Map-based Autonomous Personal Localisation and Tracking

(extended abstract)

Ivan Spassov, Michel Bierlaire, Bertrand Merminod

EPFL

- **Objective**

Localisation and tracking are the principal tasks of the navigation process. Nowadays there are many methods for localise and track a person. Methods like GPS- and WiFi-positioning that rely on the reception of external signals and methods like INS-positioning that use inertial measurements. Talking about autonomous localisation and tracking we consider a method that is independent from the reception of external data, from its precision and its availability. Following this idea in our approach we ignore the use of systems like GPS and WiFi and we focus on the use of inertial navigation system (INS) only. Naturally the INS is connected to a map database thus constituting a totally autonomous system. Our approach is addressed mainly to the problem of personal navigation in the city and indoors.

Consider a personal navigation system which contains inertial sensors that capture and measure different characteristics of the human's movement like speed and direction. Using these inertial measurements a relative position of the user is computed periodically thus representing user's trajectory by set of consecutive points. Each point is determined as a function of the previous point and parameters like distance and azimuth. The trajectory is defined in local coordinate system with origin in the first point.

The personal INS is connected to a map database defined by a link-node model, which represents the street network of some region as a graph. For indoor navigation, a similar link-node model is applied to represent the network of all corridors, passageways, elevators and staircases in the building. The graph is defined in an absolute coordinate system.

The problem to solve is to localize and track the person using the map database and inertial measurements of the personal INS. The idea behind our approach is to associate the user's trajectory to the contents of the map database by applying statistical methods in combination with map-matching.

- **Methodology**

First of all, we need to answer the question: what are the characteristics of a 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. The map database cannot be modified. Instead, the trajectory could be transformed from a set of consecutive points, to a polygon. That is possible thanks to a dedicated *motion model*.

The construction of the polygon is based on the detection of specific movements like turns, taking elevator or staircase, etc. With every step new values of the distance and the turn rate become available. The bearing of each step with respect to the previous one can be computed. The person can make a turn spread over several steps or a sharp change of direction in one step only. In both cases the total change of direction is computed and the position of a pivot point is determined. The detection of vertical movements is based mainly but not only on barometric measurements. Because of the big measurement error of the barometer additional information is needed. This information comes from the speed measurement. We use the phenomena that the user slows down when enter in the staircase and accelerates when leaving it. Respectively when entering the elevator the user stops and goes when leaving it. By detecting the points of change of speed and computing the altimetry difference between each pair of points the vertical movements are detected.

The modification of the user's trajectory with respect to the motion model transforms the upcoming set of inertial measurements to a set of polygonal parameters (distances and angles). We assume that the trajectory (the polygon) is a part of the graph.

Now we have two sources of data, the map database and the polygonal representation of the user's trajectory. The problem to solve is to associate the data from both sources, i.e. map-matching. That is to find the placement of the polygon in the contents of the map database. Once the matching is made the last edge of the polygon represents the user's location on the map. So we need to select from the map database the link associated to the last polygon edge. While the database has a finite number of elements, the polygon is updated with a distance and an angle periodically. Every time the polygon is updated an estimation of the user's location will be performed. The estimation relies on prior information (the trajectory, actual measurements and database) that could be used to compute a posterior estimation via the Bayesian inference.

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 can be calculated using prior information and actual measurements. Because of the non-linear nature of the estimation problem, non-linear filtering techniques like *particle filters* (Sequential Monte Carlo methods) are applied.

The walking person is considered as a dynamic system, whose trajectory is modified with respect to the motion model. The evolution of that dynamic system is defined by the following state space model:

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

$$y_t = h\left(x_t, x_{t-1}\right) + z_t \tag{1b}$$

with the following elements

x_t	state vector
u_t	motion input
y_t	measurement vector
z_t	measurement error
$h(x_t, x_{t-1})$	dimensions of x_t and x_{t-1} according to the database

The state vector x_t represents the location (the link) in moment t. The dynamic process is discretized regarding the motion model, so estimation is made every time the new measurements are available. The measurement vector $y_t = (l_t, \alpha_t , \gamma_t)^T$ includes the distance, horizontal angle and the vertical angle of movement detected by the motion model. The measurement noise e_t is assumed Gaussian. 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 measurements up to moment t. The problem to solve is to estimate x_t using the set of all available measurements Y_t .

From a 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. 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.

Considering the state space model and assumptions made, the filtering density is estimated:

$$p(x_t|Y_t) = \frac{p(y_t|Y_{t-1}, x_t)\, p(x_t|Y_{t-1})}{p(y_t|Y_{t-1})} \tag{2}$$

Here $p(x_t|Y_{t-1})$ is called *prior* of the state at moment t. It is obtained by using the state space model (1) and the Chapman-Kolmogorov equation:

$$p(x_t|Y_{t-1}) = \int p(x_t|x_{t-1})\, p(x_{t-1}|Y_{t-1}) dx_{t-1} \tag{3}$$

considering the assumption that the state follows a first order Markov process. The transition density $p(x_t|x_{t-1})$ is defined by the evolution of the dynamic system (1a).

In (2) the *likelihood function* $p(y_t|Y_{t-1},x_t)$, is defined by the measurement model and the known statistics of the measurement noise e_t . The *evidence* $p(y_t|Y_{t-1})$ has function of a normalizing constant. Thus $p(x_t|Y_t)$ can be computed recursively in two stages: *prediction* and *update*.

- Prediction

$$p(x_t|Y_{t-1}) = \int p(x_t|x_{t-1})\,p(x_{t-1}|Y_{t-1})\,dx_{t-1} \tag{4}$$

- Update

$$p(x_t|Y_t) = \frac{p(y_t|Y_{t-1},x_t)\,p(x_t|Y_{t-1})}{p(y_t|Y_{t-1})} \tag{5}$$

where $p(y_t|Y_{t-1}) = \int p(y_t|x_t)\,p(x_t|Y_{t-1})dx_t$

$p(x_t|Y_{t-1})$ is the prior from (2)

$p(y_t|Y_{t-1}, x_t)$ is the likelihood function

Often in sequential estimation algorithms, the measurements are assumed to be independent given the states:

$$p(y_t|x_t, A) = p(y_t|x_t) \tag{6}$$

In our case such assumption will not be reasonable because we consider that the trajectory made up to moment $t\text{-}1$ holds additional information, which is critical for the estimation.

This conceptual formulation of the problem, based on the Bayesian inference cannot be determined analytically. The solution can be achieved by applying Sequential Monte Carlo (SMC) methods also known as particle filters.

Particle filtering is defined as a sequential process for estimation of the states (parameters or hidden variables) of a system when new sets of observations become available. The principle of the SMC methods is to discretize a given density using a great number of samples also known as particles. This operation transforms the intractable integrals of the Bayesian solution into tractable discrete sums of weighted samples.

If we note the samples as $x_t^{(i)}$, and their weights as $w_t^{(i)}$, where $i=\{1,2,\ldots,N\}$, $N >> 0$ the posterior density $p(x_t|Y_t)$ could be approximated by (7) where $\delta(x_t - x_t^{(i)})$ is the Dirac delta measure

$$\hat{p}(x_t|Y_t) = \frac{\sum_{i=1}^{N} w_t^{(i)}\delta(x_t - x_t^{(i)})}{\sum_{i=1}^{N} w_t^{(i)}} \tag{7}$$

In our case $p(x_t|Y_t)$ is discretized considering each link as a particle with associated weight. The weight $w_t^{(i)}$ reflects the probability that the i^{th} particle represents the user's location at moment t.

The aim is to recursively compute $w_t^{(i)}$ applying prediction and update steps of the filtering process. On each iteration of the process prior data is used to evaluate the posterior density.

The posterior density $p(x_{t+1}|Y_{t+1})$ is calculated using (7). That is, the weights $w_{t+1}^{(i)}$ are updated according to

$$w_{t+1}^{(i)} = p(y_{t+1}|Y_t, x_{t+1}^{(i)})w_t^{(i)} \tag{8}$$

and normalized by:

$$\bar{w}_{t+1}^{(i)} = \frac{w_{t+1}^{(i)}}{\sum_{i=1}^{N} w_{t+1}^{(i)}} \tag{9}$$

The location is estimated by the sample with maximal weight, noted as x_t. Note that at moment t several samples could have maximal weight thus representing the location in different places on the map. In the next iterations the additional information on the trajectory will help to solve this ambiguity. With the convergence of the filter only one sample will have a maximal weight of 1.

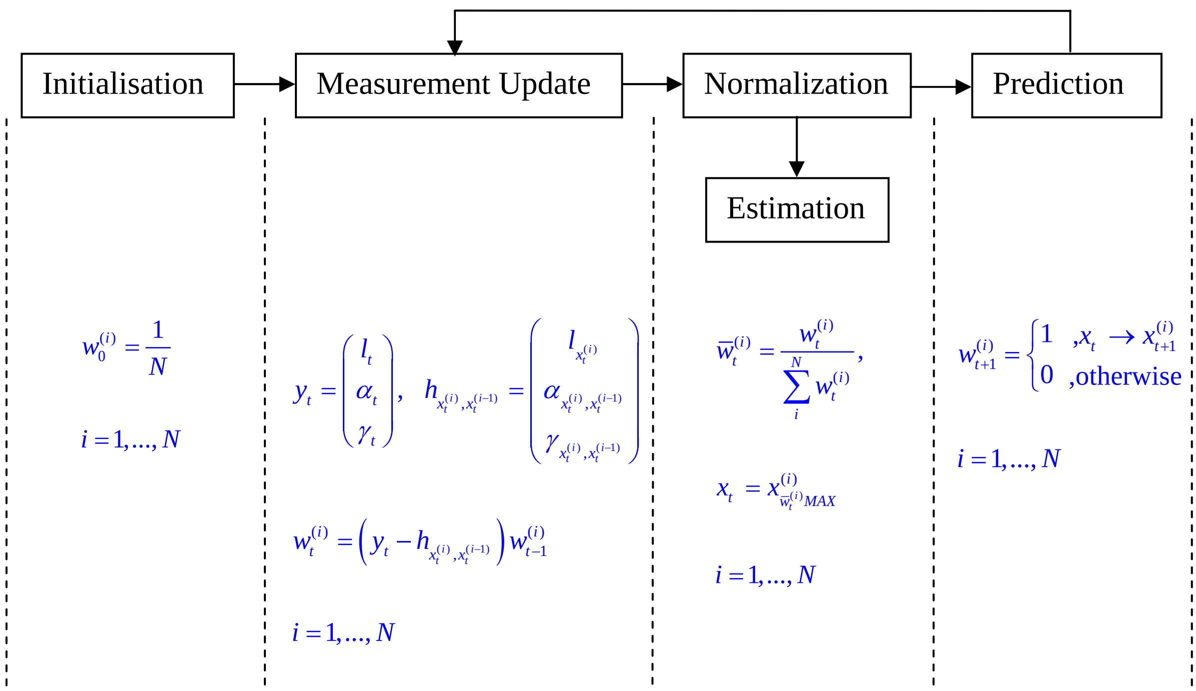

Figure 1: The particle filter implementation

The prediction step chooses a new set of samples by giving a weight 1 to the neighbour samples of x_t and a weight 0 to the rest of the samples. Then the algorithm turns to the update step (Figure 1). Once the user's location is found the link and direction of walk of the user are determined.

Next step in the development of the algorithm will be the activation of tracking mode. By observing simple geometric constrains every point of the trajectory will be associated to element of the map database. In parallel the localisation mode will stay active in order to keep the knowledge on the history of measurements.

- **Tests and contributions**

A large map database for the campus of the EPFL[1] has been created to support the functionalities such as shortest path computation and guidance. This database is used in our tests. As a navigation system we have used the Personal Navigation Module (PNM), developed by Vectronix AG[2]. The module is to be attached on the back side on the user's belt. The measurements are saved on a pocket PC. Both algorithms, the modification of the trajectory and the particle filter, are written in MS Visual Basic and run in post-treatment mode.

The common constraint is that the trajectory has to be performed on places covered by the map database. That is, the person must not leave the area represented by the link-node model. With a normal walk, the user's location is determined after several iterations of the algorithm. Since the localisation is referred to a link from the database its precision depends on the length of that link.

In tracking mode the precision will be increased by matching each point of the trajectory to the element of the map database. Thus a positioning with a sub-metric precision is expected. Besides the precise positioning the map-matching process will deliver corrections to the raw measurements that could be used for recalibrate the inertial sensors and to avoid the influence of their drifts.

The key idea in the Monte Carlo simulation is to discretize the posterior density by a large set of weighted samples. In our case the discretized posterior density $p(x_t|Y_t)$ is represented using all links of the database. Thus we work with the entire density involving all the samples into the computation at every moment. This is not an issue in post-processing, but a real-time implementation will impose restrictions.

Using inertial measurements only, the process of localisation is entirely autonomous and gives promising results. That method of localisation can be applied to many pedestrian navigation tasks. In particular, it suits the needs of fire-brigades and security services.

[1] http://plan.epfl.ch

[2] http://www.vectronix.ch