Indoor Localization Using Step and Turn Detection Together with Floor Map Information

Lukas Köping, Frank Ebner, Marcin Grzegorzek, Frank Deinzer
University of Applied Sciendes Würzburg-Schweinfurt,
Pattern Recognition Group, University of Siegen

In this work we present a method to estimate an indoor position with the help of smartphone sensors and without any knowledge of absolute positioning systems like Wi-Fi signals. Our system uses particle filtering to solve the recursive state estimation problem of finding the position of a pedestrian. We show how to integrate the information of the previous state into the weight update step and how the observation data can help within the state transition model. High positional accuracy can be achieved by only knowing that the pedestrian makes a foot step or changes her direction together with floor map information.

Categories and Subject Descriptors: Indoor Localization

Additional Key Words and Phrases: Particle Filter, Step Detection, Turn Detection

1 INTRODUCTION

Indoor localization considers the problem of finding the position of a person, robot or, more general, of a target object inside buildings. While Global Navigation Systems (GNSS) like GPS solved the localization problem in outdoor areas almost completely, this technique cannot be used indoor. This is because walls, windows, roofs and other obstacles attenuate the GNSS signals too heavily. However, a successful realization of an indoor localization system would pave the way for dozens of new applications. Imagine the possibility of guiding firemen through smoky buildings leading them directly to people in emergency situations. In commercial scenarios one can think of systems that guide customers through big shopping malls or museums to name a few.

A distinction must be made between indoor localization systems for pedestrians and indoor localization systems for robots. In the latter case the intention of movement is known to the system, e.g., if the robot wants to go left or right in the next time step. In pedestrian systems this is typically an unknown factor. Additionally, robots can be equipped with dozens of sensors, allowing to receive different forms of information of the environment like laser scans or video sequences and processing this information directly on an embedded high-performance computer. On the other side, pedestrian indoor localization systems can only rely on sensor data collected by devices like smartphones, since it is unlikely that people will agree to wear

Author‘s address: Lukas Köping, Frank Ebner; Fakultät für Informatik und Wirtschaftsinformatik, Hochschule für Angewandte Wissenschaften, Sanderheinrichsleitenweg 20, Würzburg, Deutschland

additional devices. This assumption might be softened in case of emergency teams for whom it is common to wear additional equipment, nevertheless, it would be asked too much from firemen to wear heavy laser scanners with them. Furthermore, the whole processing must be doable on a small device with the additional restriction of low energy consumption.

In this work we will introduce an approach that utilizes only sensor data of a smartphone. The process of indoor localization will be formulated as a state estimation problem in a dynamic, non-linear system. Since state estimations rarely can be calculated in an analytically closed form, we will use particle filtering as an approximation technique. For this, we will use floor maps as a priori information.

The paper is organized as follows. Section 2 gives a brief overview over indoor localization systems. Section 3 summarizes recursive state estimation and particle filtering. In addition we give a theoretical formulation for the integration of observation data into the state transition model and for the integration of old state information into the weight update step. Section 4 matches the recursive state estimation with indoor localization. In section 5 we show some experimental results. Finally, section 6 summarizes our work and illustrates future work.

2 RELATED WORK

Today's indoor positioning systems mostly make use of absolute positioning with Wi-Fi signals. In an on-line phase the received signal strengths (RSS) are measured at different locations and listed in a radio map. During the localization process the signal strengths at the current position of the target object are measured and compared against the radio map entries. For this, [Bagosi and Baruch 2011] evaluated different methods of matching the current fingerprint against the signal strength map. Among them, deterministic approaches like the Centroid, K nearest neighbour and weighted K nearest neighbour were compared against a probabilistic Gaussian method, showing the latter one outperforming all the other methods and working with an average error rate of 1 meter.

On the other side, dead reckoning systems [Woodman and Harle 2008; Link et al. 2011; Wang et al. 2012; Zhang et al. 2013] estimate the current position with respect to a starting or reference point. Inertial navigation for example relies on the information of gyroscopes, accelerometers and magnetometers to infer a position by using Newton's laws of motion. However, even small errors in measurement will sum up during integration at every time step and finally lead to heavily defective positional estimations [Köping et al. 2012]. It was shown that the integration of floor map information together with methods like map-matching can improve this approach [Davidson 2010]. Nevertheless, the positional estimation of dead reckoning systems must be corrected by external signals [Woodman and Harle 2008]. A popular approach in this context is to combine Wi-Fi localization and readings from inertial sensors. A major drawback of this method lies in the development of a signal strength map - a process that has to be repeated periodically to incorporate environmental changes.

In the last few years the use of smartphones continuously rose because they provide most of the necessary sensors like accelerometers, gyroscopes, magnetometers, Wi-Fi modules, Bluetooth and cameras. Additionally, smartphones constitute a practical way for people to navigate through buildings, since they do not have to wear additional, unhandy devices like laptops or laser scanners with them. [Link et al. 2011] uses step detection and path matching to infer a position. A step is detected by utilizing specific patterns in accelerometer data. The detected step and its heading are matched against an expected route, which is a path, suggested by the system. It describes the shortest distance between the user's current position and its desired destination. For this, they propose two different path matching algorithms that utilize dynamic programming techniques and incorporate detected steps, the heading and map information. [Zhang et al. 2013] uses a modified Kalman filter to achieve accurate orientation. During localization also step detection as well as step length detection is performed. In contrast to [Link et al. 2011], where step length is derived from a person's height, [Zhang et al. 2013]'s system estimates step length with the help of accelerometer data. In [Wang et al. 2012] the dead-reckoning system is stabilized by so called internal landmarks. These are special locations on a map, where characteristic signals can be measured. For example elevators and stairs reveal unique signal curves in accelerometer data. Given that such a characteristic signal is measured, it is possible to reset the defective position estimation and use this location as new starting point for dead-reckoning position.

3 RECURSIVE STATE ESTIMATION

Recursive state estimation is a technique to infer a hidden state $\mathbf{q}_t$ at a given time t with the help of observations $\mathbf{o}_t$. Typically this technique is often found in the field of robotics, where different states like, e.g., the robot's position must be estimated given some sensor measurements [Thrun et al. 2006]. Thereby, $\mathbf{q}_t$ contains all relevant information to describe the present system and the set of all observations up to time t can be written as $\langle\mathbf{o}\rangle_t = \{\mathbf{o}_{t'} \mid 0 \leq t' \leq t\}$. Often enough the goal is to estimate a most probable state $\hat{\mathbf{q}}_t$ at time t:

$$\hat{\mathbf{q}}_t = \arg\max_{\mathbf{q}_t} p(\mathbf{q}_t|\langle\mathbf{o}\rangle_t) \tag{1}$$

Given Bayes rules, $p(\mathbf{q}_t|\langle\mathbf{o}\rangle_t)$ can be written as

$$p(\mathbf{q}_t|\langle\mathbf{o}\rangle_t) = \frac{p(\langle\mathbf{o}\rangle_t|\mathbf{q}_t)p(\mathbf{q}_t)}{p(\langle\mathbf{o}\rangle_t)} \tag{2}$$

It describes the probability density of a state at time t given a series of observations. A convenient way to rewrite (2) is given by [Deinzer 2005]:

$$p(\mathbf{q}_t|\langle\mathbf{o}\rangle_t) = p(\mathbf{o}_t|\mathbf{q}_t) \int p(\mathbf{q}_t|\mathbf{q}_{t-1})p(\mathbf{q}_{t-1}|\langle\mathbf{o}\rangle_{t-1})d\mathbf{q}_{t-1} \tag{3}$$

Three essential parts can be identified in (3). The likelihood of an observation while in a given state is denoted as $p(\mathbf{o}_t|\mathbf{q}_t)$, the state transition probability as $p(\mathbf{q}_t|\mathbf{q}_{t-1})$ and $p(\mathbf{q}_{t-1}|\langle\mathbf{o}\rangle_{t-1})$ contains the information of all previous time steps.

In our approach we deviate from the standard form given by (3) and integrate observation data into the state transition model: The information of the pedestrian making a foot step or changing its direction will be used to estimate a new state:

$$p(\mathbf{q}_t|\mathbf{q}_{t-1}, \mathbf{o}_{t-1}) \tag{4}$$

In addition to that, state information of the previous time step $\mathbf{q}_{t-1}$ is added to $p(\mathbf{o}_t|\mathbf{q}_t)$. For this, the position of the last time step is used to evaluate the current position. In section 4 this approach is described in more detail. Formally, the integration of the last time step's information leads to:

$$p(\mathbf{o}_t|\mathbf{q}_t,\mathbf{q}_{t-1}) \tag{5}$$

Subsequently we will show how to integrate (4) and (5) into (3). Our state at time t does not only depend on the current observation but also on the previous state, so that we are looking for $p(\mathbf{q}_t|\langle\mathbf{o}\rangle_t, \mathbf{q}_{t-1})$:

$$\begin{aligned}
p(\mathbf{q}_t|\langle\mathbf{o}\rangle_t,\mathbf{q}_{t-1}) &= \frac{p(\mathbf{q}_t,\langle\mathbf{o}\rangle_t,\mathbf{q}_{t-1})}{p(\mathbf{q}_{t-1},\langle\mathbf{o}\rangle_t)} && (6)\\
&= \frac{p(\mathbf{q}_t,\mathbf{o}_t,\langle\mathbf{o}\rangle_{t-1},\mathbf{q}_{t-1})}{p(\mathbf{q}_{t-1},\langle\mathbf{o}\rangle_{t-1},\mathbf{o}_t)} && (7)\\
&= \frac{p(\langle\mathbf{o}\rangle_{t-1})p(\mathbf{q}_{t-1}|\langle\mathbf{o}\rangle_{t-1})p(\mathbf{q}_t|\langle\mathbf{o}\rangle_{t-1},\mathbf{q}_{t-1})p(\mathbf{o}_t|\mathbf{q}_t,\langle\mathbf{o}\rangle_{t-1},\mathbf{q}_{t-1})}{p(\langle\mathbf{o}\rangle_{t-1})p(\mathbf{q}_{t-1}|\langle\mathbf{o}\rangle_{t-1})p(\mathbf{o}_t|\mathbf{q}_{t-1},\langle\mathbf{o}\rangle_{t-1})} && (8)\\
&= \frac{p(\mathbf{q}_t|\langle\mathbf{o}\rangle_{t-1},\mathbf{q}_{t-1})p(\mathbf{o}_t|\mathbf{q}_t,\langle\mathbf{o}\rangle_{t-1},\mathbf{q}_{t-1})}{p(\mathbf{o}_t|\langle\mathbf{o}\rangle_{t-1},\mathbf{q}_{t-1})} && (9)\\
&= \frac{1}{k}p(\mathbf{q}_t|\langle\mathbf{o}\rangle_{t-1},\mathbf{q}_{t-1})p(\mathbf{o}_t|\mathbf{q}_t,\langle\mathbf{o}\rangle_{t-1},\mathbf{q}_{t-1}) && (10)\\
&= \frac{1}{k}p(\mathbf{q}_t|\langle\mathbf{o}\rangle_{t-1},\mathbf{q}_{t-1})p(\mathbf{o}_t|\mathbf{q}_t,\mathbf{q}_{t-1}) && (11)\\
&= \frac{1}{k}p(\mathbf{o}_t|\mathbf{q}_t,\mathbf{q}_{t-1})\int p(\mathbf{q}_t|\mathbf{q}_{t-1},\langle\mathbf{o}\rangle_{t-1})p(\mathbf{q}_{t-1}|\langle\mathbf{o}\rangle_{t-1})d\mathbf{q}_{t-1} && (12)\\
&= \frac{1}{k}p(\mathbf{o}_t|\mathbf{q}_t,\mathbf{q}_{t-1})\int p(\mathbf{q}_t|\mathbf{q}_{t-1},\mathbf{o}_{t-1})p(\mathbf{q}_{t-1}|\langle\mathbf{o}\rangle_{t-1})d\mathbf{q}_{t-1} && (13)
\end{aligned}$$

For (6) the definition of the conditional probability is used, while (8) is the result of applying the Multiplication Theorem of Probabilities. (8) can than be reduced to (9), whereby the denominator is a constant and denoted as k in (10). Using Markov's assumption, $p(\mathbf{o}_t|\mathbf{q}_t, \langle\mathbf{o}\rangle_{t-1}, \mathbf{q}_{t-1})$ is rewritten as $p(\mathbf{o}_t|\mathbf{q}_t, \mathbf{q}_{t-1})$. This is true since the probability of a current observation is independent of previous observations. Applying the Total Law of Probability for $p(\mathbf{q}_t|\mathbf{q}_{t-1}, \langle\mathbf{o}\rangle_{t-1})$ finally results in $p(\mathbf{q}_t|\mathbf{q}_{t-1}, \mathbf{o}_{t-1})$.

Since an analytical solution for densities of the form of (13) exists only in rare cases, a common approach to approximate such densities is particle filtering [Arulampalam et al. 2002; Gordon et al. 1993; Isard and Blake 1998; Doucet and Johansen 2011]. A particle $\pi_t^i, i \in 1,\dots,N$ represents a potential state at time t, whereby N different particles are used. Every particle is defined as tuple

$$\pi_t^i = \left(\mathbf{q}_t^i, w_t^i\right). \tag{14}$$

$\mathbf{q}_t^i$ is the represented state of particle π_t^i, w_t^i is the weight of the particle telling about its importance. Additionally, weights are normalized so that

$$\sum_{i=1}^{N} w_t^i = 1\,. \tag{15}$$

The posterior density is then equivalent to

$$p(\mathbf{q}_t|\langle\mathbf{o}\rangle_t) \approx \sum_{i=1}^{N} w_t^i\ ,\delta\left(\mathbf{q}_t - \mathbf{q}_t^i\right), \tag{16}$$

where $\delta(\cdot)$ is Dirac's delta function and $N \to \infty$ holds true. Practically, in the particle filter at every time step new particles are drawn according to

$$\pi_t^i \propto p(\mathbf{q}_t | \mathbf{q}_{t-1}, \mathbf{o}_{t-1}), \tag{17}$$

which is called the state transition step. In the evaluation step every particle is then weighted with

$$w_t^i = p(\mathbf{o}_t | \mathbf{q}_t, \mathbf{q}_{t-1}). \tag{18}$$

Finally, in a resampling step a new particle generation for the next time step t+1 is generated by drawing N new particles according to

$$\pi_t^i \propto w_t^i \tag{19}$$

4 RECURSIVE STATE ESTIMATION IN INDOOR LOCALIZATION

Our approach uses recursive state estimation with particle filtering as realization to infer the position of a pedestrian at a given time t with the help of different sensor data collected by a smart phone. If particle filtering is used, one must be aware of the state's dimension, since it has a vast impact on the number of samples to be simulated. For our approach it is enough to incorporate the x- and y-coordinate of the searched position as information, so that

$$\pi_t^i \propto w_t^i \tag{20}$$

The incoming sensor data at time t is treated as observation $\mathbf{o}_t$, thereby $\mathbf{o}_t$ consists of

$$\mathbf{o}_t = \begin{pmatrix} \text{step}_t \\ \text{turn}_t \\ \alpha_t \end{pmatrix}, \tag{21}$$

where step_t and turn_t are both boolean variables and step_t = true, if it was detected that the target object made a footstep in the last time step and turn_t=true, if the target object made a left or right turn in the last time step. If a turn was detected the direction of the change can be determined by the peak value in the turn detection process (see chapter 4.1). α_t denotes the heading information. The heading information is initialized at the beginning of the position tracking at a known starting position. We assume to know the initial direction of the pedestrian's body regarding the floor map. This direction usually can be estimated due to the context, e.g., if the pedestrian walks upstairs, her body direction is known at the end of the stairway and can be used as heading information. Another possibility initializing the heading information is to use the orientation data at a given time step, which today is easily available on every smartphone.

4.1 Step and Turn Detection

Step detection: We use a step detection process to estimate if our target object made a move in the last time interval, whereat the method we use is described in detail in [Link et al. 2011]. Basically, the linear acceleration towards the ground is measured. In this signal a step is detected by picking a data point, which exceeds a certain threshold. In addition this data point must be greater than one of its following data points by a certain amount, whereby the following data point must lie within a given time interval. After a certain dead time, in which no step can be recognized, the step detection process starts from the beginning (see Fig.1).

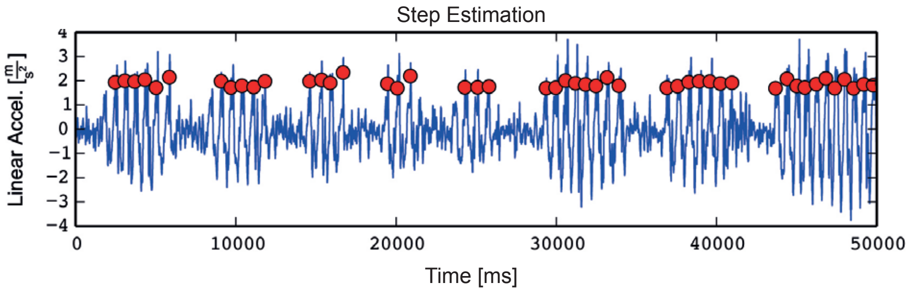

Fig. 1. Step detection process: The blue curve represents the linear acceleration in ground direction. Red dots represent detected steps.

Turn detection: In addition to the step detection process, turn detection is integrated into our system. Based on the gyroscope data of the smartphone it is possible to recognize if the target object made a left or right turn or is still walking straight. Although the smartphone orientation data offers similar information, this data cannot be reliably used for pedestrian localization systems. This is because the geomagnetic sensors suffer heavily from small influences outgoing from the building‘s characteristic or parts of the building, e.g., metal doors or metal railing, not to mention computers or monitors. The gyroscope on the other side does not suffer from such influences. This is why a left or right turn can easily be detected in the data signal of the smartphone‘s yaw angle. For this, we can use a similar approach like in the step detection process:

We denote g_t as a data point of the gyroscope at the measurement time t, which tells how fast the phone is turning around a given axis and is given in $\frac{\text{rad}}{\text{sec}}$. We are then looking for a measurement g_t, which is bigger by a threshold Δg than one of the measurements in the time interval $(t; t + \Delta t]$. If a measurement g_t was found, a new turn can only be detected after a time t_{noTurn}. To distinguish between left and right turns, we can use the sign of g_t. If g_t is positive a left turn is detected, otherwise a right turn is detected. We have achieved good results using $= 1.0\frac{\text{rad}}{\text{sec}}$, Δt = 300ms and t_{noTurn} = 2500ms. Furthermore, we assume detected turns to have an angle of 90°. While this might not be realistic in most cases, such a rough estimation is enough as input for the recursive state estimation. If a turn was detected, the global heading α_t in $\mathbf{o}_t$ is updated accordingly by adding or subtracting 90°.

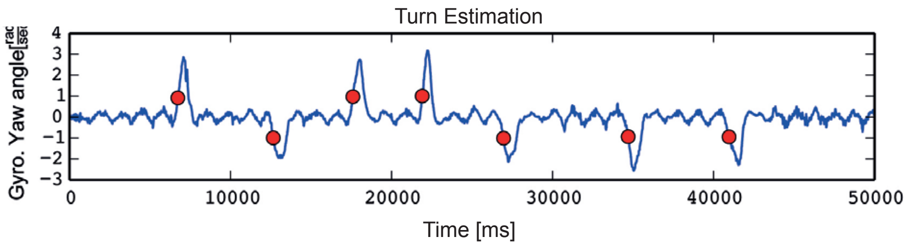

Fig. 2. The estimation of turns is shown using the gyroscope signal. Positive peaks are left turns, while negative peaks are right turns.

4.2 State Transition and Weight Update

State transition: The state transition simulates a possible movement of a pedestrian during two time steps. A crucial information that is integrated into this model is the knowledge taken from floor maps. Utilizing this information, the movement estimation can be restricted due to walls, doors or other obstacles. In addition to that, we assume pedestrians to move with a typical walking speed, which results in an average step length. The step detection process is integrated into the state transition model such that we differentiate between two cases when drawing a new particle:

$$\pi_t^i \propto p(\mathbf{q}_t|\mathbf{q}_{t-1}, \mathbf{o}_{t-1}) = \begin{cases} N_{\text{step}}(d_t|\mu_{\text{step}}, \sigma_{\text{step}}) & \text{if step}_t = \text{true} \\ N_{\text{noStep}}(d_t|\mu_{\text{noStep}}, \sigma_{\text{noStep}}) & \text{otherwise} \end{cases} \quad (22)$$

where μ_{step} denotes the average step length of a pedestrian and is assumed to be 70 cm, σ_{step} is the standard deviation and assumed to be 50cm, μ_{noStep} to be the mean value when no step was detected and σ_{noStep} to be the standard deviation in this case, and assumed to be 0 cm and 10 cm respectively. d_t is the walking distance between two time steps.

One must be aware that drawing from both distributions also takes floor map information into account. This means that particles cannot cross walls and, therefore, all distance measurements from one point to another also must incorporate this restriction [Köping et al. 2012]. In the phase of the state prediction we do not rely on any heading information, which leads the particles to spread in every possible direction.

Weight update: Information of direction is integrated in the weight update step. At every time step t a particle is evaluated as follows: Based upon the position x_{t-1} and y_{t-1} of the previous state q_{t-1}, a position $x_{t,\text{exp}}^i, y_{t,\text{exp}}^i$ for each particle i is computed. This is the position for the new state at time t, if it followed a deterministic model. Essentially, this expected position is calculated with

$$x_{t,\text{exp}}^i = x_{t-1}^i + l' \cdot \cos(\alpha_t + \Delta\alpha_t) \quad (23)$$

$$y_{t,\text{exp}}^i = y_{t-1}^i + l' \cdot \sin(\alpha_t + \Delta\alpha_t) \quad (24)$$

where l‘ is the step length. Its value depends on whether a footstep was detected or not and, therefore, its value is either 0 cm or 70 cm. α_t is the global heading information contained in $\mathbf{o}_t$ and $\Delta\alpha_t$ is the change of heading. The value of $\Delta\alpha_t$ is either ±90°, if a left or right turn was detected in the last time step, or 0°, if no turn was detected. Finally, we can evaluate each state using a bivariate Gaussian

$$N(\mathbf{q}_t| (x_{t,\text{exp}}^i, y_{t,\text{exp}}^i)^T, \Sigma) \quad (25)$$

where Σ represents the covariance matrix and its values are estimated empirically with $\mu_x = x_{t,\text{exp}}^i, \mu_y = y_{t,\text{exp}}^i, \sigma_x = \sigma_y = 2.0$ and correlation of 0.0.

5 EXPERIMENTS

We tested our method with two different walking paths (see Fig. 3), the first one with a total distance of 23m and a walking time of about 36 seconds, the second with a total distance of 42m and a walking time of about 55 seconds. The sequences contain also periods, where the target object stands still to prove that our method is also capable of dealing with no movement. We walked both sequences 10 times each, to compensate for measurement variance of the sensors. All sensor data was collected with a Samsung Galaxy SII smartphone while the latter was laying flat on the hand during walking. After recording the data the particle filter computation was done on a Core i-5 2520M CPU@2.50GHz and 4.00GB RAM. All particles were updated every 150ms with a total number of 10,000 particles. Since the calculation for one whole run only took few seconds, we expect that an implementation of our system on an up-to-date smartphone can calculate a position estimation in real time.

For estimation of the total error we evaluated the distance between the real position, where the walk ended, and the estimated position at the end of the simulation run. On average our system produces an error of about 80cm for walk 1 and 92cm for walk 2. The worst position estimation had an error of about 1.4m for walk 1 and 1.9m for walk 2, the best estimation had an error of 35cm and 30cm for walk 1 and walk 2, respectively. The ground truth data was generated by walking from one turning point to the next turning point in a straight line.

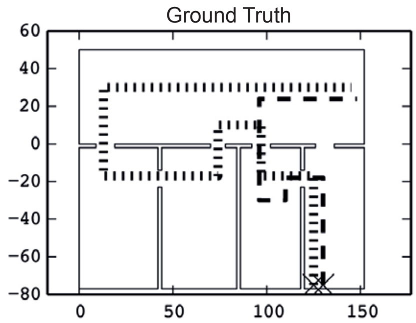

Fig. 3. Ground truth data for walk1 and walk2

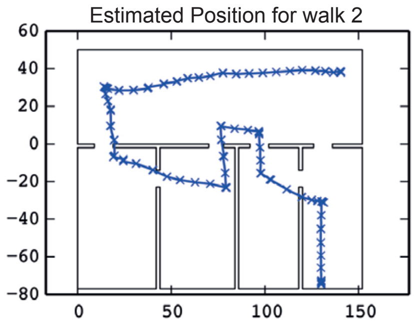

Fig. 4. Exemplary walking path for walk2

One source of error is the duration of the walk. The errors of wrongly estimated steps and step length sum up after some time. This was especially evident during a walk in a circle in the corridor, where we could not rely on map information too much. Nevertheless it should be mentioned that the main reason for the positional error is not necessarily the duration of a walk, but the missing information of walls and obstacles while in the long corridor. This is because the information of walls and doors often is enough to overcome the detection of too many or too little foot steps or a natural variation of step length. In the lower right of Figure 4 this principle can be shown. Clearly, either not enough steps are detected or the step length was estimated incorrectly. Without map information, after the first left turn, the estimated route would be a horizontal line to the left. Instead the position gets corrected to a position a few meters northwards.

Figure 4 shows an exemplary positioning estimation. It can be seen that the positional estimation corresponds with the real walking sequence, but the visualization of the estimated path crosses a wall. At the beginning some steps are detected. After a detected left turn and some newly detected steps particles try to cross the wall, which is not possible because of the map information. Nevertheless, because of the distribution of the particles over the map in the transition model some particles have the possibility to go far enough and to walk through the door. The new estimated position correctly is left of the wall, but simply connecting the estimated positions leads to a visualization of the estimated position, which suggests that the pedestrian walks through the wall. Such problems may be overcome if an appropriate smoothing algorithm will be applied [Doucet and Johansen 2011].

One major drawback of our system is the possibility that a left or right turn is not or wrongly detected. This leads to massive errors in the system because the state evaluation assigns a low weight to particles, which do not follow the expected route. In consequence these particles will most likely disappear after resampling.

6 CONCLUSION AND FUTURE WORK

In this paper we presented an indoor navigation system based upon the detection of steps and turns as well as on floor map information. To improve the weight update step of the particle filter we showed that it is possible to weaken the Markov assumption and explicitly integrate the information of a previous step. We showed how observation data can be used within the state transition model. To improve our system for the future, we plan to extend our current turn detection. At this time we are not capable to give a good estimation of the angle the pedestrian turned, e.g., by now we cannot detect if someone made a 180° turn. Given such information would significantly improve the current system. In addition to that we will also integrate a step length detection and enrich our system with new sensors like Wi-Fi and cameras. The additional sensors will stabilize the system in situations where a step and turn detection is only hard to achive like in escalators and elevators.

REFERENCES

BAGOSI, T., BARUCH, Z. 2011. Indoor Localization by WiFi. IEEE International Conference on Intelligent Computer Communication and Processing (ICCP) (August 2011) 449-452

WOODMAN, O., HARLE, R. 2008. Pedestrian Localisation for Indoor Environments. Proceedings of the 10th International Conference on Ubiquitous computing (September 2008) 114-123

LINK, J.A.B., SMITH, P., WEHRLE, K. 2011. FootPath: Accurate map-based indoor navigation using smartphones. International Conference on Indoor Positioning and Indoor Navigation (IPIN) (September 2011) 1-8

WANG, H., SEN, S., ELGOHARY, A., FARID, M., YOUSSEF, M., CHOUDHURY, R.R. 2012. No need to war-drive: unsupervised indoor localization. Proceedings of the 10th international conference on Mobile systems, applications, and services (2012) 197-210

ZHANG, R., BANNOURA, A., HÖFLINGER, F. 2013. Indoor localization using a smart phone. IEEE Sensors Applications Symposium (SAS) (February 2013) 38-42

KÖPING, L., MÜHSAM, T., OFENBERG, C., CZECH, B., BERNARD, M., SCHMER, J., DEINZER, F. 2012. Indoor Navigation Using Particle Filter and Sensor Fusion. Annual of Navigation 19(2) (December 2012) 31-40

DAVIDSON, P., COLLIN, J., TAKALA, J. 2010. Application of particle filters for indoor positioning using floor plans. Ubiquitous Positioning Indoor Navigation and Location Based Service (UPINLBS) (October 2010) 1-4

THRUN, S., BURGARD, W., FOX, D. 2006. Probabilistic Robotics. MIT Press

DEINZER, F. 2005. Optimale Ansichtenauswahl in der aktiven Objekterkennung. Volume 20 of Studien zur Mustererkennung.

ARULAMPALAM, M.S., MASKELL, S., GORDON, N., CLAPP, T. 2002. A Tutorial on Particle Filters for Online Nonlinear/Non-Gaussian Bayesian Tracking. IEEE Transactions on Signal Processing 50(2) (February 2002) 174-188

GORDON, N.J., SALMOND, D.J., SMITH, A.F.M. 1993. Novel approach to nonlinear/non-Gaussian Bayesian state estimation. IEE Proceedings F Radar and Signal Processing 140(2) (April 1993) 107-113

ISARD, M., BLAKE, A. 1998. Condensation - conditional density propagation for visual tracking. International Journal of Computer Vision 29(1) (August 1998) 5-28