Continuous location and direction estimation with multiple sensors using particle filtering

Kai Wendlandt, Mohammed Khider, Michael Angermann, and Patrick Robertson

***Abstract*— In this paper we discuss the use of Particle Filtering to estimate the values of several state variables describing a user's context. Since Particle Filtering algorithms are computationally efficient realizations of Bayesian Filters they perform exceptionally well to optimally combine the a priori knowledge stemming from behavioral models, such as movement models, and the noisy measurements from sensors. The estimate at each time step is obtained in the form of a probability density function that represents the entire information and quantifies the inherent uncertainty about the context.**
The concept has been realized in simulations and experiments. In this paper, the applied movement model is presented with simulated measurements from GPS and compass sensors to illustrate the concept.

I. INTRODUCTION

Navigation in outdoor areas with satellite navigation systems (GPS) is widespread and well established. Many applications for location based services rely on this infrastructure and with the European Navigation System GALILEO, an additional system will be available in the future. All these mostly mobile services obtain the users location by using one out of several positioning technologies. With location and other additional information, personalized applications and services such as restaurant finder can be provided.

One limitation of the technology used by GPS is the need for line of sight from the receiver to at least four satellites. In urban canyons, a user can receive misleading positioning data due to multipath propagation of the signal, which makes it difficult for the device to accurately calculate it's position from only the satellite range measurements.

Finally in indoor locations, the receiver sees a weak and multipath afflicted signal. A position measurement that only relies on GPS will then lead to a less accurate position estimation.

Research projects [1] and many commercial applications use systems with inertial measurement units (IMU) to provide location information during unavailability of satellites. In such applications, the fusion of the sensor outputs is commonly calculated using (Extended) Kalman Filtering which is a form of Bayesian Filtering.

Particle Filtering is a more general Bayesian Filter algorithm which has the advantage of not being subject to any linearity or Gaussianity constraints in the models (see section III). Particle filtering is therefore an appropriate method for tracking a pedestrian.

All authors are with German Aerospace Center, Institute of Communication and Navigation, 82234 Wessling, Germany `<name>.<lastname>@dlr.de`

Using these algorithms and other previous work, namely the SoftLocation concept [2], we developed a system to simulate and measure a pedestrians position and his context. This system and its performance will be presented and analyzed in the following sections.

A description of the overall system is given in section II and detailed information about the sensors is presented in section IV.

A suitable movement model for pedestrians in various situations is needed to enable the system to estimate a more accurate position from the noisy measurements. Our developed movement model is described in section V. Thereafter, simulation results with our framework are presented (section VI).

A final outlook to the potentials of such a system, that can estimate not only the position but also the context of a user is given at the end of this paper in section VII.

II. SYSTEM OVERVIEW

We developed a distributed demonstration and simulation environment for mobility, location and context applications. This system aims at comparing different algorithms for position estimation and shows the tracking of a pedestrian equipped with various sensors. Its components and the overall structure (fig. 1) are presented in this section.

The main parts are:

- *Hardware Sensor Platform* (blue, I): It is used to physically connect sensors - often via serial interfaces - and to convert the data output to a common format.
- *Location Server* (red, II): It handles all location objects and their parameters.
- *Fusion Engine* (green, III): It implements the data fusion algorithms and filtering methods.
- *Visualization Server* (yellow, IV): It analyzes and visualizes all the objects (data, maps and particles) that need to be displayed to the user. It also provides a control panel for the manipulation of parameters and the selection of visualized items.

The data can be fused locally on the sensor platform. Alternatively it can be sent to a server in real time - using any available network - for further processing. In addition, all the sensors can be replaced by simulated objects for statistical analysis of certain parameters.

A. Software details

Our implementation of the core system and the sensor platform consists of a simulation system and a live system.

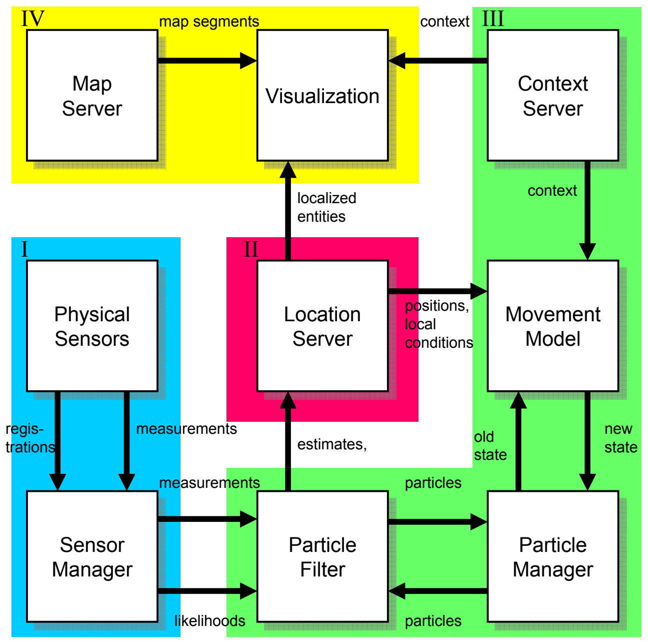

Fig. 1. Information flow between the different system components

It is entirely implemented in Java (J2SE v1.4) to achieve a platform independent tool.

In the current experimental setup, the pedestrian carries a notebook mounted on a backpack. The location sensors can be attached to this computer. A mobile data link transfers the sensor data over an IP network to a server side, where the measurements are processed and visualized.

Using the live system, all available sensors (see subsection IV) provide the software with real measured data (e.g. NMEA data from GPS receivers). This data is converted into an internal coordinate system and passed to the location server for further processing.

The server can also be provided with simulated position and direction measurements, where measurement errors due to noisy sensor data can be easily adjusted for performance analysis.

III. BACKGROUND PARTICLE FILTERING

Particle Filters represent probability densities of the states x_k by a set of random samples (Monte Carlo Sampling). The samples (=particles) are distributed according to these PDFs. Each of these particles has an associated weight. The samples with higher values of the continuous PDF will have more particles and these particles will have higher weights [3].

The posterior is:

$$p(x_k|z_{1:k}) \approx \sum_{i=1}^{N_p} w_k^i \delta(x_k - x_k^i) \quad (1)$$

where: N_p is the number of particles (samples), $\{x_k^i, i = 0, \ldots, N_s\}$ is the set of sampled states, and $\{w_k^i, i = 0, \ldots, N_s\}$ is the weight of the sampled state.

For large numbers of samples the random (Monte Carlo) sampling approximation converges to the usual functional description of the posterior PDF, and the particle filter approaches the optimal Bayesian estimate.

Particle Filtering for state estimation also relies on the general dynamic system model with

- a *movement model* which describes the transitions between the different states of the particles
- and a *measurement model*, where all the characteristics of the sensors are described.

These two models are applied sequentially in a *prediction step* and in an *update step*. They form a first order Hidden Markov Model for the estimation of the states. The general Bayesian description for this prediction and update step is shown by equations (2) and (3). We start with:

$$p(x_k|z_{1:k-1}) = \int p(x_k|x_{k-1})p(x_{k-1}|z_{1:k-1})dx_{k-1} \quad (2)$$

where: $p(x_k|x_{k-1})$ is the conditional PDF of any state x_k at time k given the previous state x_{k-1}, and $p(x_{k-1}|z_{1:k-1})$ is the PDF of the previous state x_{k-1} given the history of the measurements $z_{1:k-1}$ up to time $k-1$. The PDF $p(x_{k-1}|z_{1:k-1})$ is assumed to be available at time $k-1$. The posterior becomes:

$$p(x_k|z_{1:k}) = \alpha_k p(z_k|x_k)p(x_k|z_{1:k-1}) \quad (3)$$

where: $p(z_k|x_k)$ is the conditional PDF of any measurement z_k at time k given the state x_k at the same time. It is also called the likelihood, and $p(x_k|z_{1:k-1})$ is the prior PDF of the state x_k at time k given the history of measurements $z_{1:k-1}$ up to time $k-1$, and α_k is the normalization constant $\frac{1}{p(z_k|z_{1:k-1})}$.

Monte Carlo (MC) methods have the advantage of not being subject to any linearity or Gaussianity constraints in the model, and they also have appealing convergence properties.

In our work, we use *SIR Particle Filter* (sampling importance resampling) as presented in [4].

IV. SENSORS AND ALGORITHMS

As mentioned earlier in section II, our software framework can be easily extended by new sensors. These sensors are connected to a specific hardware interface from which a `Sensor` class extracts and provides the measurement data. A `Measurement` class collects the measurements from the `Sensor` class and prepares them for the `Likelihood Function` class where the likelihood functions $p(z_k|x_k)$ are generated. They are needed according to equation (3) in the update stage of particle filtering.

To describe the possible errors of the sensors, Gaussian noise is added to the states in the measurement model of each sensor. These parameters mean (μ) and sigma (σ) have to be analyzed and determined individually for each sensor.

Currently, our system incorporates an electronic compass and a GPS receiver as sources for location and direction. Other sensors like range measurement equipment, proximity sensors, street maps and floor plans can easily be integrated due to a software structure with a common interface (API).

A. GPS

The GPS receiver (RoyalTek Bluetooth receiver RBT 1000) provides coordinates in the WGS84 Format (World Geodetic System 1984) to the host computer via the NMEA-Protocol.

As GPS devices provide information about the geometric strength of the satellite configuration (DOP - Dilution of Precision), these values are extracted for further processing. We use the DOP values to model the Gaussian GPS positioning error in the measurement model.

B. Compass

A Palm Navigator electronic compass from Precision Navigation, Inc. was used. If aligned and calibrated correctly it gives the heading of the user which is seen as an extremely important measurement for location based services in indoor scenarios.

C. RFID

The interface for this type of sensor is currently under development. An RF-signal-strength based approach is chosen and further work will be carried out in this area. So far, a simulation class for RFID Sensors is available. The class shows the proximity effect of this short range identification technology. The achievable range of such systems is up to 20m with directional antennas and active transponders in the UHF band.

D. WirelessLAN/Bluetooth

There are already some systems available that rely on the measurement of signal strength output from WirelessLAN access points [5], [6]. They show very promising performances even in indoor scenarios, so that it is intended to first interface to such a system rather than implement another integrated solution.

E. Floor Plan

Map based positioning and floor plans are very important in indoor scenarios, as they give valuable a-priori information regarding areas in which a person can move. The representation of maps as reasonable likelihood functions is therefore a key concept in our framework.

V. MOVEMENT MODEL

Having a movement model that reproduces the behavior of the underlying dynamic system is a prerequisite for the application of Bayesian filtering algorithms. In the case of tracking moving objects, the prediction stage (equation (2)) relies on such a movement model to compute the probability density $p(x_k|x_{k-1})$ of the system state for time step k, given a certain configuration of the state at step k-1. In the context of particle filtering, this density is used to sample a particle's new state during each step of the filter algorithm. The development of a suitable movement model was a major focus for the work presented here.

In our case, we model how moving speed and direction of a person depend on his state. Several parameters affect the movement of a human being. Speed and direction at a certain time instant directly affect the position of a person at the next point in time. Speed and direction are affected by additional state variables, such as the current activity of the person, his activeness, disorientation and emotions.

A. State Variables

Our human movement model is, therefore, parameterized at the physical level by the parameters *speed*, *direction* and *position* in space. We also introduce N_s additional *movement influencing state variables* that affect these physical parameters, and which are categorized into three groups (see also figure 2):

1) The first group: is directly dependent on the current position (and direction) of the pedestrian. For examples of this category one can name the surface steepness or the density of physical obstacles at the pedestrian's position. In our implementation, a manually edited gray scale map of the area of interest is used to model the states pertaining to steepness and obstacles, where different gray levels represent different state values.

2) The second group: are static values since they do not depend on the previous time step, and they are assumed to be known by the system. They are called "known system parameters"; an example is the age of the person and can be derived from external data sources or manual input data.

3) The third group: cannot be measured directly or stored in a database because they vary according to the human behavior. These are states where such attributes as emotions, activeness, activity and arousal play an important role, and they are modeled using Markov models. The idea of using Markov chains for describing human behaviors could also be found in [7], [8] and [9]. Transition probabilities of these Markov models are set according to a-priori assumptions [10]. In our implementation they are set on a one second time interval base.

For our model, we consider N_s = 11 parameters that affect the human movement to a certain degree (see table I). This list can be extended for more complex scenarios or different applications.

Type of state	**state**	**weight**
1: Position dependent	obstacles	8
	steepness	8
2: System defined	weather	5
	age	6
	time of day	4
	weekday	4
3: Dependent on individual user	activity	11
	disorientation	12
	emotion	2
	arousal	3
	activeness	6

TABLE I
PARAMETERS THAT AFFECT THE HUMAN MOVEMENT

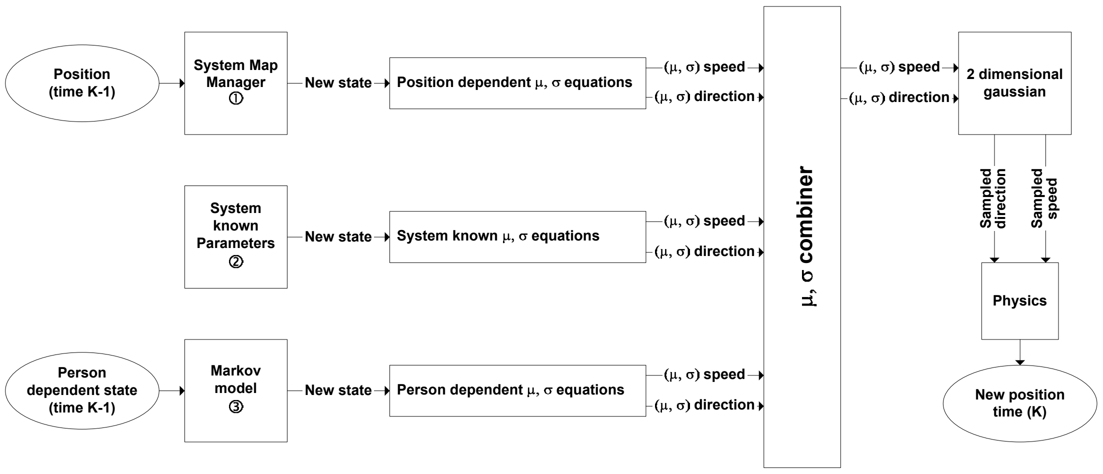

Fig. 2. Movement model with three different types of states for the determination of μ and σ

It is important to point out that the given parameters are chosen more or less arbitrarily in order to study the general idea of non-movement context variables influencing movement variables and vice versa and the possibility to infer these variables from each other. While we tried to make reasonable assumptions, we do not suggest that either the specific selection, their parametrization, dependencies or their associated weights assumed for this study model the real world exceptionally well.

Age (walking)	Mean speed (m/min)	Mean direction (degrees)	Sigma speed (m/min)	Sigma direction (degrees)
Child (a=0)	45	old direction	15	45
Mid (a=0.5)	82	old direction	13	15
Old (a=1.0)	60	old direction	11	30

TABLE II
AGE STATISTICS

B. Calculation of new Speed and Direction

We applied some known physiological and psychological evaluations [10] to parameterize our model assuming Gaussian distributions for its main physical random variables; speed v and direction α.

Relationships between the mean and the standard deviation of the user's speed and direction, and the specific values of each of the N_s = 11 state variables were based on statistical data [10], [11], as well as common-sense assumptions.

For each variable in turn, and here for the mean and standard deviation of speed and direction, a table of dependency between the state variable (e.g. Age = *Child*, *Mid* or *Old*) and the corresponding means and sigmas was set up (table II). Then, curve fitting was used to generate simple polynomial expressions out of these tables. These expressions can be found in the movement model (figure 2) as "μ, σ equations". For example, the resultant equations that relate the age state variable (the user's normalized age = $a; 0 \leq a \leq 1$) to the means and standard deviations of speed and direction are shown in equations (4) to (7):

$$\mu_v^{\text{age}} = -118 \cdot a^2 + 133 \cdot a + 45 \quad (4)$$
$$\sigma_v^{\text{age}} = -4 \cdot a + 15 \quad (5)$$
$$\mu_\alpha^{\text{age}} = \alpha(k-1) \quad (6)$$
$$\sigma_\alpha^{\text{age}} = 90 \cdot a^2 - 105 \cdot a + 45 \quad (7)$$

where μ_v^{age}, is the mean speed, σ_v^{age} is the standard deviation of speed, μ_α^{age} is the mean direction, and $\sigma_\alpha^{\text{age}}$ is the standard deviation of direction. Thus our N_s state variables of table (I) result in N_s means and standard deviations for speed and direction: μ_v^i, σ_v^i, μ_α^i, and σ_α^i; $1 \leq i \leq N_s$.

So far, these N_s states assume that each parameter *independently* influences the next chosen means and standard deviations of speed and direction. The next step is to combine these to a single new mean and standard deviation for speed and direction. This is done by applying simple linear weighting of these means and standard deviations. For example, the weighted average of the mean of the new speed, $\overline{\mu_v}$, is calculated by:

$$\overline{\mu_v} = \frac{\sum_{i=1}^{N_s} \mu_{v,i} \cdot \gamma_i}{\sum_{i=1}^{N_s} \gamma_i} \quad (8)$$

where γ_i is the weight corresponding to the i-th state variable (see table I). Finally, $\overline{\mu_v(k)}$ is limited such that:

$$-\triangle v_{min} \leq \overline{\mu_v(k)} - \overline{\mu_v(k-1)} \leq \triangle v_{max} \quad (9)$$

where $\triangle v_{max}$ and $\triangle v_{min}$ specify the maximum acceleration and deceleration, respectively (speed change over a single interval from $k-1$ to k). These two limits represent the maximum acceleration and deceleration that a normal human being can undergo during the time interval of interest. Calculating the final mean and standard deviation is the task of the "μ-σ Combiner" module in figure 2.

We can now use the mean speed (μ_v), the mean direction (μ_α), the standard deviation of speed (σ_v) and the standard deviation of direction (μ_α) at time step k to parameterize two assumed Gaussian distributions. New speed and new direction values v_k and α_k are then sampled from their respective Gaussian distributions [12].

C. Position Calculation

The sampled speed and direction will be used to calculate the new position for the next time step using the equations of motion as shown in figure 2. The distance l travelled by the pedestrian within a time duration of $\triangle t$ and with a speed v can be calculated according to:

$$l_k = v_k \cdot \triangle t \tag{10}$$

The new position can be calculated as follows, using these motion expressions:

$$x(k+1) = x(k) + l_k \cdot \cos[\alpha(k+1)] \tag{11}$$

$$y(k+1) = y(k) + l_k \cdot \sin[\alpha(k+1)] \tag{12}$$

The new position $(x(k+1), y(k+1))$ will be the input to the movement model at the next time step k+1 to continue the prediction. The movement model will use this new position to calculate obstacles and steepness states for the next time step. The old activity, orientation, arousal, emotions, activeness conditions are used to calculate their new conditions at the next time step according to their appropriate Markov models.

VI. MEASUREMENTS AND SIMULATIONS

In this section we present the results of simulations with noisy measurements which lead to position errors. We quantify the influence by computing the average (absolute) position error. This error is strongly dependent on the number of particles used in the fusion algorithm. These results help to find optimal parameters for the measurements with real sensors and they show in a promising way, that even very noisy measurements can be used to improve the position estimation.

In our location server, we replaced real sensor data by a simulation class for a GPS receiver and a compass due to lack of a reference measurement giving ground truth data.

To simulate the error of the measurement equipment, Gaussian noise was added to the simulated "true" values. The errors are expressed by variance σ (standard deviation) around the predefined values (position, angle). This standard deviation was then varied as indicated in the figures. These noisy simulated measurements are finally fed into the particle filter and the result of this step is the position estimation which is then compared to the true position of the pedestrian.

A. GPS variation

This comparison gives a quantitative measure for the performance of the particle filter. The results for average positioning errors versus the number of used particles are presented in figure 3. The different curves (except the upper black one) have resulted from different standard deviations for the GPS, whilst the compass has a fixed σ of 25°.

It is visible that as the standard deviation decreases, the average position error decreases also. As expected, using a more accurate GPS receiver (or satellite constellation; DOP) results in a more accurate location estimation. This figure also shows that having a large number of particles compensates for the noisy GPS receiver.

The average absolute error with particle filtering is lower than the error of the GPS sensor. This can be shown if we do not use particle filtering at all. The analytical approach to this is the combination of two random variables (x- and y-components of the position measurement), which then has Rayleigh distribution according to [13]: $\mathbf{z} = \sqrt{\mathbf{x}^2 + \mathbf{y}^2}$. A standard deviation of the GPS of $\sigma = 25$m leads to a mean position error of: $E\{\mathbf{z}\} = \sigma\sqrt{\frac{\pi}{2}} = 31.33285$m, which is significantly higher than the filtered measurements with an error of around 10m (blue line in figure 3).

Compared to a very simple movement model, our improved movement model has shown a noticeable reduction in the average position error. The performance of this simple model (only particles implement the simple model) is shown in the upper black curve. In this model, the standard deviation of the GPS is assumed to be 25m and the speed can only change between two states: 0m/s and 2m/s. Gaussian noise is added to the speeds and the compass heading. The new positions are calculated from the old position, the speed, the angle and the time between the measurements. Still, the comparison to a real walking human is missing. This is subject of further research, when our reference system is established.

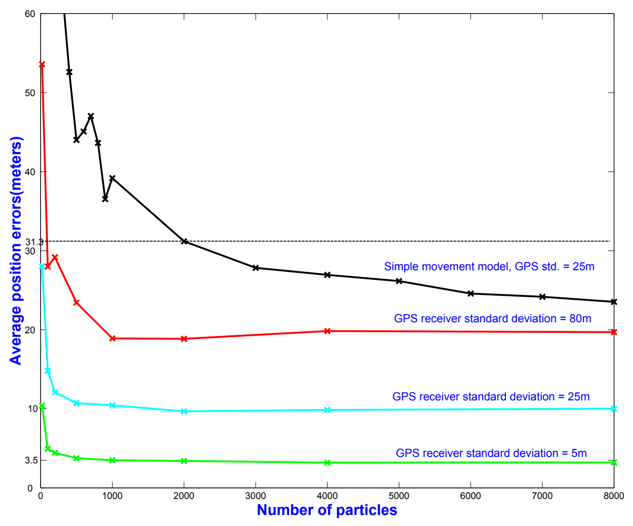

Fig. 3. Average position error vs. number of particles averaged over 5000 time steps; standard deviation: compass 25° and GPS variable

B. Compass variation

A very interesting effect can be shown in figure 4. Here, the variance of the GPS error is fixed, and the variance of the compass error is varied. The compass noise does not only affect the direction estimation, but also the position estimation as can be seen from the figure and also from the equations (11) and (12).

Increasing the accuracy of the compass reduces the position error to some extent. It can also be seen that for few particles (less than 2000) and a very accurate compass (compass is simulated to be very precise with only 0.5° standard deviation) the system does not converge. This is because all the particles vanish during the resampling step of the filter as they obtain very low weights if they do not move exactly in the direction of the tracked object.

Since the computation time increases linearly with the number of particles used, it takes apparently longer until a steady state is reached in the case of smaller standard deviation of the simulated compass error. However, the absolute positioning error is of course smaller as mentioned before.

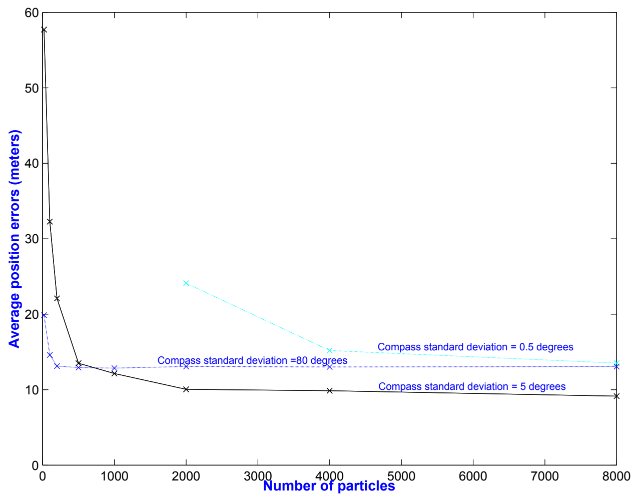

Fig. 4. Average position error vs. number of particles averaged over 5000 time steps; standard deviation: GPS 25m and compass variable

VII. SUMMARY AND OUTLOOK

Our framework is able to collect data from various sensors and combine them in a reasonable way. This is done by a common interface to the location server, which uses particle filtering with a special human movement model for the location estimation of pedestrians. Finally all the generated location objects can be displayed by the visualization server (see figure 1).

The setup can be used in a real time environment with attached sensors as well as a simulation system with various sensors, that are probably not even yet available.

The results of the performed simulations show in a promising way, how a good movement model can be used to improve the position estimation from noisy sensor data. In our case, we characterized the human walk by different human specific system states and Markov models. Each particle in the system holds such individual states.

In our simulations with particle filtering we could demonstrate an enhancement of the noisy position estimation compared to the unfiltered values. This is an advantage especially in indoor scenarios where satellite navigation is difficult. The enhancement is dependent on the quality of the movement model and on the noise of the measured data (section VI).

For numbers of particles over 2000, no further significant error reduction could be achieved. In the case of a very strictly defined measurement accuracy, the system sometimes did not converge with less than 2000 particles and much more particles were needed to reach the steady state.

We are working on an absolute reference system for the real position of the pedestrian to give results regarding the absolute position accuracy. This is subject to further research in indoor scenarios, where the developed algorithms and software will be used to track a user with measurements from RFID-Beacons, Bluetooth Access Points and cheap Inertial Measurement Units (IMU). Tuning of the movement model and additional states of the particles will hopefully further increase the accuracy of an estimation of the user's position and his situation.

REFERENCES

[1] G. Abwerzger, B. Hofmann-Wellenhof, B. Ott, and E. Wasle, "GPS/SBAS and Additional Sensor Integration for Pedestrian Applications in Difficult Environments," in *Institute of Navigation Global Navigation Satelite System - ION GNSS 2004*, Long Beach, California, 21–24, 2004.

[2] M. Angermann, J. Kammann, P. Robertson, A. Steingaß, and T. Strang, "Software Representation for Heterogeneous Location Data Sources Within A Probabilistic Framework," in *International Symposium on Location Based Services for Cellular Users - Locellus 2001*, February 2001, pp. 107–118.

[3] M. S. Arulampalam, S. Maskell, N. Gordon, and T. Clapp, "A Tutorial on Particle Filters for Online Nonlinear/Non-Gaussian Bayesian Tracking," *IEEE Transactions on Signal Processing*, vol. 50, pp. 174–188, February 2002.

[4] N. Gordon, D. Salmond, and A. Smith, "Novel approach to nonlinear and non-Gaussian Bayesian state estimation," vol. 140. Inst. Elect. Eng., 1993.

[5] P. Ibach, V. Stantchev, F. Lederer, A. Wei, T. Herbst, and T. Kunze, "WLAN-base Asset Tracking for Warehouse Management," in *IADIS International Conference e-Commerce*, Porto, Portugal, December 2005.

[6] (2006) Ekahau Positioning Engine. [Online]. Available: http://www.ekahau.com/

[7] A. Pentland and A. Liu, "Modeling and Prediction of Human Behavior," *Neural Computation*, vol. 11, no. 1, pp. 229–242, 1999.

[8] T. Zhao and R. Nevatia, "3D Tracking of Human Locomotion: A Tracking as Recognition Approach," in *16th International Conference on Pattern Recognition, 2002*, December 2002.

[9] A. Adam and S. Amershi, "Identifying Humans by Their Walk and Generating New Motions Using Hidden Markov Models," The University of British Columbia, Topics in AI: Graphical Models and Computer Animation, Tech. Rep., December 2004.

[10] G. A. Bekey, "Walking," *The Handbook of Brain Theory and Neural Networks, MIT press*, pp. 1045–1049, 1995.

[11] Green, R.D., and L. Guan, "Quantifying and Recognizing Human Movement Patterns from Monocular Video Images - part I: A new Framework for Modeling Human Motion," *IEEE Transactions on Circuits and Systems for Video Technology*, pp. 179–190, 2004.

[12] A. Leon-Garcia, *Probability and Random Process for Electrical Engineering*, 2nd ed. Addison-Wesley Publishing Company, 1994.

[13] A. Papoulis and S. U. Pillai, *Probability, Random Variables and Stochastic Processes*, 4th ed. McGRAW-Hill Book Company, 2002.