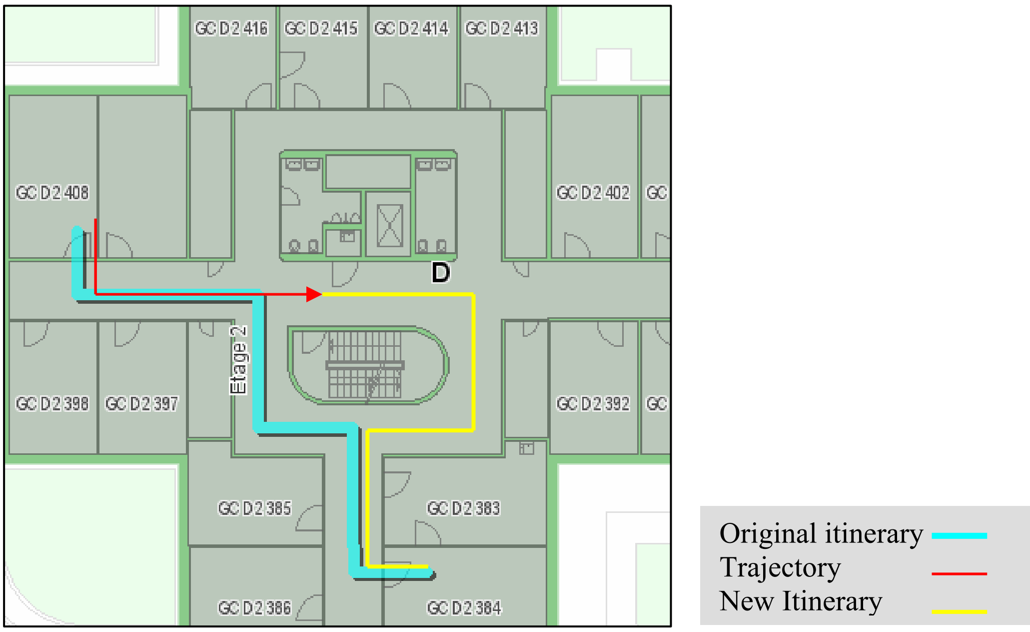



Figure 3.72: Deviation from the itinerary. The new itinerary is longer.

### 3.4.3. Conclusions

The main methods that take place in this approach are the computation of the itinerary and the continuous localization. The results of the guidance algorithm are dependent on the performance of these methods.

For the first method we refer to the well known Dijkstra's algorithm for optimal path computation. It uses only data from the graph and shows excellent performance, computing itinerary.

We consider the second method, the continuous localization, as more pertinent to the performance of the algorithm. It is very important to have reliable location estimation, because during the walk, in every computation of new itinerary the point of origin depends on the user's location. The detection of deviation from the itinerary as well the definition of the new origin of the itinerary relies on the performance of this localization algorithm on the junctions. As already discussed in chapter 3.3.1, the localization algorithm shows robustness on the junctions. Even though, it is possible that some of the user's steps are matched to a wrong edge at the junction (Fig. 3.48). These point fixes may be the reason for false deviation detection. That problem may be tackled with the introduction of some tolerance, in order to send the warning message after the number of point fixes out of itinerary becomes bigger. Besides that remark the guidance algorithm using inertial measurements only shows good results. We keep to our assumption that the person performs a normal walk

# Bibliography

Abwerzger, G., Hofmann-Wellenhof, B., Ott, B., Wasle, E. (2004): GPS/SBAS and Additional Sensor Integration for Pedestrian Applications in Difficult Environments, Presented at the ION GNSS 2005, September 21-24, Long Beach, California.

Alt, H., Godau, M. (1995), Computing the Frechet distance between two polygonal curves, Internat. J. Comput. Geom. Appl., 1995.

Alt, H., Knauer, C., Wenk, C. (2001), Matching Polygonal Curves with Respect to the Frechet Distance, 2001

Althaus, Philipp, Verschure, Paul F.M.J. (1999), Distributed Adaptive Control 5: Bayesian Theory of Decision Making, Implemented on Simulated and Real Robots, Institute of Neuroinformatics, University/ETH Zurich

Buchin, Kevin, Buchin, Maike, Wenk, Carola (2006), Computing the Frechet Distance Between Simple Polygons, EWCG 2006, Delphi, March 27–29, 2006

Constantin, V. and Wasser, F. (2006) Développement d'algorithmes basés sur la distance de Fréchet appliqués à la mobilité des personnes, Projet de semestre, ENAC – TOPO, automne 2006

Doucet, A., de Freitas, N. and Gordon, N. (2001) Sequential Monte Carlo Methods in Practice, New York: Springler-Verlag.

Lachapelle, G., Mezentsev, O., Collin, J. and MacGougan, G. (2003), Pedestrian and Vehicular Navigation Under Signal Masking Using Integrated HSGPS and Self Contained Sensor Technologies, World Congress, International Association of Institutes of Navigation, Berlin, 21-24 October 2003

Ladetto, Q. (2003) Capteurs et Algorithmes pour la Localisation Autonome en Mode Pédestre, Phd Thesis, École Polytechnique Fédérale de Lausanne, 2003.

Ladetto, Q., van Seeters, J., Sokolowski, S., Sagan, Z., Merminod, B. (2002), Digital Magnetic Compass and Gyroscope for Dismounted Soldier Position & Navigation, NATO Research and Technology Agency, Sensors & Electronics Technology Panel

Merminod, B. (2003) Topométrie générale, Laboratoire de topométrie, EPFL, Lausanne, Switzerland

Pelletier, S. (2002), Computing the Fréchet distance between two polygonal curves, Computational Geometry 308-507A, McGill University, Fall 2002

Pyo, Jong-Sun, Shin, Dong-Ho, Sung, Tae-Kyung (2001), Development of a map matching method using the multiple hypothesis technique, 2001 IEEE Intelligent Transportation Systems Conference Proceedings - Oakland (CA), USA - August 25-29, 2001

Quddus, M., Noland, R, Ochieng, W. (2006), A High Accuracy Fuzzy Logic Based Map Matching Algorithm for Road Transport, Journal of Intelligent Transportation Systems, 10(3):103–115, 2006

Quddus, M., Ochieng, W., Zhao, L., Noland, R. (2003) A general map matching algorithm for transport telematics applications, Centre for Transport Studies Dept. of Civil and Environmental Engineering Imperial College London

Rey, L. (2006), Navigation pédestre à l'EPFL, projet ENAC, SSIE – TOPO, 2006

USUI, Sumio, TSUJI, Junichiro, WAKIMOTO, Koji, TANAKA, Satoshi, KANDA, Junshiro, SATO, Fumiaki and MIZUNO, Tadanori (2005), Evaluation of Positioning Accuracy for the Pedestrian Navigation System, IEICE Transactions on Communications 2005 E88-B(7):2848-2855; doi:10.1093/ietcom/e88-b.7.2848

Venkatasubramanian, S. (1999), Geometric Shape Matching and Drug Design, PhD thesis, Department of Computer Science, Stanford University, August 1999

# Chapter 4

## 4.1. Conclusions

The functionality of any navigation system can be separated in two levels: hardware level and software level. The first level considers the performance of the sensors of the system. The second level deals with the data treatment. The motivation to create a methodology for autonomous personal positioning and navigation indoors led us to the development of several algorithms. This solution has been chosen for two reasons.

First, an algorithm represents a development on software level. It is preferable instead of combining more performing and more expensive inertial sensors (development on hardware level). Moreover, that eliminates the physical overcharge of the person, since the algorithms and the map database are stored on the user's PDA. Thus the autonomy of the process is guaranteed.

Second, the implication of map database information in the positioning process makes the positioning solution more robust and reliable. Since we use inertial measurements only, the information of the map database is indispensable for the process.

In fact, the algorithms provide an absolute position without using "absolute" measurements (e.g. GPS). Nevertheless, the algorithms are flexible enough and allow the implication of external measurements in the computation, where such measurements are available.

The main assumptions in this approach are that the person performs a normal walk and the trajectory is performed in places covered by the map database. At this stage and with this reduced set of information such assumptions are reasonable.

The map database represents a crucial source of information for the process of localization. So far it has been used for the task of continuous localization. Here we show that it can be used for initial localization as well.

## 4.2. Perspectives

**Real time localization**

If we take a more global view of the evolution of the personal positioning technology we can classify the Personal Navigation Assistants (PNA) in several generations [Bernstein 1996]. The first generation simply provides the user with a map and the ability to search the map (e.g. search for an address or a landmark). Second generation PNAs provide both a map and the user's rough location (a few hundreds of meters). Third generation PNAs provide a map, the user's location, and information on the POI in the user's vicinity.

Nowadays this evolution continues with the introduction of the inertial sensors for personal positioning. The combination of INS with the map database and the development of positioning algorithms (presented this research) can be considered as the next step of the personal positioning. Of course it is not the final step.

Even if the discussed algorithms are dedicated to work in real time mode, they have been written and tested in post treatment mode. Their implementation for real time performance will be the logical continuation of the development in this domain.

The algorithms can be implemented on a suitable software language (e.g. Java), and installed on the user's portable device. This development concerns the INS as well. The reason is that the output of the algorithms consists in corrections to the user's position that can be used to compute calibration parameters for the inertial sensors as well. That is another aspect of the real time problematic which must be treated in parallel with the implementation of the algorithms.

**Detecting more sophisticated movements**

In the frame of some real life situations like e.g. firefighter guidance, the assumption for normal walk can not be held. That imposes the need for modeling more sophisticated movements of the person, like jumping, going backwards, etc. This perspective is closely connected with the future development of advanced sensors. Moreover, it can be regarded as an important step of the real time implementation of the pre-processing algorithms.

**DB uncompleted**

The constructed environment is in a constant development. We are witnesses of the creation of new buildings or modification of existing structures. These modifications must be reflected in the map database representing the constructed environment. That requires the maintenance of the map database in order to ensure a regular update.

However we can assume that sometimes the map database is not updated with the modifications of the structure. A typical case that is often presented is the appearance of a new door or the elimination of an existing door.

The other case is the way to represent the big areas like for example the foyers at the entrance of some buildings (Fig. 4.1). In that case the solution can be found either in the modelization of the map database or in the positioning methodology.

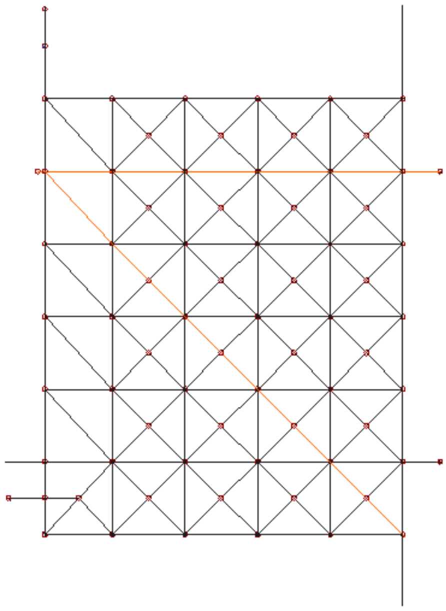


Fig. 4.1 A big square and its graph representation (EPFL campus)

**Automatic creation of map DB from plans**

The creation of the map database of the building is a long, expensive and complicated task. It passes through three main phases. The first phase is digitizing, whereby the AutoCAD plans are transformed into a graph. The second phase consists in the definition of attributes for the elements of the graph. The third phase is the creation of the database tables.

While the third phase is fairly automated with the application of specific file emulators, the first two phases are very time consuming. As example we can give the graph representation of the EPFL, consisting of near 32 000 edges, constructed in six months.

That process can be partially automatized. We can use the digital floor plans of the building as input to generate a Voronoi diagram [Evennou 2007]. Then to extract only part of the diagram and to construct the graph representation of the floor. The Voronoi diagram is often applied in the domain of mobile robots for path computations.

**Other definition of the map DB**

The graph representation of the building can be considered as the first approach to map database definition for indoor navigation. After applying the map matching process the user's position is estimated as a point of the graph. This does not correspond exactly (order of decimeters) to the real position of the person, because the graph elements pass through the middle of the corridors. That deterministic solution gives birth to new reflections on the database modelization. If we use the 2D floor plans directly we can define the elements of the building as zones (corridor, room, etc.).

Thus we could approach the position estimation by a continuous distribution. That solution can be very robust, but yet it would be expensive in terms of computation.

# Bibliography

Bernstein, David and Kornhauser, Alain (1996), An Introduction to Map Matching for Personal Navigation Assistants, New Jersey TIDE Center, New Jersey Institute of Technology, Newark, NJ

Evennou, Frédéric (2007), Techniques et technologies de localisation avancées pour terminaux mobiles dans les environnements indoor, PhD thesis, UNIVERSITE JOSEPH FOURIER - GRENOBLE I

## Annex A - NMEA messages supported by PNM: *RMC* and *GGA*.

$GPRMC, <*utc*>,A, <*lat*>,N, <*lon*>,E, <*speed*>,<*azi*>,<*date*>,,<*chk*>

| | |
|---|---|
| <*utc*> | UTC time in hhmmss.sss format (Hours, Minutes, Seconds), decimal representation |
| <*lat*> | Latitude in dd°mm.mmm (Degrees, Minutes), with North-South indicator, decimal representation |
| <*lon*> | Longitude in dd°mm.mmm (Degrees, Minutes), with East-West indicator, decimal representation |
| <*speed*> | Speed in knots, decimal floating point representation |
| <*azi*> | Azimuth in ddd°.m (Degrees, Minutes) |
| <*date*> | Date in ddmmyy (Day, Month, Year), decimal representation |
| <*chk*> | Standard NMEA checksum |

Example:

$GPRMC,163116.397,A,4647.5289,N,00709.6331,E,2.9,284.7,060106,,*0A

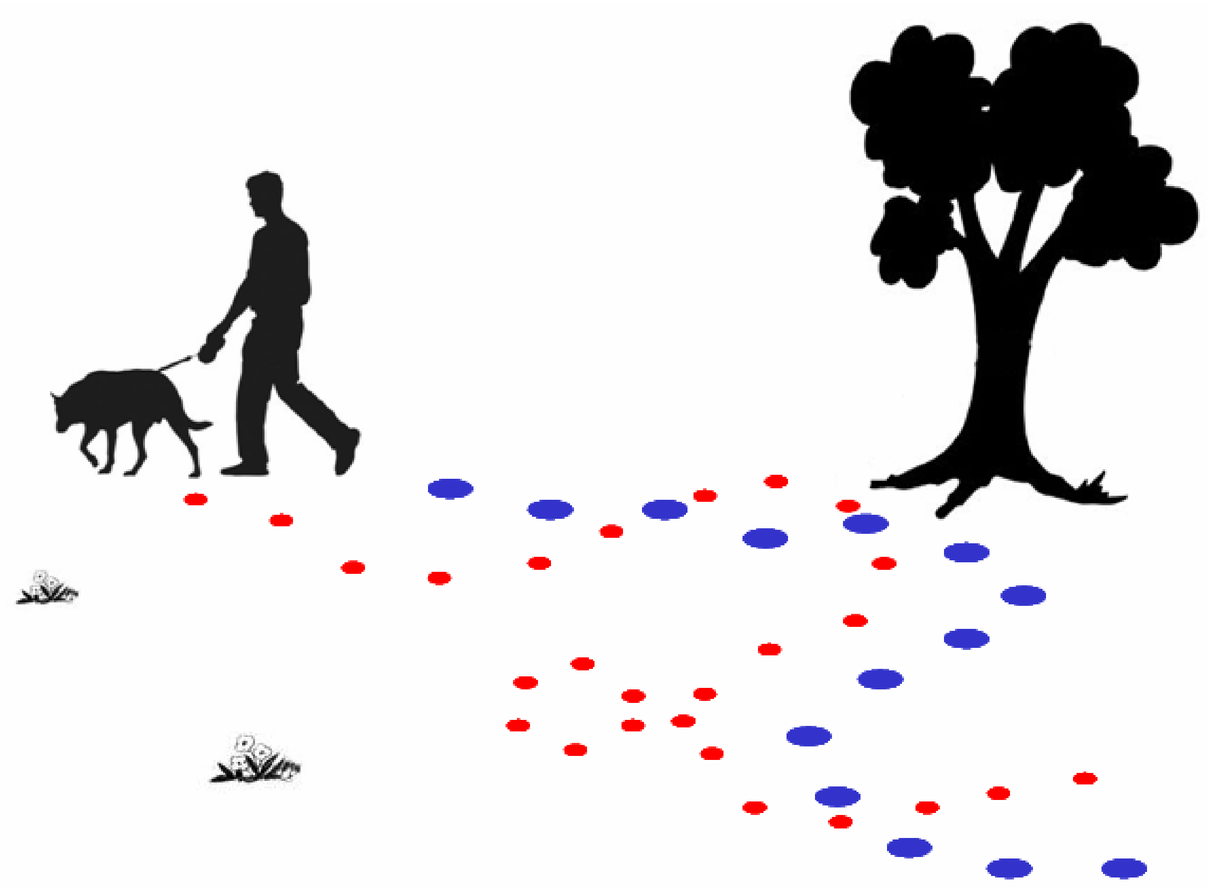
$GPGGA, <*utc*>,<*lat*>,N, <*lon*>,E, <*posfix*>,<*sat*>,<*hdop*>,<*alt*>,M,,,,0000*<*chk*>

| | |
|---|---|
| <*utc*> | UTC time in hhmmss.sss format (Hours, Minutes, Seconds), decimal representation |
| <*lat*> | Latitude in dd°mm.mmm (Degrees, Minutes), with North-South indicator, decimal representation |
| <*lon*> | Longitude in dd°mm.mmm (Degrees, Minutes), with East-West indicator, decimal representation |
| <*posfix*> | Position fix |
| <*sat*> | Number of satellites used |
| <*hdop*> | Horizontal dilution of precision, decimal floating point representation |
| <*alt*> | Altitude in meters, decimal floating point representation |
| <*chk*> | Standard NMEA checksum |

Example:

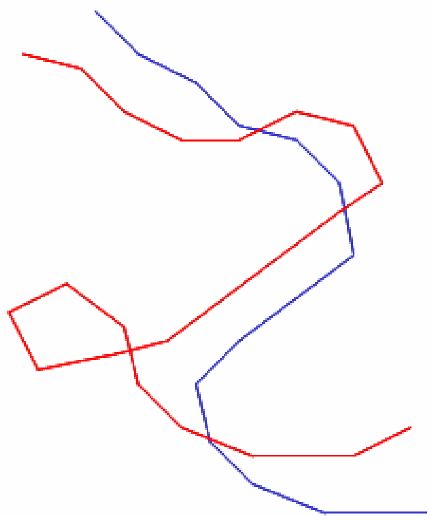
$GPGGA,163116.397,4647.5289,N,00709.6331,E,7,12,4,656.9,M,,,,0000*3B

Annex B - Illustration of the Fréchet distance between two curves.



Suppose a man is walking his dog. The man and his dog determine their own curves. Both are allowed to control their speed, but are not allowed to go backwards.

The Fréchet distance between the curves corresponds to the minimal length of the leash, which permits the man and his dog to pass their curves.



# CURRICULUM VITAE


| | |
|---|---|
| Name : | **Ivan Spassov** |
| Date of birth : | 09 November 1972 |
| Country of origin : | Bulgaria |
| e-mail : | ivan.spassov@epfl.ch |


## Education

| | |
|---|---|
| 2003 - 2007 | Doctoral student at the laboratory of topometry, EPFL<br>Supervisors: Prof. Bertrand Merminod and Prof. Michel Bierlaire<br>Thesis: Algorithms for Map-Aided Autonomous Indoor Pedestrian Positioning and Navigation |
| 1992 – 1997 | Master of Science in Geomatics<br>University of Architecture, Civil Engineering and Geodesy, Sofia, Bulgaria |
| 1987 – 1990 | Certificate in construction<br>Technical school of construction, Sofia, Bulgaria |

## Professional experience

| | |
|---|---|
| Sep.2000 - Fev.2003 | Geomatics for the urban heating system |
| Mar.2001 - Aug.2001 (stage) | Geomatics for the telecommunications |
| Dec.1998 - Mar.2000 | Digital photogrammetry and cartography |
| Sep.1997 - Nov.1998 | Geomatics for the rural management |

## Languages

French – fluent, English – fluent, Bulgarian – maternal language

## Principal publications

- Navigation pédestre dans un environnement construit, XYZ, N°100, Septembre 2004
- Indoor Navigation Enhanced by Map-Matching, European Journal of Navigation, Vol. 3, N°3, August 2005
- Bayesian Inference for Autonomous Personal Localization Indoors, ISTS 2006, September 2006, EPFL, Lausanne, Switzerland