Map-based Autonomous Personal Localisation Indoors

Ivan Spassov, EPFL – ENAC – TOPO
Michel Bierlaire, EPFL – ENAC – TRANSP-OR
Bertrand Merminod, EPFL – ENAC – TOPO

Conference paper STRC 2007

STRC | 7th Swiss Transport Research Conference
Monte Verità / Ascona, September 12. – 14. 2007

Map-based Autonomous Personal Localisation Indoors

Ivan Spassov
EPFL ENAC TOPO
Lausanne - Switzerland

Phone: 021.693.2751
Fax: 021.693.5740
email: ivan.spassov@epfl.ch

Michel Bierlaire
EPFL ENAC TRANSP-OR
Lausanne - Switzerland

Phone: 021.693.2537
Fax: 021.693.5570
email: michel.bierlaire@epfl.ch

Bertrand Merminod
EPFL ENAC TOPO
Lausanne - Switzerland

Phone: 021.693.2754
Fax: 021.693.5740
email: bertrand.merminod@epfl.ch

September 2007

Abstract

A novel method for autonomous personal localisation indoors is presented. It is based on inertial measurements of the human walk and information from the digital map database of the building.

Consider a person equipped with a navigation system which contains set of inertial sensors and map of the building. Speed, turn rate and barometric altitude are measured and time-stamped on each step of the person representing his trajectory as a sequence of points.

In our approach central place takes the association of the user's trajectory with the graph representation of the map, process known as map-matching. A pre-processing step detects critical movements of the person. After detection of turns and half-turns, the methodology is improved with new techniques that detect vertical movements such as taking elevator and staircase. Thus the trajectory is transformed into 3D polyline. Then similar geometric forms are identified in both the trajectory and the graph. So far the problem was addressed to one floor only (discussed last year). Recently new functionalities are added to the methodology so user's localisation can be performed considering all floors of the building.

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 position is entirely represented by its probability density function (PDF) in the frame of Bayesian inference. Localisation process starts with the determination of the edge of the graph occupied by the person. A recently developed methodology allows estimating more precisely the user's position on the edge. The method gives good results and assures a continuous localisation of the person.

Keywords

Bayesian – pedestrian – localisation – indoors – map-matching – map database

1. Introduction

With the development of complex buildings and structures inevitably arises the question of indoor personal navigation. Directing the user to his final goal, known as *route guidance*, is the main task of any navigation system. Normally, destination point is defined from the beginning and can be redefined in any moment. But in order to make possible the route guidance we need to determine the position of the person. In this research we focus on this fundamental problem of the navigation process, namely personal localization.

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 user's portable device and the network of distributed sensors.

The world famous positioning system is GPS, capable to determine user's position with an accuracy of up to 5 meters [Legat et al. 2000]. However, indoors the application of GPS generally is out of question. Assisted-GPS (A-GPS) dramatically improves the performance of GPS receivers [Abwerzger et al. 2004]. A-GPS is very useful 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. 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.

The motivation in our approach is the fully autonomous personal positioning indoors. So we need to find a way to position the person without relying on external measurements. That would set the user independent of the availability and drawbacks of those systems.

The only positioning method that allows this autonomy can be provided by an inertial navigation system (INS). Based on MEMS technology such system contains inertial sensors; it has its own power supply and can be easily carried by the user [Macheiner 2004]. Of course, the inertial navigation system is connected to user's portable device (e.g. PDA) which contains a *digital map database* of the region of interest [Zweiacker 2003]. So, using this minimal equipment the condition of autonomy is fulfilled and the question is how to determine the user's position.

We do not investigate the hardware of the navigation systems. We have to consider the information provided by the inertial measurements and the digital map database; to see how it could be treated and to send results to the user. So the solution must be found on software level, i.e. in the development of algorithms for positioning.

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 inertial measurements are our first source of data.

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 well known 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 position using information from the map database and inertial measurements of the navigation system. Our proposed solution associates the user's trajectory with the graph applying statistical methods in combination with map-matching. The methodology is divided in two stages – initial localization and continuous localization. The first stage aims at finding the location and orientation of the user's in the building, i.e. the edge of the graph where the user is. The second stage, determines the exact position of the person on that edge.

First of all, we need to answer the question: what are the elements of the trajectory that could be associated with some elements of the map database? Similar geometric forms must be identified in both the trajectory and the graph. Since the trajectory is defined by sequence of points, this set must be transformed to 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 movements of the pedestrian trajectory like turn, stop, vertical movement, etc. These critical movements are defined as points and connected in a 3D polyline thus representing an adequate input for the process of localisation.

After 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 we propose a solution 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 repetitively every time when new measurements become available.

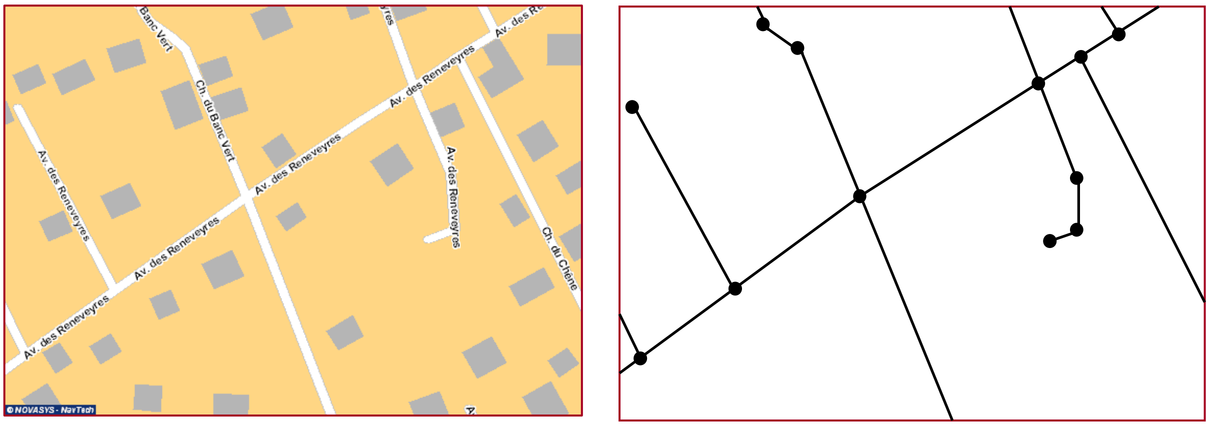
The proposed method consists in the development of algorithm for personal positioning. This method has many advantages. It is autonomous, i.e. allows personal positioning without relying on external positioning systems. It is simple to implement in the PDA and does not charge the user with additional equipment. It is not expensive in comparison with the sensor-based positioning methods.

2. Map database

Generally the geographical database contains information on the position, dimensions, capacity, functionality, etc. of the geographical objects.

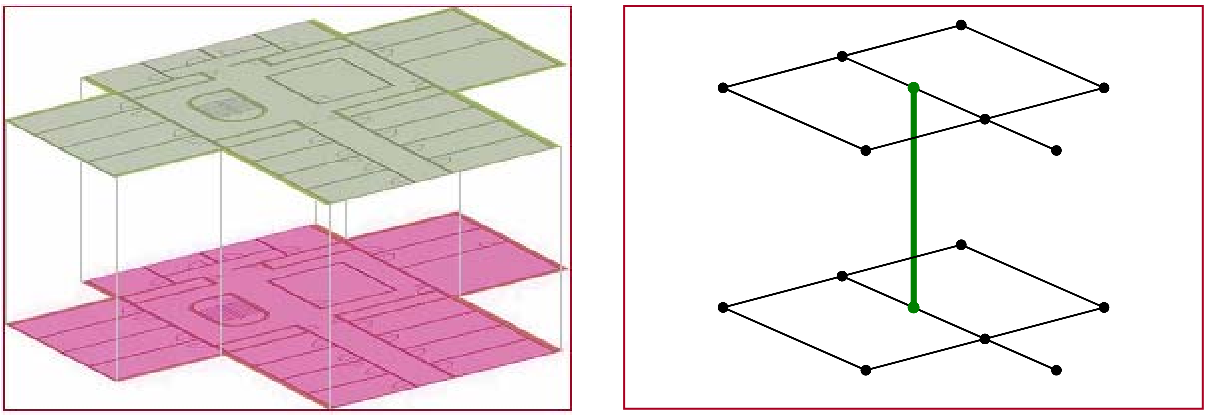
For the purposes of the navigation process the connections between these objects are of interest. In the urban areas these connections are defined by the street network, which is represented by a planar graph [Bernstein, Kornhauser 1996]. The streets are defined by edges or links and the crossings – by nodes (Fig.1). That graph representation of the street network is created by applying a *link-node model*. In order to create the map database for a building the same link-node model is used [Gilliéron et al. 2004]. This model includes all connections like corridors, passageways, elevators and staircases. The start and the end of each edge are defined by a node. Edges are assumed to coincide with the axis of the corridors (Fig.2). Each edge connects two nodes and each node is known with its coordinates in a fixed coordinate system (e.g. in the national coordinate system).

Figure 1 Street network represented by the link-node model as a graph



Thus the graph is absolutely defined. Using the node coordinates different properties of the edges, like length and azimuth could be computed. An important property of the building data model is that the vertical connections are considered.

Figure 2 Floors of building represented by the link-node model



The elevators and staircases are represented as edges connecting two nodes from different floors. In the context of indoor pedestrian navigation the map database could be constituted by the limits of the building. However, it could be connected with the street network database or with the map database of other buildings. Based on that graph representation, there exist many algorithms like computing the shortest path between two points of interest.

3. Initial localization

The localization methods aim at determining the location of the user. Considering the 3D graph representation of the building, in our approach 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.

3.1 Pre-processing of the raw data

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.

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

v - Speed (knots)

r - Heading (degrees)

h - Barometric altitude (meters)

t - Time (hhmmss.sss)

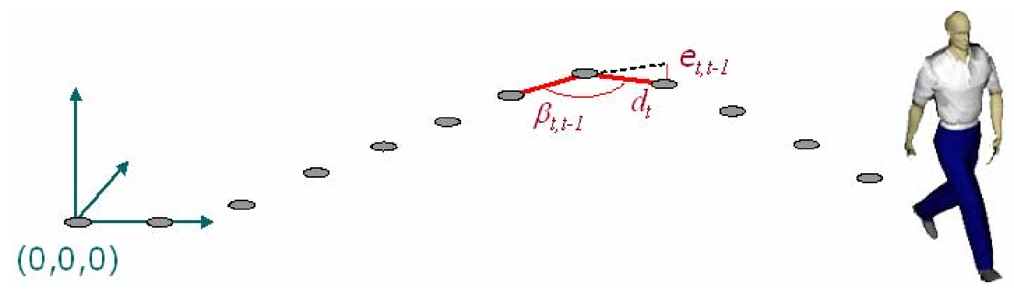
In terms of geometry the trajectory can be considered as a sequence of points where the raw data (Speed, Heading, Barometric altitude and Time) is known at each point.

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).

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

These parameters are computed for each point of trajectory, i.e. every time new measurements become available (Fig. 3).

Figure 3 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 rests a sequence of points and at this stage could not be associated to 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 exceptionally dependent on this transformation.

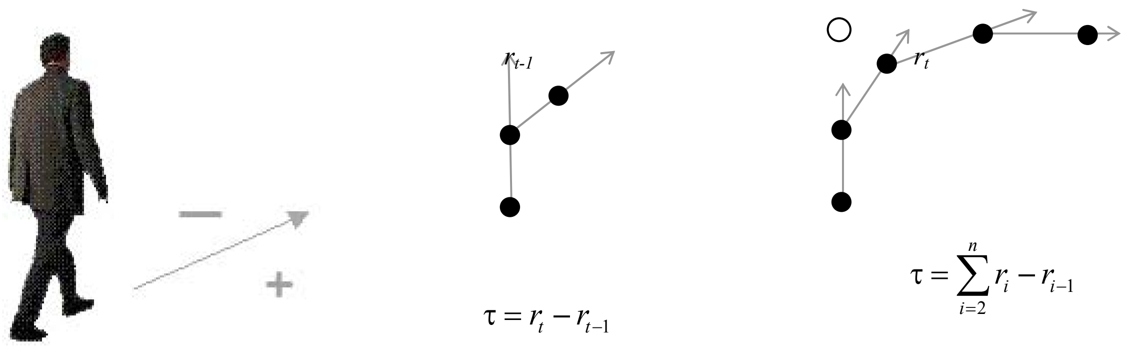
The idea here is to detect the critical movements of the person like turns and vertical movements (taking elevator or stair case) and define them as *critical points* of trajectory. Then, the 3D polyline will be formed by segments connecting consequently the critical points.

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 δ_t of segment at moment t.

Turn detection

Here we will 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 person goes straight (Fig. 4a). The person can make a turn with sharp change of direction in one step only (Fig. 4b) or spread over several steps (Fig. 4c).

Figure 4 Detection of the turns



(a) (b) (c)

During the walk in straight direction the measured heading differs from a step to another in the range of ±2 degrees. 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 figure 4, the value of τ is computed. If the person changes his direction in one step the value of τ will correspond to the difference of two consecutive headings (Fig. 4b). We can not consider each change of direction as turn, so we need to establish a threshold for the value of τ which corresponds to a turn. For that we have proceeded to several tests in the buildings of our campus. We have got an empirically derived threshold of 18°. So, changes of direction that give $|\tau| \leq 18°$ are not considered as turns.

The detection of every turn must be indicated by a critical point mentioned above. For the case of figure 4b the critical point coincides with the step position where the turn has been made. In the other case (Fig. 4c) 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. 4). The detection of turns defines the critical points of the trajectory in horizontal sense.

Vertical movement detection (Change-of-floor)

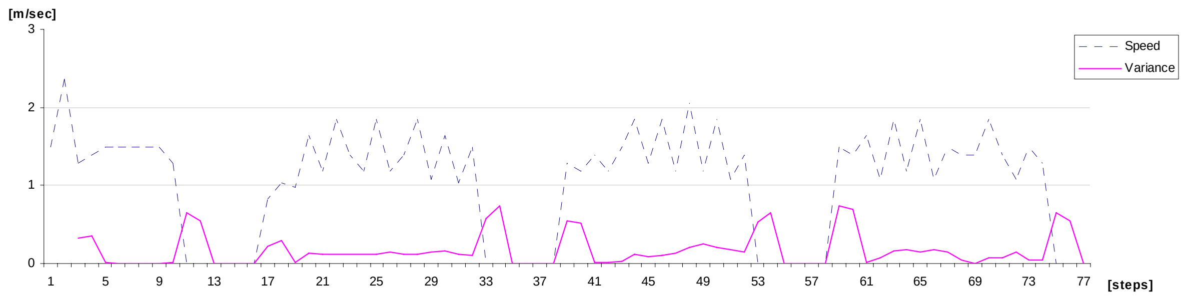
The other important critical points are those who represent a vertical movement of the user, i.e. taking elevator or staircase. The measurement that indicates the advancement of the trajectory in vertical direction is the barometric altitude. However, detecting a vertical movement is not an easy task if we use barometric measurements only. The reason is that the personal navigation system use low-cost barometer, its measurement error is very big and does not allow for detection of vertical displacement on every step [Lachapelle et al. 2003].

In order to be able to detect a vertical movement (change-of-floor) we need additional information, besides the barometric altitude. This time the idea is to analyse the behaviour of the user when move from one floor to another and 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. That change in the speed will give us only additional information, but the detection of vertical movement will be based on the main source of data, the raw measurements.

Based on that phenomenon we define four different state events that the user can perform: *go, accelerate, slow* and *stop*. If we detect the points of the trajectory where these state events occur that would give us additional information on the user's behaviour. That information will be reflected in the speed measurements and the task is 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 and there will be an important difference between their barometric altitudes.

For detect the state points the speed variance is computed on every step taking the last 3 steps. Figure 5 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.

Figure 5 Speed and speed variance of stop-go trajectory



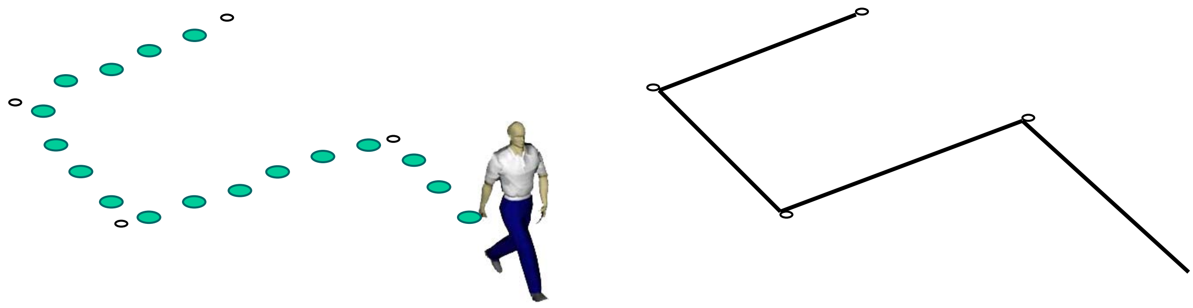
We have proceeded to several test trajectories in order to define a threshold for the speed variance for which a state point is detected. Based on these tests and on a engineering judgement an empirical threshold is evaluated. Thus, if the variance is bigger than 0.26, a state point is indicated.

Then for detect change-of-floor the elevation between every pair of state points is computed by comparing their barometric altitudes. The computed elevation approximates the total height of the floors, ascended/descended by the user. The precision of the barometric altitude 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.

Figure 6 The sequence of points (a) generalized by a polyline (b)



a) 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. Figure 6 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.

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*. If the user has passed several floors, to each floor a vertical segment corresponds.

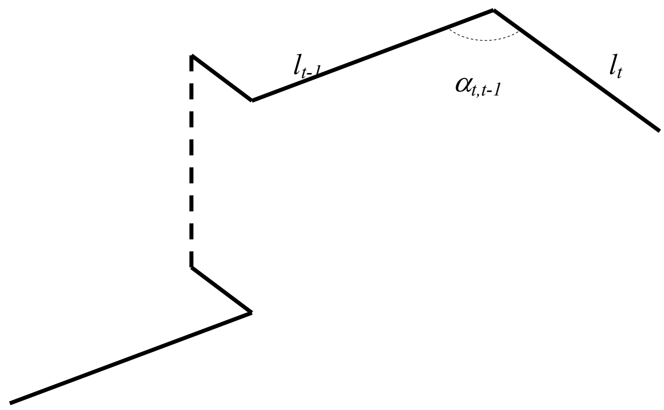
In the pre-processing step the changes of the floor (staircase or elevator) are always represented by vertical segments. Considering the staircases and elevators as devices for move from one floor to another, we assume their functionality as topological connections rather than spatial connections. It 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.

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 δ_t of segment at moment t (Fig. 7).

These parameters constitute the polyline as a sequence of segments. Every time new segment is fixed, 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 new set of relative parameters (l_t, $\alpha_{t,t-1}$, δ_t).

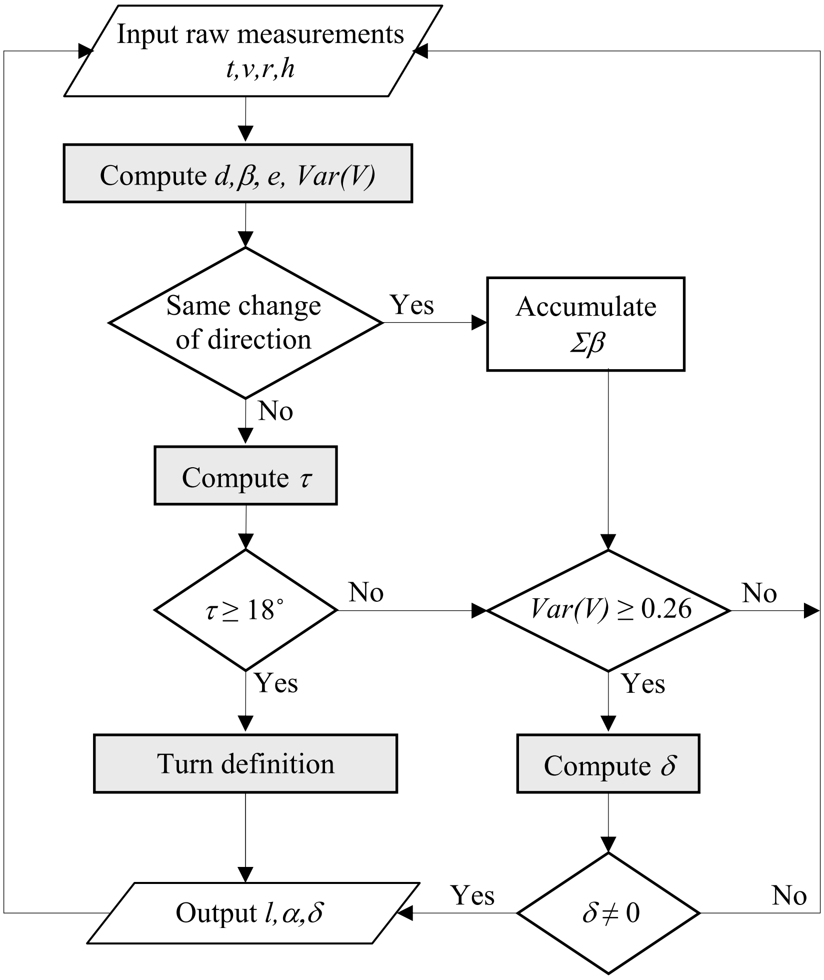
Figure 7 Relative parameters of the 3D polyline at moment t. The dashed line represents a vertical movement



Special attention must be paid to the elevation δ_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 (δ_t = 1), take one floor down (δ_t = -1), stays on the same floor (δ_t = 0).

A flowchart of the pre-processing step is shown in figure 8. 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 8 Flowchart of the pre-processing step



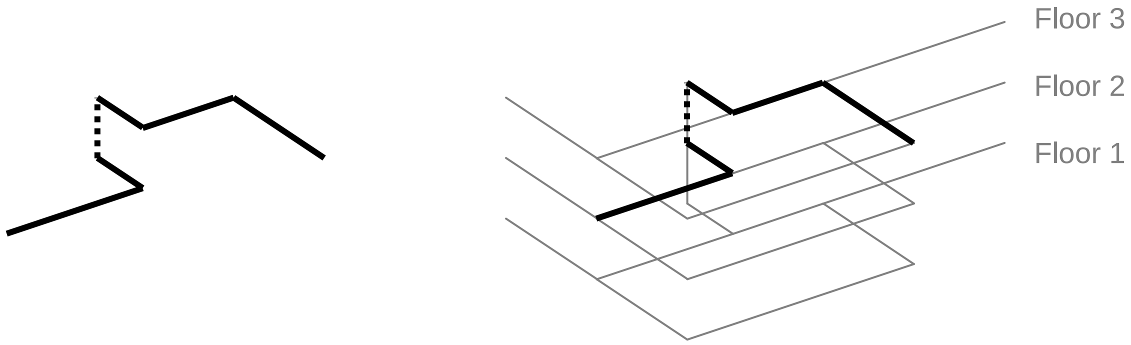
3.2 Problematic of the localization

The initial localization is to find the edge of the graph occupied by the person and 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. 9a) covers certain part of the graph of the building (Fig. 9b).

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 9 The 3D polyline as part of the graph



The aim is to find in the contents of the graph the set of successive edges that fits best the form of the polyline at moment *t*.

Every time new critical point is detected, 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. 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.

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.

3.3 Bayesian formulation

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. Here we discuss how the Bayesian inference is applied in our approach.

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:

$$x_t = f\left(x_{t-1}, u_t\right) \tag{2a}$$

$$y_t = h\left(e^{(i)}, e^{(i+1)}\right) + z_t \tag{2b}$$

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 (2a) 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 (2b) 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 $X_t=\{x_0 , x_1 ,..., x_t\}$, respectively $Y_t=\{ y_1 , y_2 ,..., y_t\}$ 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.

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:

$$p(x_t|x_{t-1}, x_{t-2}, \ldots, x_0) = p(x_t|x_{t-1}), \quad \text{and} \quad p(x_0|x_{-1}) = p(x_0) \tag{3}$$

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].

$$\begin{aligned} p(x_t|Y_t) &= \frac{p(Y_t|x_t)p(x_t)}{p(Y_t)} \\ &= \frac{p(y_t, Y_{t-1}|x_t)p(x_t)}{p(y_t, Y_{t-1})} \\ &= \frac{p(y_t|Y_{t-1}, x_t)p(Y_{t-1}|x_t)p(x_t)}{p(y_t|Y_{t-1})p(Y_{t-1})} \\ &= \frac{p(y_t|Y_{t-1}, x_t)p(x_t|Y_{t-1})p(Y_{t-1})p(x_t)}{p(y_t|Y_{t-1})p(Y_{t-1})p(x_t)} \\ &= \frac{p(y_t|Y_{t-1}, x_t)p(x_t|Y_{t-1})}{p(y_t|Y_{t-1})} \end{aligned} \tag{4}$$

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 $i=1 \ldots n_e$ 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:

$$\Delta l^{(i)} = l_t - L(e^{(i)}) \tag{5a}$$

$$\Delta \alpha^{(i)} = \alpha_t - B(e^{(i)}, e^{(i+1)}) \tag{5b}$$

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 α_t and $B(e^{(i)}, e^{(i+1)})$. These residuals are used to compute:

$$w_l^{(i)} = 1 - \frac{\Delta l^{(i)}}{\sum_{i=1}^{n_e} \Delta l^{(i)}} \tag{6a}$$

$$w_\alpha^{(i)} = 1 - \frac{\Delta \alpha^{(i)}}{\sum_{i=1}^{n_e} \Delta \alpha^{(i)}} \tag{6b}$$

And then:

$$w_m^{(i)} = w_l^{(i)} \cdot w_\alpha^{(i)} \tag{7}$$

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 α_t. That is the case where no vertical movement is detected and δ_t is not taken into account. Respectively, in the case of vertical movement we treat only δ_t without taking into account α_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 δ_t, which will indicate an elevation or descending. We write:

$$w_m^{(i)} = \begin{cases} 1, & if\, \delta_t = \Delta(e^{(i)}) \\ 0, & otherwise \end{cases} \tag{8}$$

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:

$$w_h^{(i)} = \prod_{j=2}^{T-1} q^{(j)} \tag{9}$$

where

$$q^{(j)} = \begin{cases} 1, & \text{if } e^{(j-1)} \to e^{(j)} \quad \text{and} \quad p(y_j | Y_{j-1}, x_j, x_{j-1}) \to 1 \\ 0, & \text{otherwise} \end{cases} \tag{10}$$

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 sum of both sub weights:

$$w_t^{(i)} = w_m^{(i)} \cdot w_h^{(i)} \tag{11}$$

Finally, for the likelihood we write:

$$p(y_t | Y_{t-1}, x_t, x_{t-1}) = p(y_t | x_t, x_{t-1}) p(x_{t-1} | Y_{t-1}) \tag{12}$$

Following the concept of the Bayesian theorem we will compute the posterior probability multiplying the likelihood by the prior.

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

$$p(x_t | Y_{t-1}) = \sum_{x_{t-1}} p(x_t | x_{t-1}) p(x_{t-1} | Y_{t-1}) \tag{13}$$

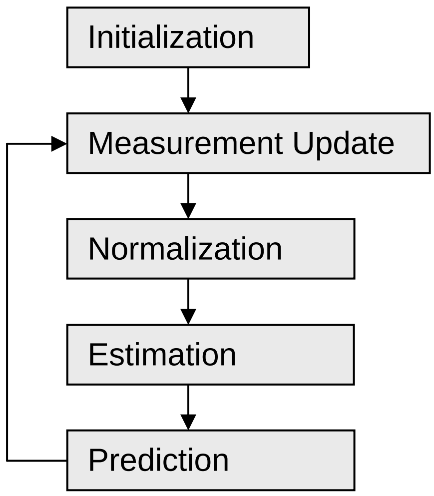
The quantity $p(x_{t-1}|Y_{t-1})$ is available from the computation of the posterior probability for the last segment, and the model $p(x_t|x_{t-1})$ simply characterizes the topology of the graph. Consider x_{t-1} as estimation at moment t. Thus, $p(x_t|x_{t-1}) = 1$ if x_t is a possible successor of 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.

3.4 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.

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 figure 10 and are discussed in details further.

Figure 10 Principal phases of the algorithm for initial localization



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. 11a). It can be written as follows:

$$w_0^{(i)} = \frac{1}{n_e}, \quad i = 1,...,n_e \tag{14}$$

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 11b). 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 (11). 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).

In the *normalization* phase the updated weights are normalized as follows:

$$\overline{w}_t^{(i)} = \frac{w_t^{(i)}}{\sum_{i=1}^{n_e} w_t^{(i)}}, \quad i = 1,...,n_e \tag{15}$$

After the normalization we estimate the location of the person on the map by choosing certain edges whose weight ($\hat{x}_t$) is not smaller than a threshold. That choice is illustrated on figure 11c. 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. We write:

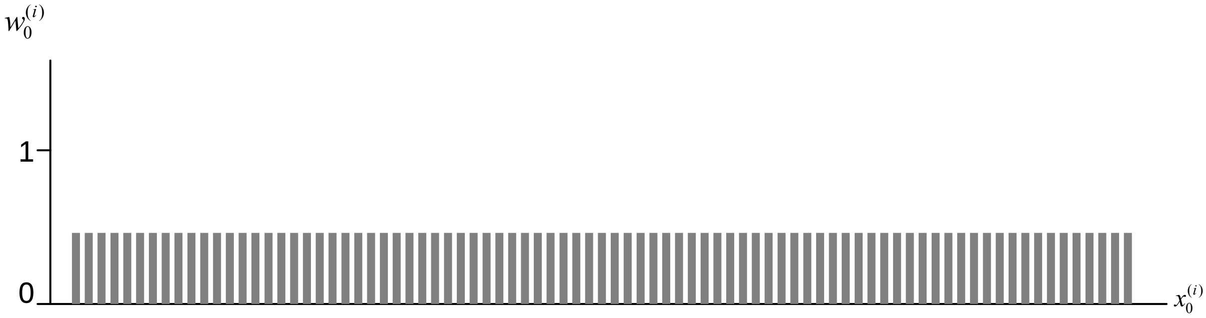
$$\hat{x}_t = x_t^{(i)} \left| \left(\overline{w}_t^{(i)} \geq \overline{w}_t^{(Th)} \right) \right., \quad i = 1,...,n_e \tag{16}$$

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:

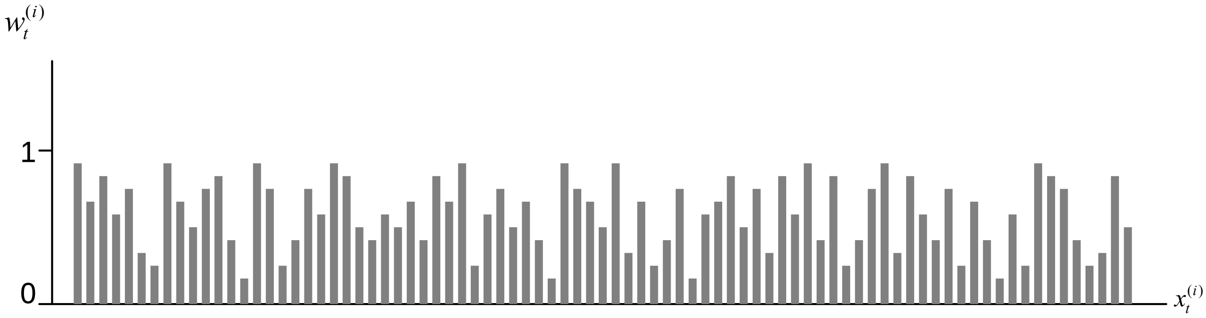
$$w_{t+1}^{(i)} = \begin{cases} 1 & ,\hat{x}_t \quad is \quad neighbor \quad of \quad x_{t+1}^{(i)} \\ 0 & ,\text{otherwise} \end{cases}, \quad i = 1,...,n_e \tag{17}$$

Here $w_{t+1}^{(i)}$ is the prior weight of the edge i. The neighbours of the last estimated edges are supposed to be the predicted edges and receive weight 1. The other edges of the graph receive weight 0 (Fig. 11d).

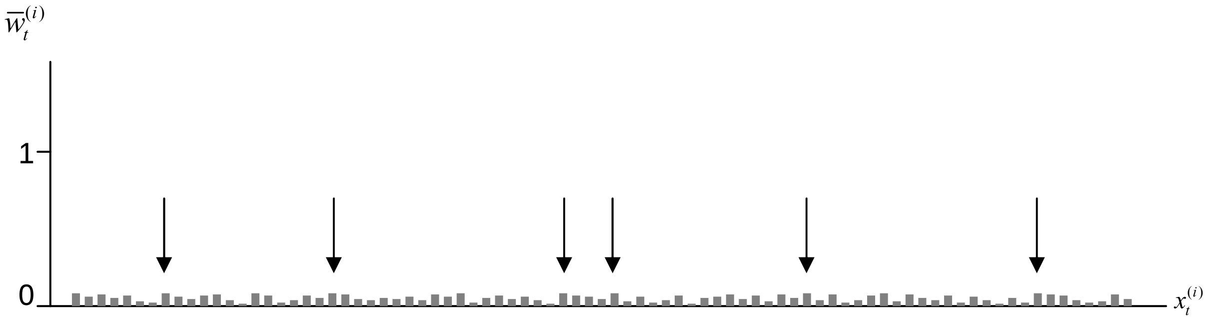
Figure 11 Principal phases of the algorithm for initial localization. The edges of the graph are settled on the abscissa. The ordinate reflects the weights of the edges



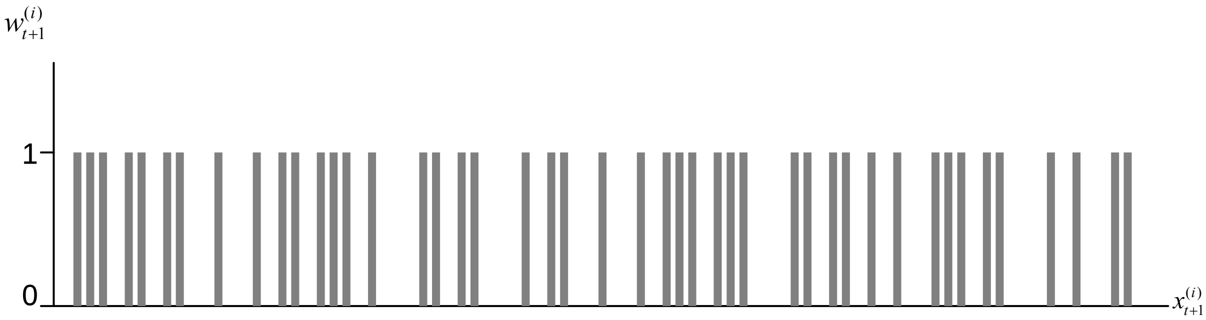
a)



b)



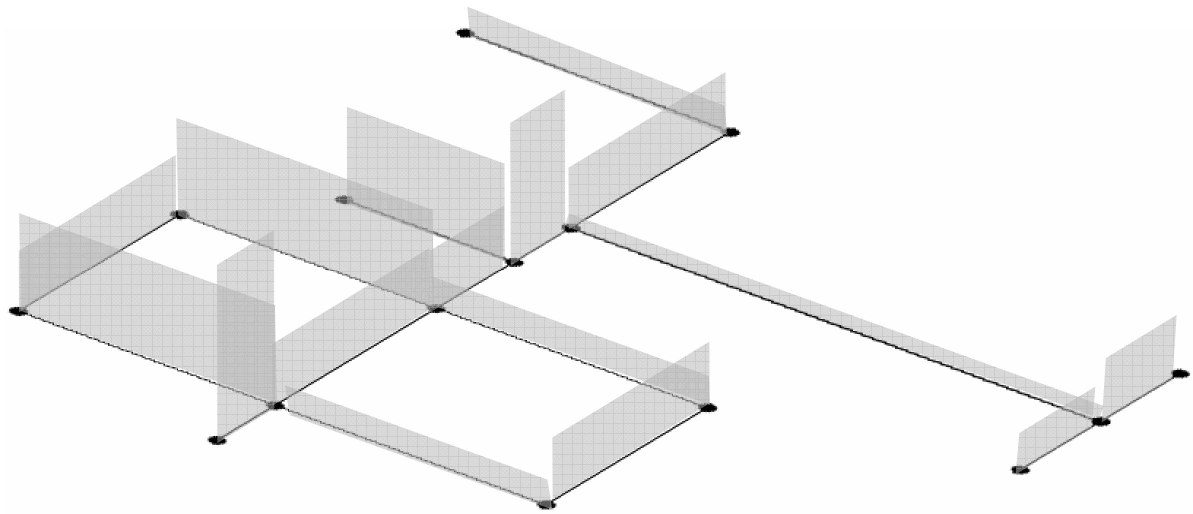
c)



d)

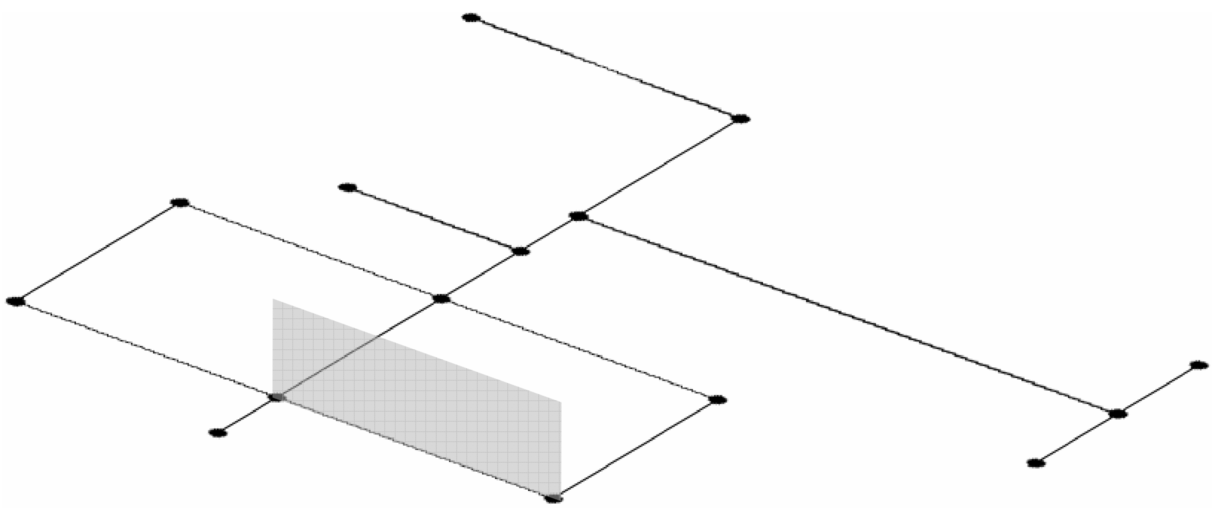
At moment t the weights computed for each edge of the graph define the probability distribution as discrete multimodal distribution. Those edges that possess the highest weight are estimated as best match to the last segment of the 3D polyline (Fig. 12).

Figure 12 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. 13).

Figure 13 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.

Figure 14 Flowchart of the initial location

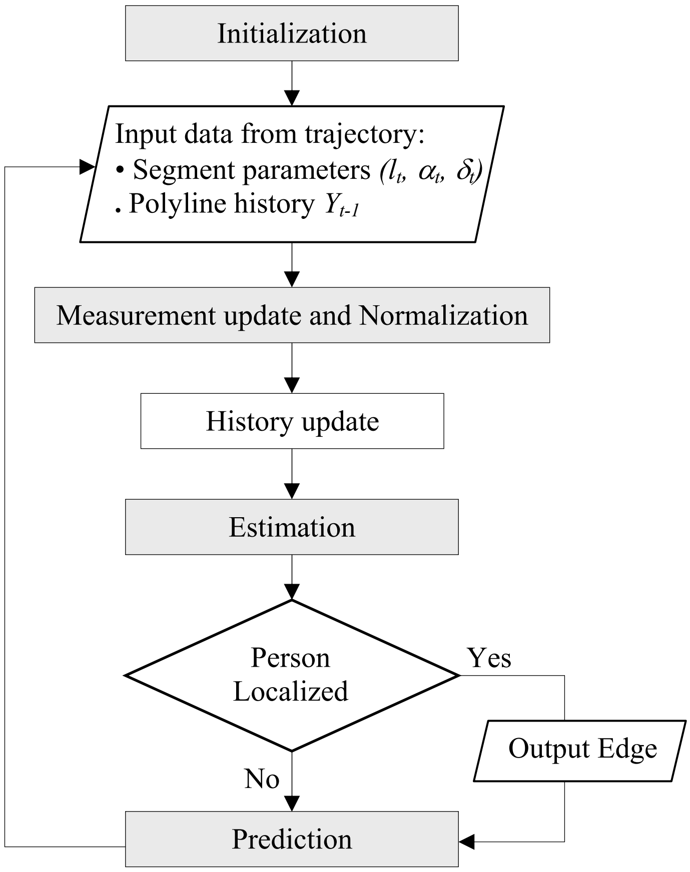


Figure 14 gives a general view of the algorithm of initial localization. The gray cages present the main phases which are explicitly discussed further in separate flowcharts.

The phase of *history update* aims at the 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 15 Flowchart of the phases of measurement update and normalization

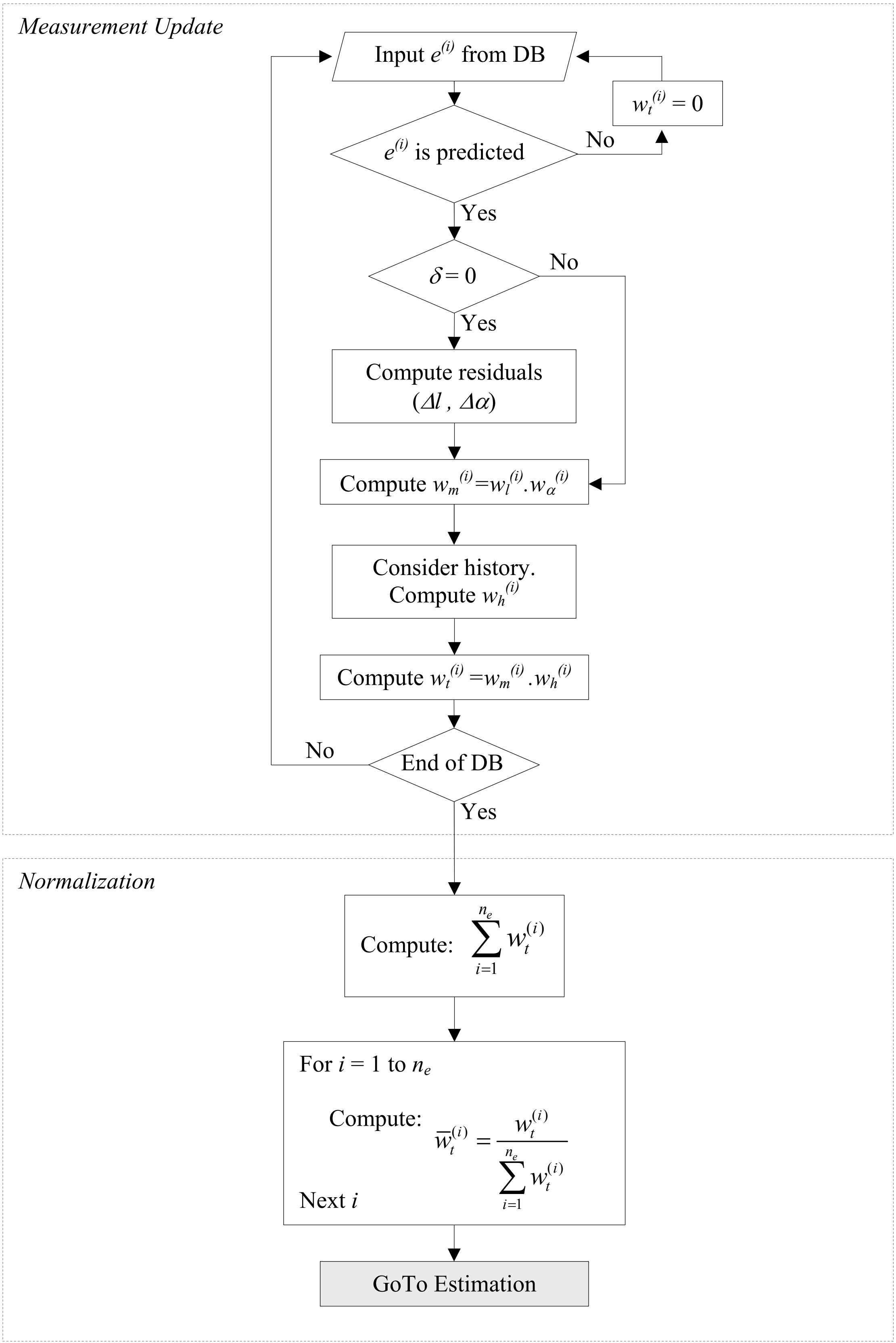


Figure 16 Flowchart of the estimation phase

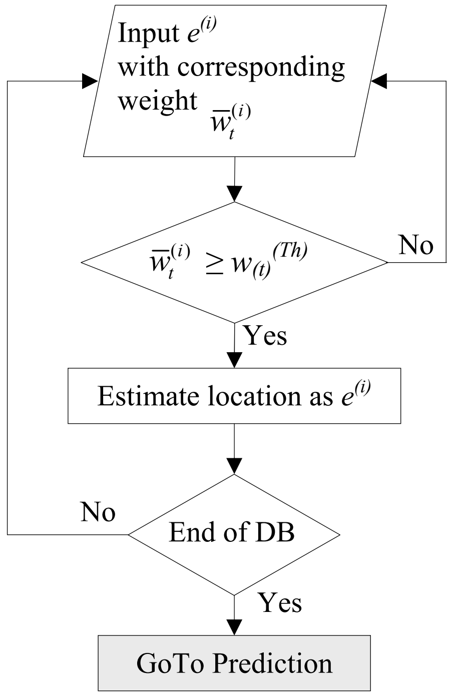
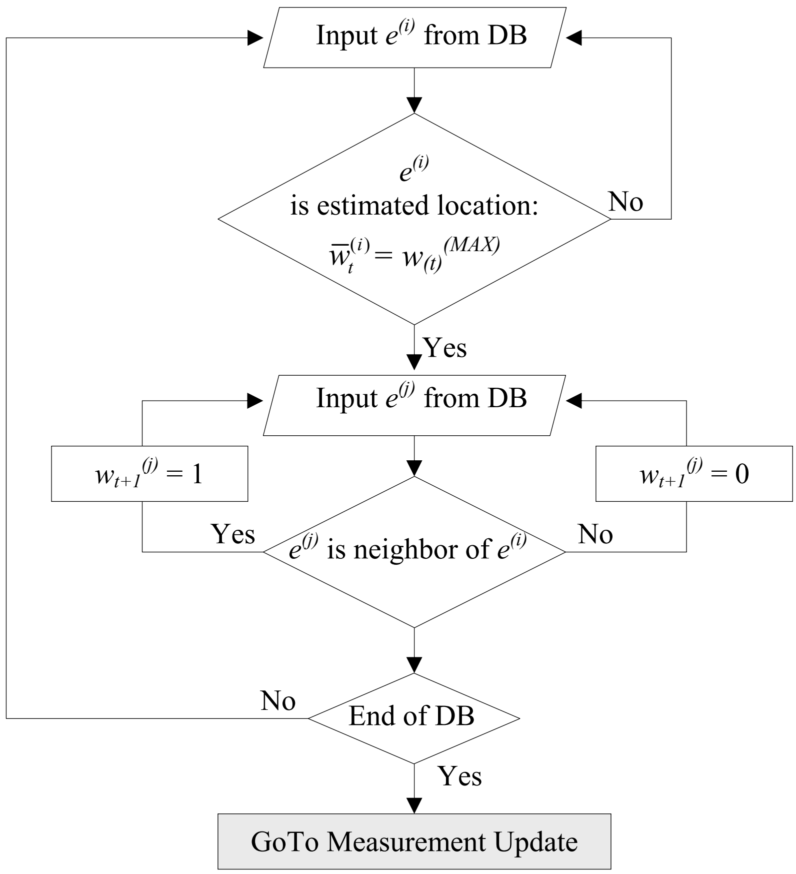


Figure 17 Flowchart of the prediction phase



3.5 Tests, Results and analysis

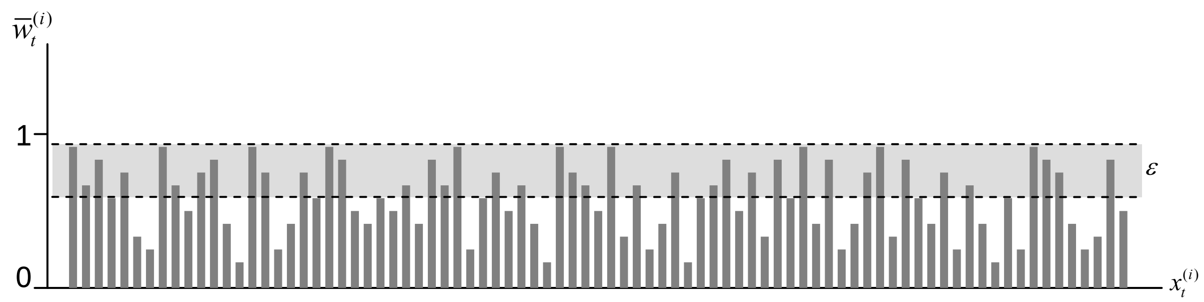
Although the algorithm is designed to work in real time mode, it is written and tested in post-treatment mode. Several scenarios were made to test the robustness and the efficiency of the algorithm for initial localization. Stair-cases and elevators were included in some of the trajectories. In the tests four persons with different height were involved. Table 1 shows the results.

Table 1

No. Trajectory	Person localized	No. iterations before localization	Walked distance before localization	ε [%]
1	Yes	5	25	5
2	Yes	6	21	10
3	Yes	8	29	10
4	Yes	6	22	5
5	Yes	6	45	25
6	Yes	6	30	15
7	Yes	5	31	5
8	No	-	-	-
9	No	-	-	-
10	No	-	-	-

We have introduced a parameter ε that defines what weight must have an edge in order to be estimated as user's location. It defines a threshold for the weights of the edges in the estimation phase.

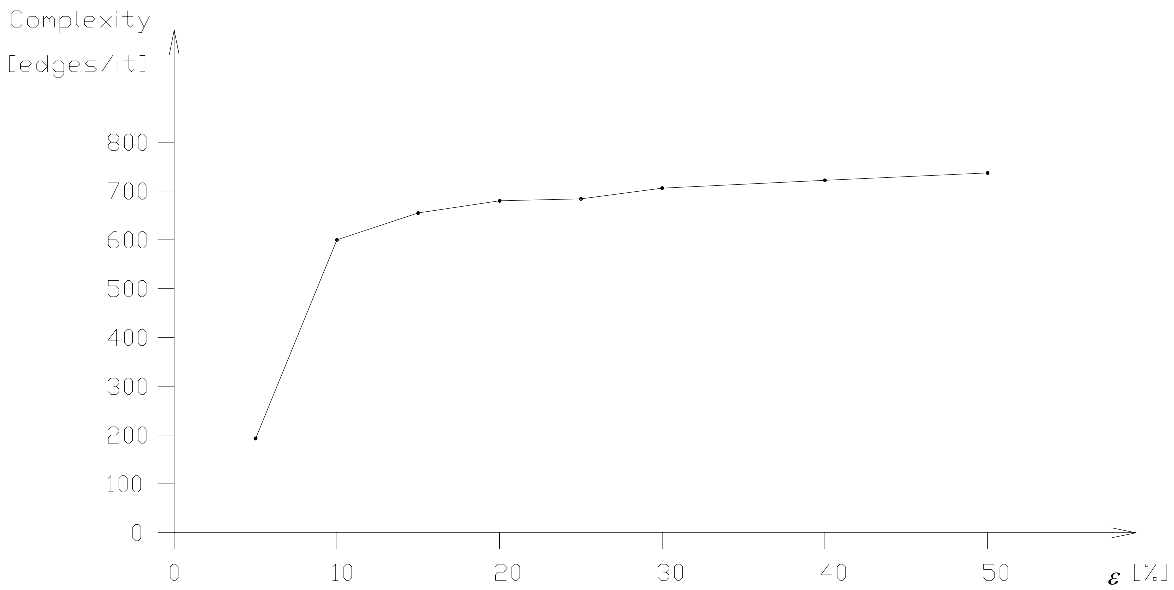
Figure 18 Definition of the parameter ε. In this figure ε = 35%.



The reason to introduce this parameter is the following. If we consider in the estimation phase of the algorithm only the edges with maximal weight as user's location, we ignore the rest of the edges. It is possible that at moment *t* the location edge has not a maximal weight, but a "near-maximal" weight. Thus, we risk not estimating the correct location of the user. For avoid such faults we need to define a limit for the weights considered in the estimation phase. That limit is represented as percentage of the maximal weight (Fig. 18).

We have analyzed the algorithm for its efficiency (whether the user is localized or not) and complexity (what is the memory consummation) as function of the parameter ε (Fig. 19).

Figure 19 Graph of complexity vs. ε



Logically, with the augmentation of ε the complexity grows, i.e. more edges are treated on each iteration. There is not the same relation between ε and the efficiency. There, the algorithm is efficient for values of ε limited by a threshold. The aim of this analysis is to define an optimal value of ε for which the complexity is minimal and the algorithm is efficient. Based on those test trajectories (1-7) where the person is localized we have defined an optimal value for ε of 11%.

Another point represents the number of edges taken into account as history of movement. Generally, the big number of edges in the history increases the complexity, but the process is more robust. Tests show that the efficacy gets sensibly low if we consider less than 4 edges.

Besides the analysis of the complexity and the efficiency we made the following observations. The geometry of the building plays an important role for the localization process. If the building has a symmetric geometry (repetitive elements), it will take more time to recognize the polyline (trajectory) in the contents of the 3D graph and to localize the user.

A crucial moment for the localization appears the use of elevator or stair-case (represented with vertical edges). The number of vertical edges in the 3D graph is relatively small. If the trajectory passes through a vertical edge (elevator or stair-case) the number of candidate edges will be drastically reduced which will hasten the localization process.

The methodology that we discuss here is not universal. There exist cases where the person is not localized (trajectories 8-10 in Table1). There are different reasons and we can subdivide them in two types – errors made during the tests, and methodological reasons springing from the imperfection of the approach chosen in this research.

The errors could be different, mainly incorrect attachment of the PNM (measurement module) on the 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 accord with the factory instructions.

As methodological reasons we consider firstly the concept of graph representation of the building. 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 probability for the passage of people. This can be included as a perspective to the future developments in this topic.

A very delicate point of the methodology is the detection of state points (where we use the speed variance). In some cases the state points are not detected, while in other cases state points are detected incorrectly. We can refer again to the correct attachment of the PNM. However, most of the problems come from methodological point of view. This technique for detect vertical movements works well in the case of taking elevator. Taking a stair-case differs, because the user can change his speed in the stair-case, typically running. On the other hand, most of the staircases have doors. To access the user must perform some sophisticated movements, including step back and turn around. At this stage of the research and with this poor set of input information (measurements and map database), such a delicate point of the methodology is not surprising and demands future development.

Another important detail is the performance of sophisticated movements. The localization process relies on the hypothesis that the user performs normal walk in the building. There are many examples of particular movements that can break the process: running, entry and exit room, stop to talk with someone, etc. Therefore, the user must avoid that kind of movements in the phase of initial localization.

4. 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.

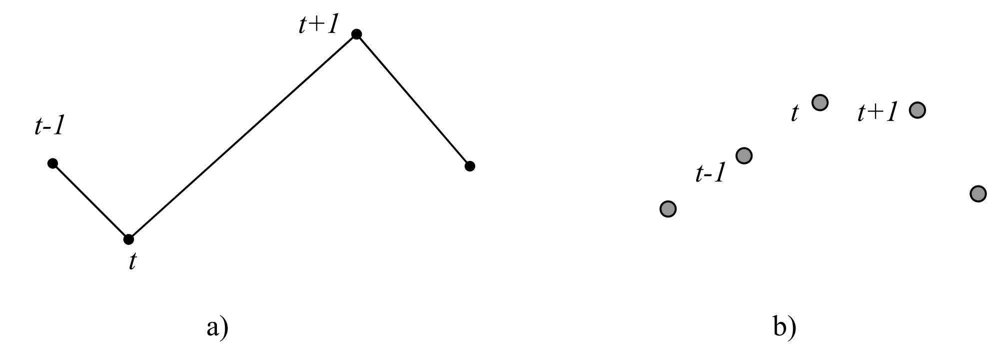
Now we need to determine where exactly on the location edge is the person. Contrary to the initial localization in the continuous localization user's location is not presented as edge but as a point, named for simplicity *location point*. That point is considered as a part of the edge and will be estimated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

4.1 Theoretical formulation

In order to assure a continuous localization process we need to know 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.

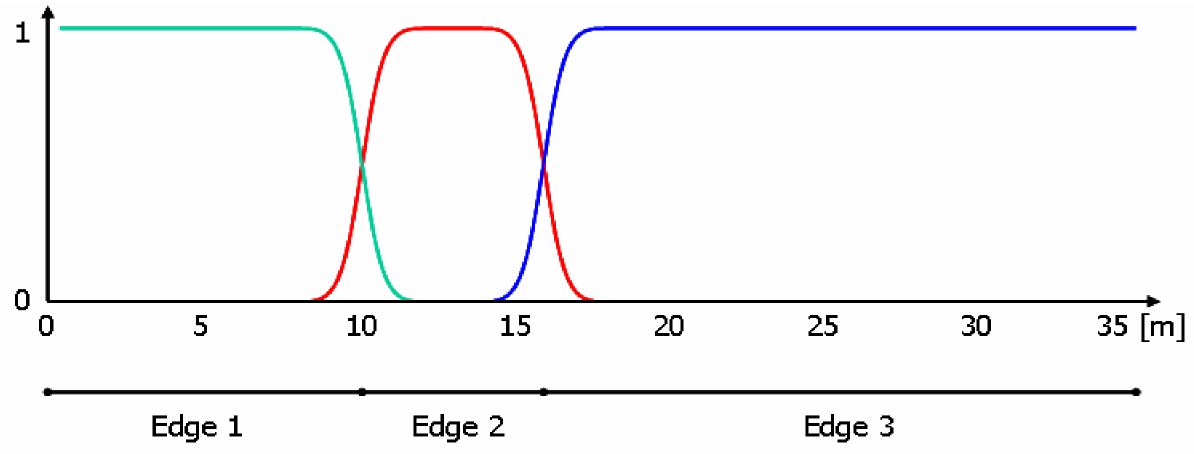
The determination of the correct edge is based on the Bayesian inference, but the methodology is different from that in the initial localization. First, the time discretization of the process reflects the acquisition of new measurements on every step (Fig. 19) and not on the definition of every polyline segment. Second, the likelihood computation is based on the distance and heading of every stride instead of the polyline parameters.

Figure 20 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. Figure 20 shows three neighbour edges and their probabilities to be the location edge depending on the passed distance.

Figure 21 Probability of 3 neighbour edges



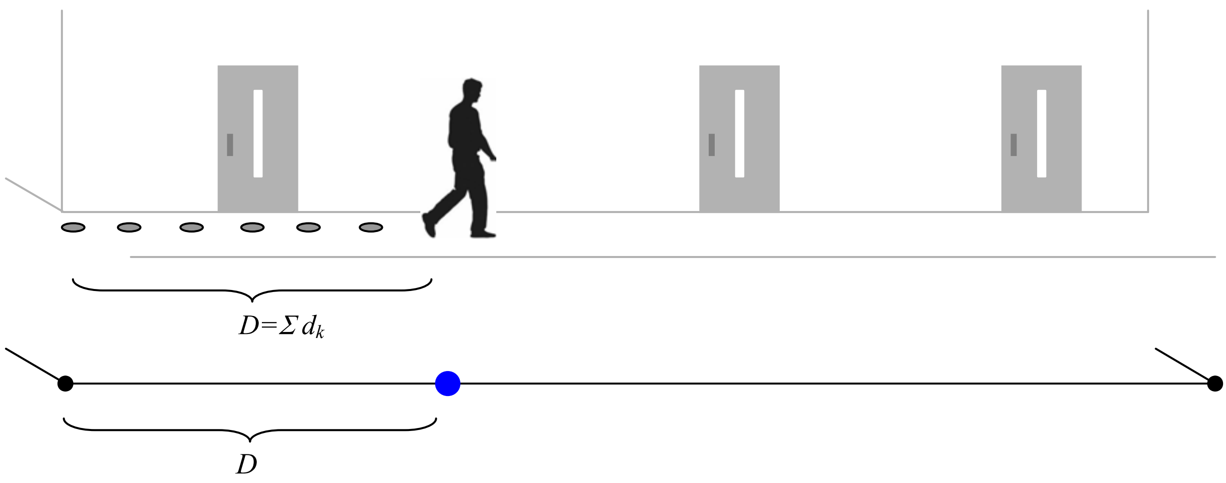
The moment of stepping from one edge to another is of great importance in this approach. At this moment we start to accumulate the distance of the strides from the beginning of the edge.

$$D_k = \sum d_k \tag{18}$$

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. 21).

Figure 22 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 ($k = 0$).

4.2 Algorithm

The main task 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:

$$\gamma_t^{(j)} = A(e^{(j)}) - \sum\left(\hat{r}_t - \hat{r}_{t-1}\right) \tag{19}$$

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:

$$w_t^{(j)} = 1 - \frac{\gamma_t^{(j)}}{\sum_{j=1}^{G} \gamma_t^{(j)}} \tag{20}$$

where G is the set of the candidate edges.

The update phase is performed on every user's step. As estimation of the location edge at moment t we take the edge with maximal weight $w_t^{(j)}$.

$$\hat{e}_t = e_t^{(j)} \left| \left(w_t^{(j)} = w_t^{(MAX)} \right), \quad j = 1, \ldots, G \right. \tag{21}$$

where $\hat{e}_t$ is the estimated location edge. This phase is followed by the estimation of the location point (Fig. 21).

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).

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. 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:

$$D_k - L(e^{(j)}) \leq 4\%\left(D_k\right) \tag{22}$$

At that moment there can be several candidate edges $e^{(j)}$, $j=1,\ldots,G$. 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 figures 22 and 23.

Figure 23 Update phase of the continuous localization

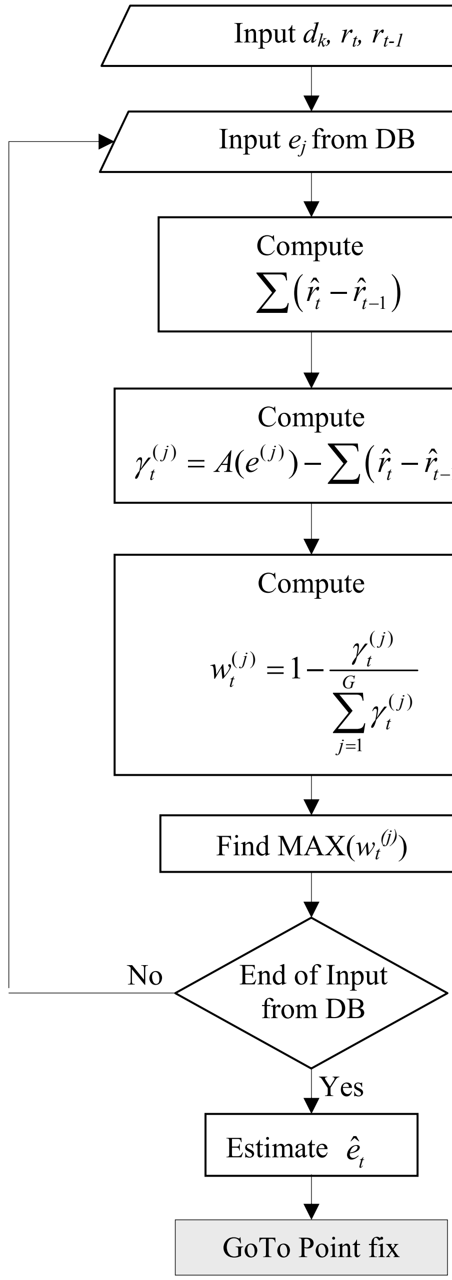
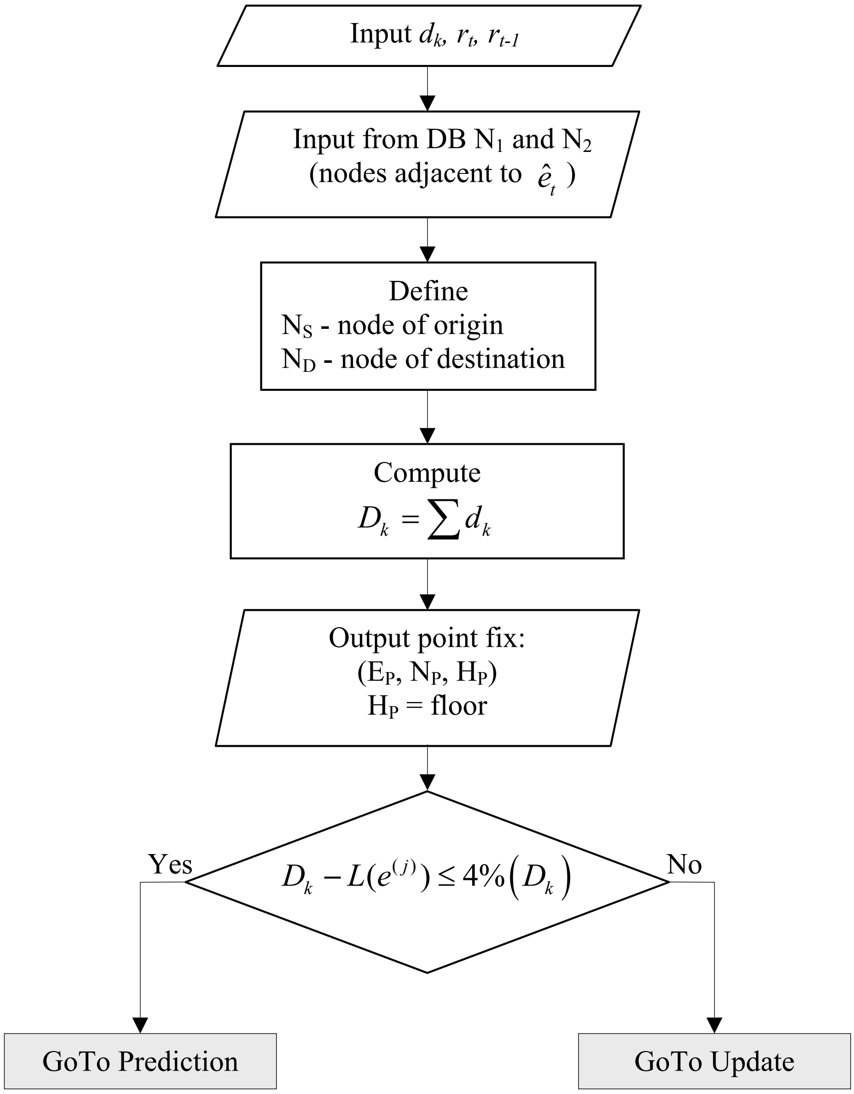


Figure 24 Estimation of the point fix on the location edge



4.3 Tests, results and analysis

This algorithm for continuous localization is tested with many trajectories and the analysed aspects are the precision 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* (18). 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 2 shows the results of this comparison for some of the edges passed in the test trajectories.

From the results in Table 2 we can easily distinguish the biggest error of -1.19m. However we can not say that a bigger error corresponds to bigger distance.

Table 2

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

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.

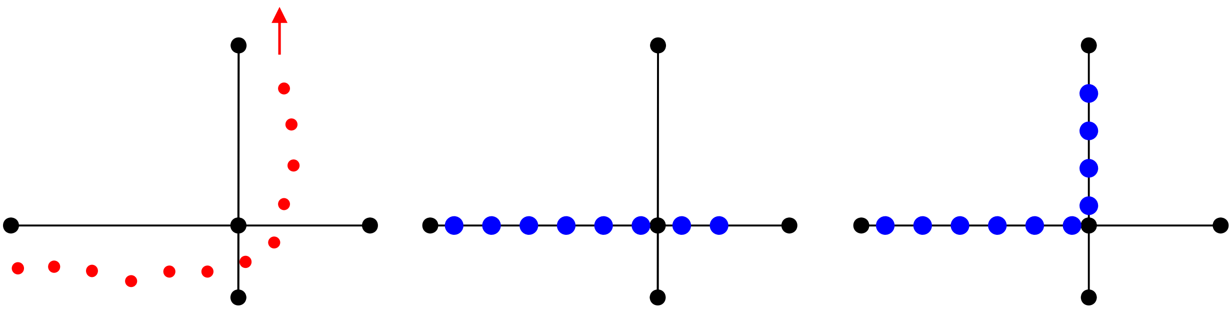
In fact it is possible that the point fix does not correspond to the physical position of the person. The reason is that we perform the prediction earlier then the user has arrived on the junction.

The explanation is given regarding the criterion in (22). 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 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 of precision is if the precision of localization is sufficient. Nowadays the efforts in the domain of pedestrian navigation are pointed to achieve a localization precision of 3 meters [Abwerzger et al. 2004] or even 1 meter [Usui et al. 2005].

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 25 Testing the robustness on the junctions

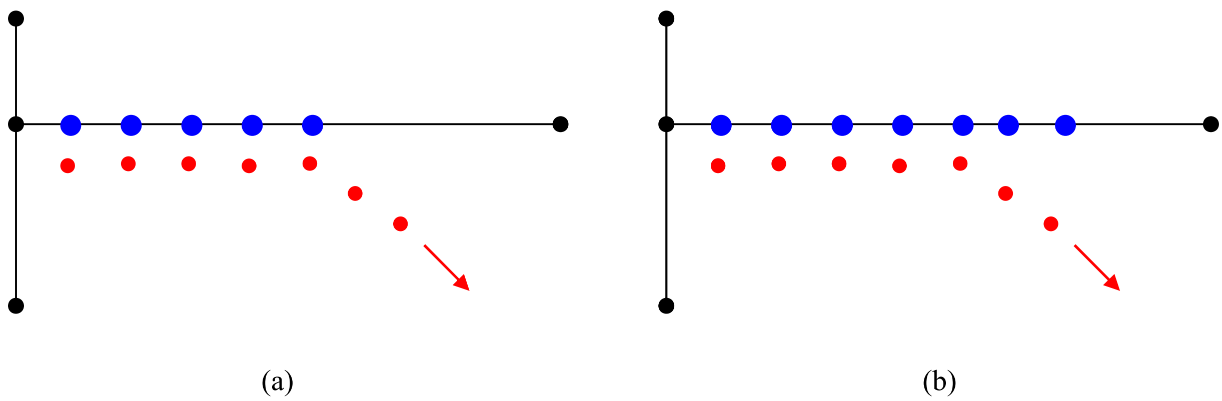


(a) (b) (c)

A part of trajectory passing through a crossroad is shown on figure 24a. Even if on that crossroad the direction of walk is changed the firs two point fixes after the junction are estimated on the edge in right direction (Fig. 24b). 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. 24c).

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].

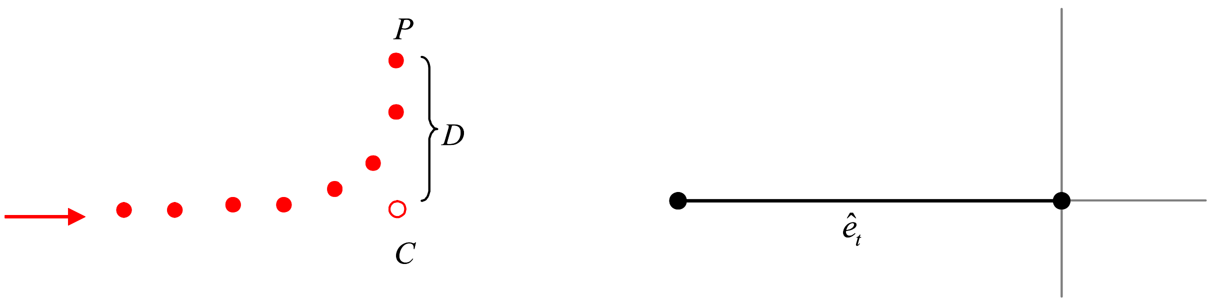
Figure 26 Testing the robustness in case of gross error in the heading measurement



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. 25a). 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. 25b).

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 27 Illustration on the activation of the continuous localization process



In fact, the location edge $\hat{e}_t$ is determined after the person has physically leaving it (Fig. 26). That determination depends on the moment of definition of the last critical point. 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:

$$
\begin{aligned}
D &= \left| CP \right| \\
\gamma_t^{(j)} &= A(\hat{e}_t) + \tau
\end{aligned} \tag{23}
$$

where j is the number of the candidate edges, $A(\hat{e}_t)$ is the azimuth of the location edge and τ is the accumulated angle from figure 4. The activation of the continuous localization starts at this moment using the defined values for D and γ. As candidate edges, $e^{(j)}$, the neighbours of the location edge are chosen.

Note that the process will not switch from initial localisation to continuous localisation, but both will continue working in parallel. Thus we will keep count on the history of walk which will assure a control for the location edge estimation.

5. Conclusions

The Bayesian inference is chosen in this research because it is very effective in the treatment of multimodal non-Gaussian distributions, which is the case in the problematic of the initial localization presented here. Before estimate 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 in certain moment t several edges can be defined as location edges (localization ambiguity on Fig.18). We need to keep count on all possible estimated edges before to localize the person.

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 is 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 autonomous technique for localization, the statistical approach allows the use of external information when it is available.

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.

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 performs normal walk and the trajectory is made on area covered by the map database (the graph). With normal walk the time needed to localize the person depends 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 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 most delicate part of the process is the implementation of the pre-processing procedure. The thresholds for detecting critical movements like turns and vertical movements must be precisely chosen. This phase is in direct connection with the performance of the navigation module (PNM) used in this research. It must be calibrated for the individual in function of his height and walking behaviour.

6. References

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.

Antonini, G., Bierlaire, M., Weber, M. Discrete choice models of pedestrian walking behavior, Transportation Research Part B.

Arulampalam, S., Maskell, S., Gordon, N., Clapp, T. (2001) A Tutorial on Particle Filters for On-line Non-linear/Non-Gaussian Bayesian Tracking, ©QinetiQ Ltd, ©DSTO

Bernstein, D. and Kornhauser, A. (1996) An Introduction to Map Matching for Personal Navigation Assistants**,** New Jersey TIDE Center, New Jersey Institute of Technology

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

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

Gilliéron P.-Y., Ladetto Q., Büchel D., Spassov I., Hagin C. and Pasquier M., Navigation pédestre: le futur en marche, Géomatique (revue du Québec), Vol. 31, Nr. 1, 2004.

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

Gustafsson, F., Gunnarsson, F., Bergman, N., Forssell, U., Jansson, J., Karlsson, R., Nordlund, P-J. (2002) Particle Filters for Positioning, Navigation and Tracking, IEEE Transactions on signal processing, vol. 50, No.2

Jong-Sun Pyo, Dong-Ho Shin, Tae-Kyung Sung (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

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

Lachapelle, G., Mezentsev, O., Collin, J., MacGougan, G. (2003) Pedestrian and Vehicular Navigation Under Signal Masking Using Integrated HSGPS and Self Contained Sensor Technologies, Department of Geomatics Engineering, University of Calgary, 2500 University Drive, NW, Calgary, Alberta, T2N 1N4, Canada

Ladetto Q., Seeters J. van, 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

Ladetto, Q. (2002) Capteurs et algorithmes pour la localisation autonome en mode pédestre**,** *PhD thesis No.2710*, Laboratoire de topométrie, EPFL, Suisse

Ladetto, Q., Gabaglio, V., Merminod, B. , Terrier, P., Schutz, Y. (2001) Human Walking Analysis Assisted by DGPS, Swiss Federal Institute of Technology, Lausanne, Switzerland

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

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

Philipona, C. (2002) Ne perdez pas le nord!, Camptocamp SA, Parc scientifique, Lausanne, Switzerland

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

Sumio USUI, Junichiro TSUJI, Koji WAKIMOTO, Satoshi TANAKA, Junshiro KANDA, Fumiaki SATO and Tadanori MIZUNO (2005), Evaluation of Positioning Accuracy for the Pedestrian Navigation System, IEICE Transactions on Communications 2005

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

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