Integration of Foot-Mounted Inertial Sensors into a Bayesian Location Estimation Framework

Bernhard Krach and Patrick Robertson

***Abstract*— An algorithm for integrating foot-mounted inertial sensors into a Bayesian location estimation framework is presented. The proposed integration scheme is based on a cascaded estimation architecture. A lower Kalman filter is used to estimate the step-wise change of position and direction of the foot. These estimates are used in turn as measurements in an upper particle filter, which is able to incorporate nonlinear map-matching techniques. Experimental data is used to verify the proposed algorithm.**

***Index Terms*— Pedestrian Navigation, Inertial Integration, Indoor Navigation, Map-Matching**

I. Introduction

The use of inertial sensors is becoming widespread for pedestrian navigation, especially for indoor applications. Basically two approaches can be distinguished. The pedometer-approach employs an accelerometer for detecting individual steps whilst the stride length and stride direction are themselves estimated using additional sensors, such as global navigation satellite systems (GNSS), or a priori information. Given a detected step, its length and its direction, a person's position can be determined by dead-reckoning [1] [2] [3]. Other methods have been studied in [4]. The latest approaches are based on full six degree of freedom (6DOF) inertial navigation. A foot-mounted 6DOF strapdown inertial platform comprising triads of accelerometers and gyroscopes is used to dead reckon via a conventional strapdown navigation algorithm. An indirect feedback extended Kalman filter runs in parallel to the strapdown algorithm. Rest phases of the foot, which are detected from the accelerometer signals, trigger zero-velocity (virtual) measurements that are used to update the filter (ZUPT). Due to the regular ZUPT measurements we can estimate and correct the drift errors, which accumulate in the strapdown solution [5] [6] [7] [8]. It was shown in [5] that this approach can achieve very good performance even with today's low-cost micro-electro-mechanical (MEMS) sensors because the ZUPTs are so frequent that errors build up only slowly during each step the pedestrian makes. Nevertheless, the proposed Kalman filter approach is not optimal, as the algorithm does not take into account prior dynamic knowledge about the motion of the pedestrian or the motion of her foot and there is no mathematically sound procedure when considering the incorporation of nonlinear map-matching techniques or additional nonlinear / non-Gaussian sensors typically used in an indoor scenario.

To address this here we propose a cascaded estimation architecture: To estimate the navigation parameters of the foot we use a state-of-the-art integration filter comprising a conventional strapdown navigation algorithm along with an indirect feedback extended Kalman filter and a ZUPT detection algorithm for the foot/shoe that is suitably equipped with a 6DOF inertial sensor suite [5]. For each step we compute foot displacement and heading change values from the foot's filter and exploit them as measurements within a higher-level main fusion (particle) filter, which is able to consider the nonlinear dynamics of the human by means of a dedicated pedestrian movement model, including also maps and building constraints, and which operates at a much lower sampling rate.

The paper is organized as follows: At first a brief review of sequential Bayesian estimation and particle filtering is given. Subsequently our integration approach is motivated and details on the filter design are addressed, including the choice of an appropriate proposal density for the upper-level particle filter. Experimental results conclude the paper.

II. Sequential Estimation

A. Optimal Solution

The task of a navigation system is commonly to estimate successively a set of navigation parameters, here referred to as the hidden state $\mathbf{x}_k$, based on an evolving sequence of noisy measurements $\mathbf{z}_k$ (over the temporal index k). If the future state given the present state and all its past states depend only on the present state (and not on any past states), the temporal evolution of navigation parameters can be modeled as a first-order Markov process as illustrated in Figure 1. If it is also assumed that the noise affecting successive measurements is independent of the past noise values, such that each observation depends only on the present state, the optimal solution is given by the application of *sequential Bayesian estimation*. The reader is referred to [9] which gives a derivation of the general framework for optimal estimation of temporally evolving (Markovian) parameters by means of inference; and we have chosen similar notation. The entire history of observations can be written as

$$\mathbf{Z}_k \widehat{=} \{\mathbf{z}_q, q = 1, \ldots, k\} \ , \tag{1}$$

It can be shown that the sequential estimation algorithm is recursive, as it uses the posterior PDF computed for time instance $k-1$ to compute the posterior PDF for instance k . For a given posterior PDF at time instance $k-1$, $p(\mathbf{x}_{k-1}|\mathbf{Z}_{k-1})$, the *prior* PDF $p(\mathbf{x}_k|\mathbf{Z}_{k-1})$ is calculated in the so-called *prediction step* by applying the Chapman-Kolmogorov equation:

$$p(\mathbf{x}_k|\mathbf{Z}_{k-1}) = \int p(\mathbf{x}_k|\mathbf{x}_{k-1})p(\mathbf{x}_{k-1}|\mathbf{Z}_{k-1})\mathrm{d}\mathbf{x}_{k-1} \ , \tag{2}$$

The authors are with the German Aerospace Center DLR, Institute of Communications and Navigation, Oberpfaffenhofen, 82234 Wessling, Germany (e-mail: `firstname.lastname@dlr.de`).

978-1-4244-1799-5/08/$25.00 ©2008 IEEE

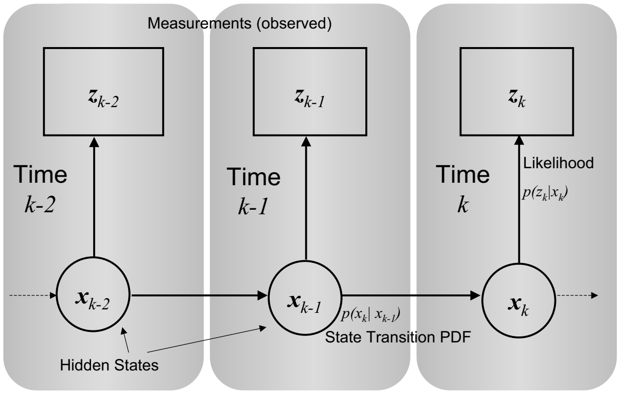


Fig. 1. Illustration of the hidden Markov estimation process for three time instances. Our measurements are the sequence $\mathbf{z}_q, q = 0, \ldots, k\}$, and the parameters to be estimated are $\mathbf{x}_q, q = 0, \ldots, k\}$

with $p(\mathbf{x}_k|\mathbf{x}_{k-1})$ being the state transition PDF of the Markov process. In the *update step* the new posterior PDF for step k is obtained by applying Bayes' rule to $p(\mathbf{x}_k|\mathbf{z}_k, \mathbf{Z}_{k-1})$ yielding the normalized product of the likelihood $p(\mathbf{z}_k|\mathbf{x}_k)$ and the prior PDF:

$$\begin{aligned} p(\mathbf{x}_k|\mathbf{Z}_k) &= p(\mathbf{x}_k|\mathbf{z}_k, \mathbf{Z}_{k-1}) \\ &= \frac{p(\mathbf{z}_k|\mathbf{x}_k, \mathbf{Z}_{k-1})p(\mathbf{x}_k|\mathbf{Z}_{k-1})}{p(\mathbf{z}_k|\mathbf{Z}_{k-1})} \\ &= \frac{p(\mathbf{z}_k|\mathbf{x}_k)p(\mathbf{x}_k|\mathbf{Z}_{k-1})}{p(\mathbf{z}_k|\mathbf{Z}_{k-1})} \ . \end{aligned} \tag{3}$$

B. Suboptimal Solutions

The optimal estimation algorithm relies on evaluating the integral (2), which is usually a very difficult task, except for certain additional restrictions imposed on the model and the noise process. Thus beside the restricted optimal algorithms such as the Kalman filter or the grid-based methods, a large number of suboptimal algorithms exist, e.g. the extended and the Sigma-Point Kalman Filter, which are nonlinear adaptations of the generic Kalman filter concept [10].

A further family of suboptimal algorithms are the Sequential Monte Carlo (SMC) filters [9] [11]. In these algorithms the posterior density at step k is represented as a sum, and is specified by a set of N_p particles:

$$p(\mathbf{x}_k|\mathbf{Z}_k) \approx \sum_{j=1}^{N_p} w_k^j \cdot \delta(\mathbf{x}_k - \mathbf{x}_k^j) \ , \tag{4}$$

where each particle with index j has a state $\mathbf{x}_k^j$ and has a *weight* w_k^j. The sum over all particles' weights is one. The SMC filters are not restricted with respect to the model and the noise process, but the number of particles is a crucial parameter, as only for $N_p \rightarrow \infty$ does the approximate posterior approach the true PDF. The particles are drawn according to a so-called proposal density, $q(\mathbf{x}_k|\mathbf{x}_k^j, \mathbf{z}_k)$, such that their respective weight is calculated using

$$w_k^j \propto w_{k-1}^j \frac{p(\mathbf{z}_k|\mathbf{x}_k^j)p(\mathbf{x}_k^j|\mathbf{x}_{k-1}^j)}{q(\mathbf{x}_k^j|\mathbf{x}_{k-1}^j, \mathbf{z}_k)} \ . \tag{5}$$

The selection of the proposal density is crucial for the performance of the particle filter. Although the optimal proposal density can be derived theoretically [9], it is in practice often both impossible to actually draw from this density and very difficult to compute the corresponding weight according to (5).

C. Incorporation of Independent Sensors

This section deals with the case where a range of M sensor outputs makes up the overall measurement vector $\mathbf{z}_k$. Separating the measurement vector $\mathbf{z}_k$ into sub-vectors for each sensor

$$\mathbf{z}_k \hat{=} \{\mathbf{z}_{m,k}, m = 1, \ldots, M\} \ , \tag{6}$$

and writing $\mathbf{z}_{m,k}^-$ for $\mathbf{z}_k$ after omitting $\mathbf{z}_{m,k}$, i.e. $\mathbf{z}_{m,k}^- = \mathbf{z}_k \backslash \mathbf{z}_{m,k}$. If we assume independent perturbations of the sub-vectors then this is equivalent to writing

$$p(\mathbf{z}_{m,k}|\mathbf{x}_k, \mathbf{z}_{m,k}^-) = p(\mathbf{z}_{m,k}|\mathbf{x}_k) \ , \tag{7}$$

so that given the actual state, the measurements $\mathbf{z}_{m,k}^-$ will not affect the measurement $\mathbf{z}_{m,k}$. In this case the overall likelihood function can be written in product form according to the factorization of Bayes' rule [12] as

$$p(\mathbf{z}_k|\mathbf{x}_k) = C \cdot \prod_{m=1}^{M} p(\mathbf{z}_{m,k}|\mathbf{x}_k) \tag{8}$$

with C being a normalizing constant. In other words, the sensors can be incorporated into the weight update (5) by simple multiplication.

III. Integration of Inertial Sensors

A. Motivation of Cascaded Approach

The most widespread approach to integrate strapdown inertial sensors into a navigation system is to use a direct/indirect extended Kalman filter together with a strapdown navigation computer [13] [14] [15]. The combination of the two algorithms may be interpreted as a "probabilistic" inertial navigation system (INS) and allows one to calculate an approximation of the posterior PDF of position, velocity, attitude, and sensor errors based on the sequence of measurement received from the sensors of the 6DOF inertial platform. The approximated posterior/prior PDF is a Gaussian, whose mean is given by the strapdown solution corrected by the Kalman filter state vector and whose covariance matrix is given by the covariance matrix of the Kalman filter. The advantage of this approach is that the resulting Gaussian PDF can be joined analytically with linear/linearized Gaussian likelihoods of further sensors during the filter update step (5) as described in the previous section.

Despite the fact that the Kalman filter implements a Bayesian filter, this integration approach suffers from the major drawback that it does not follow (2) and (3) straightforwardly for two reasons:

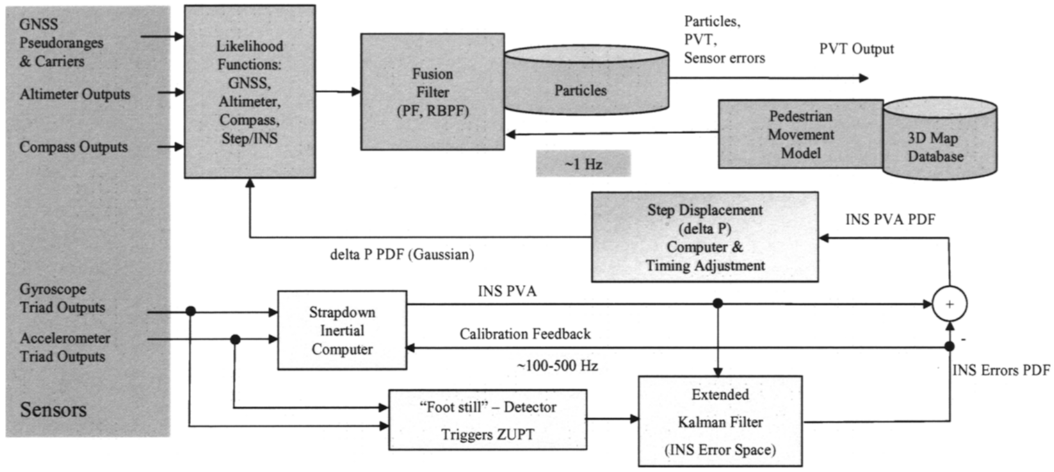


Fig. 2. Bayesian location estimation framework architecture with upper particle filter (dark gray) and lower Kalman filter for stride estimation (light gray)

- The Kalman filter indeed uses a probabilistic state transition model, but this model is based solely on pure kinematic relations between velocity, position, attitude, and sensor errors rather than on a true probabilistic characterization of the dynamics of the tracked object (e.g. a person traveling by foot).
- No likelihood function is used to incorporate the accelerometer and gyroscope measurements into the algorithm. Accelerometer and gyroscope measurements enter the algorithm directly via the strapdown computations and no explicit use is made of any prior knowledge about the object's dynamics. As a consequence the performance of a conventional INS is mainly determined by the quality of the inertial sensors.

To overcome this drawback it would indeed be optimal to formulate a Bayesian estimator whose dynamic model characterizes - besides position, velocity, attitude, and sensor errors - also accelerations and turn rates of the navigating object using a Markov chain whose state transitions occur at the sensor measurement rate, which is relatively high for inertial sensors. Due to nonlinear state evolution constraints this can be generally a very difficult task, especially when considering a Markov-chain characterization of a pedestrian and the motion of her foot/shoe on which the INS is mounted. Because of this, currently only the conventional integration approach seems to be feasible in order to estimate just the foot's movement for each step. Indeed, for the considered application of foot-mounted inertial sensors this is not a major drawback, as the inertial sensor errors can be constrained efficiently through the use of ZUPT measurements.

However, it is generally desirable to consider further prior dynamic knowledge about the pedestrian in an overall navigation filter. To take benefit of both the accurate foot-mounted inertial system and a dedicated pedestrian movement model including nonlinear effects such as building plans, we propose a cascaded estimation architecture as illustrated in Figure 2. We have decided to employ a particle filter framework for the upper level fusion filter. This is because we will include sensors and process models (movement models) that are nonlinear, and often with non-Gaussian noise models. In particular the movement model needs to incorporate the building plan which is highly nonlinear. A lower Kalman filter is used to provide stepwise computed values of foot displacement and heading change, here referred to as the step-measurement, which is used as measurements within the upper particle filter and enters the algorithm via a Gaussian likelihood function along with the measurements and likelihoods of further available sensors.

Our framework has been implemented in the Java programming language and can process incoming sensor data in real-time, allowing live visualization of the location estimate. The filter will perform sensor fusion roughly every two seconds or when triggered to do so by a specific sensor - in our case we will perform an update cycle at the latest once every two seconds and also upon each step-measurement.

To distinguish the low rate operations of the upper filter from the high rate operations of the lower filter below, the terms k*-rate* and l*-rate* are introduced. The upper filter is associated with the k-rate, which is approximately the step-rate, and the lower filter is associated with the l-rate, which is given by the rate of the inertial sensors. Corresponding variables will be indicated by the subscripts $(\bullet)_k$ and $(\bullet)_l$.

B. Upper Filter

The particle filter adopts re-sampling of the particles at every time step. With exception of the step-measurement it adopts the state transition probabilities as proposal function and uses the product of the sensors' likelihood functions in the weight computation (8), (5) - the standard SIR formulation [9]. The incorporation of the INS-step-measurement, however, does not follow this approach, as will be explained later.

1) State Model: In the particle filter we keep track of the pedestrian's position $\mathbf{r}_k$ and her heading Ψ_k. To allow the incorporation of the step-measurement the state vector has been extended by the step specific states, $\Delta\mathbf{r}_k$ and $\Delta\Psi_k$, which relate $\mathbf{r}_k$ and Ψ_k to the time index $k-1$. We define:

$$\mathbf{x}_k = \begin{pmatrix} \mathbf{r}_k \\ \Psi_k \\ \Delta\mathbf{r}_k \\ \Delta\Psi_k \end{pmatrix} \tag{9}$$

Here we have chosen that the new location and heading depend deterministically on the past state (and on the current state through the Δ-states). The Δ-states encode the change in location and in heading during the last step, and allow us to write:

$$\begin{aligned} \mathbf{r}_k &= \mathbf{r}_{k-1} + \mathbf{C}(\Psi_{k-1})\Delta\mathbf{r}_k \ , && (10) \\ \Psi_k &= \Psi_{k-1} + \Delta\Psi_k \ , && (11) \end{aligned}$$

where $\mathbf{C}(\Psi_{k-1})$ is the rotation matrix, where we have assumed that the rotation Ψ_{k-1} is always in the horizontal plane:

$$\mathbf{C}(\bullet) = \begin{pmatrix} \cos(\bullet) & -\sin(\bullet) & 0 \\ \sin(\bullet) & \cos(\bullet) & 0 \\ 0 & 0 & 1 \end{pmatrix} \ . \tag{12}$$

2) Measurement Model: The step-measurement $\mathbf{z}_k$, which will be the only used measurement within the scope of this paper, is assumed to depend only on the current state $\mathbf{x}_k$ and a noise term $\mathbf{n}_\Delta$:

$$\mathbf{z}_k = h(\mathbf{x}_k, \mathbf{n}_\Delta) \ . \tag{13}$$

In particular we use

$$\mathbf{z}_k = \begin{pmatrix} \Delta\mathbf{r}_k \\ \Delta\Psi_k \end{pmatrix} + \mathbf{n}_\Delta \ , \tag{14}$$

with $\mathbf{n}_\Delta$ being zero-mean element-wise uncorrelated Gaussian noise. The variances are adjusted to reflect the uncertainty of the step-measurement.

3) Movement Model: A probabilistic movement model is used to characterize the temporal evolution of a state $\mathbf{x}_k$. Given that this evolution follows a Markov process the movement can be characterized by a transitional density $p(\mathbf{x}_k|\mathbf{x}_{k-1})$, and our model follows the Markovian approach. The movement model used here aims to reflect the physical constraints that are imposed on the movement of a pedestrian, in particular in an indoor scenario, where walls can have a large impact. Formally, the new state $\mathbf{x}_k$ is assumed to depend only on the previous state $\mathbf{x}_{k-1}$ and a noise term $\mathbf{n}_d$:

$$\mathbf{x}_k = f(\mathbf{x}_{k-1}, \mathbf{n}_d) \ . \tag{15}$$

Recall from (10) that the new location and heading depend deterministically on the past state and on the current state through the Δ-states. However, the probabilistic part of the movement movement model is incorporated into the temporal evolution of the displacement states $\Delta\mathbf{r}_k$ and $\Delta\Psi_k$:

$$\begin{aligned} \Delta\mathbf{r}_k &= f(\mathbf{x}_{k-1}, \mathbf{n}_r) \ , && (16) \\ \Delta\Psi_k &= g(\mathbf{x}_{k-1}, \mathbf{n}_\Psi) \ , && (17) \end{aligned}$$

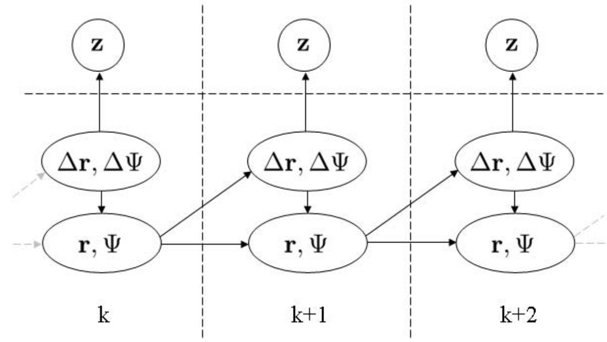


Fig. 3. Dynamic Bayesian network illustration of the pedestrian model used in the upper particle filter. The arrow joining two time slices and pointing diagonally and upward encodes the nonlinear probabilistic dependency of the pedestrian's step on the presence of walls at or near a certain location. The other arrows encode deterministic relationships.

which depend only on the past state $\mathbf{x}_{k-1}$ and the noise terms $\mathbf{n}_r$ and $\mathbf{n}_\Psi$. The nonlinearity that is imposed by the walls is included in (16) in that the displacement of the location $\Delta\mathbf{r}_k$ depends on the presence of nearby walls and obstacles.

An illustration of the pedestrian model used here in terms of a dynamic Bayesian network is shown in Figure 3.

A very simple movement model is used here: Given that a displacement $\Delta\mathbf{r}_k$ intersects with one of the walls that are stored in the map database, we assign it the probability $p(\mathbf{x}_k|\mathbf{x}_{k-1}) = 0$. In other situations, if a wall has not been crossed, we want to draw according to:

$$\begin{aligned} \Delta\mathbf{r}_k &= \mathbf{n}_r \ , && (18) \\ \Delta\Psi_k &= n_\Psi \ , && (19) \end{aligned}$$

where $\mathbf{n}_r$ and n_Ψ are drawn from mutually uncorrelated zero-mean white Gaussian noise processes, whose variances are adapted to the movement of a pedestrian. Despite the fact that this model is suitable for the case of a wall crossing, it is quite coarse otherwise, as is does not adequately represent the probability with which a pedestrian will move, given a known building layout or map [16]. To alleviate this, future work will incorporate more accurate movement models than the one used here.

C. Lower Filter

As the integration method proposed in [5] was shown to have both good performance and low complexity, we also follow this approach for the step estimation algorithm. The lower filter operates at the rate given by the output of the 6DOF sensor suite, which is in the range of 100–500 Hz, depending on the hardware settings.

1) Algorithm Fundamentals: A strapdown navigation algorithm [14] processes the vector of acceleration and turn rate measurements $\mathbf{z}_l = [\mathbf{a}_l \ \boldsymbol{\omega}_l]^T$, which is provided by the inertial sensors, to compute position $\mathbf{r}_l$, velocity $\mathbf{v}_l$, and attitude $\boldsymbol{\Psi}_l$. In parallel an extended Kalman filter is used to estimate the errors of the strapdown calculations. Typically 15 states are estimated by the filter [5], [13]: position errors $\delta\mathbf{r}_l$, velocity errors $\delta\mathbf{v}_l$, attitude errors $\delta\boldsymbol{\Psi}_l$, accelerometer biases $\delta\mathbf{a}_l$, and gyroscopic biases $\delta\boldsymbol{\omega}_l$. The error estimates $\delta\mathbf{r}_l$, $\delta\mathbf{v}_l$, and $\delta\boldsymbol{\Psi}_l$ are perturbations around the filter operating point $\mathbf{r}_l$, $\mathbf{v}_l$, $\boldsymbol{\Psi}_l$ that is calculated by the strapdown algorithm.

Recalling from section III-A that the lower filter provides estimates of position, velocity, attitude, and sensor errors in the form of a Gaussian PDF. In the subsequent processing only position and heading are states of interest and we write for concise notation:

$$\mathbf{x}_l = \begin{pmatrix} \mathbf{r}_l \\ \Psi_l \end{pmatrix} \ , \tag{20}$$

whereby Ψ_l is the yaw angle derived from $\boldsymbol{\Psi}_l$. From the posterior PDF of the lower filter the (marginalized) posterior $p(\mathbf{x}_l|\mathbf{Z}_l)$ can be derived in a straightforward manner. Note that we are addressing the Bayesian estimation only at the lower level here, and have not yet included other sensors than the INS or the pedestrian specific movement model.

2) Rest Phase Detection: The reliable identification of the foot's rest phases is crucial for the update of the lower filter. Different approaches have been proposed to trigger the ZUPT measurement [5], [6]. Here we basically follow these ideas and monitor the magnitude of the acceleration vector [6], which is sensed by the accelerometer triad. If the signal remains within a threshold interval around earth gravity for a certain time interval ZUPTs are triggered until the threshold condition is violated. In our approach the ZUPT detection is also used to trigger the update of the upper filter. Each time a ZUPT is triggered in the lower filter the elapsed time since the last update of the upper filter is checked. If this time exceeds a certain threshold, for instance one second as illustrated in Figure 4, a new update of the upper filter is initiated.

3) The Step Sensor: The lower filter is used to process the high rate inertial measurements. To exploit them in the upper filter a (virtual) step sensor model is derived from the lower filter error characteristics in order to provide a probabilistic estimate of the traveled distance and the change in heading for each step the pedestrian makes.

To provide the step measurements the following operations are performed at the interface between the lower filter and the virtual step sensor: As illustrated in Figure 5 each time a new upper filter cycle (k-cycle) is triggered (III-C.2) the expectation of the lower filter $\hat{\mathbf{x}}_l$ is stored in the variable $\hat{\mathbf{x}}_L \widehat{=} \hat{\mathbf{x}}_l$ with $L = k$. Please note that variables associated to the lower filter are indicated by the subscript $(\bullet)_L$ for those time instances l for which k-cycles are triggered. Introducing the step displacement variable $\Delta\mathbf{x}_L = \mathbf{x}_L - \mathbf{x}_{L-1}$ its expectation is almost independent from previous steps due to the ZUPTs that are applied. Thus we have $\Delta\hat{\mathbf{x}}_L = \mathbb{E}(\Delta\mathbf{x}_L|\mathbf{Z}_L))$, and may write for the displacement with respect to the coordinate system of the lower filter:

$$\begin{aligned} \Delta\hat{\mathbf{x}}_L &= \hat{\mathbf{x}}_L - \hat{\mathbf{x}}_{L-1} && (21) \\ &= \begin{pmatrix} \hat{\mathbf{r}}_L \\ \hat{\Psi}_L \end{pmatrix} - \begin{pmatrix} \hat{\mathbf{r}}_{L-1} \\ \hat{\Psi}_{L-1} \end{pmatrix} && (22) \\ &= \begin{pmatrix} \Delta\hat{\mathbf{r}}_L \\ \Delta\hat{\Psi}_L \end{pmatrix} \ . && (23) \end{aligned}$$

Also, since $\Delta\hat{\mathbf{x}}_L = \mathbb{E}(\Delta\mathbf{x}_L|\mathbf{Z}_L)) \approx \mathbb{E}(\Delta\mathbf{x}_L|\mathbf{Z}_L\backslash\mathbf{Z}_{L-1})$,

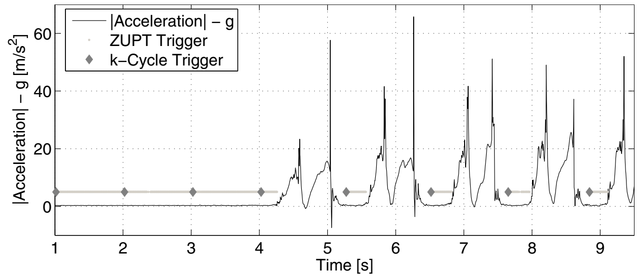


Fig. 4. Magnitude of acceleration vector subtracted by gravity g during the beginning of a walk sequence. ZUPT triggers and k-cycle triggers are shown along.

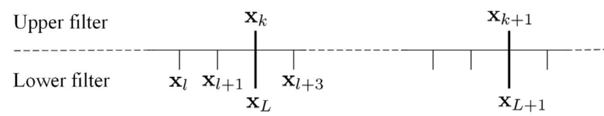


Fig. 5. Relation between upper and lower filter scheduling

we have independent measurements from step to step.

Finally, the displacement with respect to the heading at the previous k-cycle is computed and we have our step measurement

$$\mathbf{z}_k = \begin{pmatrix} \mathbf{C}^T(\Psi_\varepsilon)\mathbf{C}^T(\Psi_{L-1})\Delta\hat{\mathbf{r}}_L \\ \Delta\hat{\Psi}_L \end{pmatrix} \ . \tag{24}$$

The average heading misalignment of the inertial sensor platform with respect to the pedestrian's heading is given by the angle Ψ_ε, which has to be fixed initially.

D. Choice of an Appropriate Proposal Density

As mentioned in II-B the selection of the proposal density is crucial for the performance of the particle filter algorithm and it showed up to be an apparent problem for the design of the upper filter in our location estimation framework. If it is not possible to use the optimal proposal density a suitable choice is often the transition density. In this case the update step of particle filtering essentially incorporates the latest sensor evidence at each step in the form of the particles' weights through the likelihood function (SIR particle filter [9]). As the optimal choice for proposal density has been shown to be not appropriate here for complexity reasons, we followed the simple transition density approach in our framework initially. Despite the fact that the transitional density $p(\mathbf{x}_k|\mathbf{x}^j_{k-1})$ is a convenient choice, it is not optimal, since the latest evidence $\mathbf{z}_k$ is not incorporated in the proposal function the particles are drawn from. For instance, if the likelihood function is narrow compared to the density after the prediction step, then only a few "lucky" particles will subsequently receive significant weights. The result is usually sample impoverishment which degrades accuracy significantly for a given finite number of particles. As the likelihood function for our step-measurements is comparatively narrow due to the high accuracy of the step measurement, it is crucial to choose the proposal density other than the state transition density in order to avoid this problem. In other words it should be avoided to draw particles that do not follow the accurate step-measurement, because they will receive low weight from the step-likelihood during the update step anyway and hence are a waste of computational resources.

To address this drawback the auxiliary particle filter was proposed [17]. But especially for more extreme situations where the likelihood is much tighter than the prior, the optimal proposal comes very close to the likelihood itself. Here we have often such a situation: The step sensor is quite accurate, whereas the movement model is governed mainly by the surrounding walls. Hence for our problem here, it is more efficient to draw from a proposal function according to the step likelihood. Recalling the weight equation (5) the likelihood cancels out - up to a constant - with the proposal if we draw the displacement from $q(\Delta \mathbf{r}_k, \Delta\Psi_k|\mathbf{z}_k)$. We can draw from this proposal function for each particle because the measurement $\mathbf{z}_k$ does not depend on the state components $\mathbf{r}_k$, and Ψ_k, under the condition of a given $\Delta \mathbf{r}_k, \Delta\Psi_k$. Furthermore, the states $\mathbf{r}_k$, Ψ_k are computed deterministically using (10), (11). Using the more efficient "likelihood"-proposal we obtain the weight update

$$w_k^j \propto w_{k-1}^j p(\mathbf{x}_k^j|\mathbf{x}_{k-1}^j) \quad . \tag{25}$$

In this case the particles follow the step measurement and for each particle a disturbance of small Gaussian noise is superimposed at every step - the variance of this mean-free noise is equivalent to $\mathbf{n}_\Delta$. The weight is then calculated from the movement model corresponding to (25). This strategy ensures that enough particles survive at each step and impoverishment is avoided. A similar approach referred to as the likelihood particle filter was proposed in [9].

IV. RESULTS

The performance achievements of shoe-mounted INS as stand-alone or coupled with GNSS and / or magnetometer has been widely reported in the literature, for example in [5]. In this paper we present results that used no initial reference position, and no source of absolute position information such as GNSS. The chosen scenario is thus the following: a pedestrian moves through a building, using only the shoe-mounted INS. The other information available to the upper fusion filter is the building layout (floor-plan). We also assume that the user is within the specified building, and on a certain known floor.

As Fig. 6 shows, the upper fusion filter - a particle filter - starts with a uniform distribution of particles in the known area. Each particle, according to (9), includes its location and current heading. Over time only those particles will survive which are compatible with the layout of the floor-plan. In other words, those hypotheses of the state space will survive, which when moved according to thee lower fusion filter's estimate, have not crossed a wall. At first there are many such hypotheses, some moving in different directions compared to the true one, but over the course of time, only one hypothesis (the correct one), survives. In our case this was achieved in roughly one minute of walking.

Naturally, the rate of convergence and the reduction of modes will be a function of the actual route which was walked and of its relation to the floor plan restrictions. In an large hall without walls there will only be moderate reduction on the size of the remaining mode compared to the case with many walls. It should be noted that the surviving modes are "randomly" distributed across the layout and bear no relationship to the correct location (except the true mode, of course). As can be seen from the third time slice (25 s.) the true mode has already achieved its steady-state local uncertainty (of roughly the dimension of the corridor width). This implies that additional position information can be of significant value even if this is quite coarse (e.g. on the order of 10-50 meters).

In Fig. 7 we show an exemplary PDF as computed by the upper particle filter and after smoothing of the particle weights using a Gaussian kernel function. The user had only been walking for a short period of time and there are three modes still surviving.

V. CONCLUSIONS

In this paper a method for integrating shoe-mounted inertial sensors into a Bayesian location estimation framework is presented. The approach is characterized by a cascaded filter architecture, which allows to exploit the synergy between a conventional shoe-mounted INS and a nonlinear pedestrian movement model in an indoor scenario. An advantage of the proposed integration algorithm is that each level of the cascaded architecture can operate at an update rate appropriate to the scale: at 100 Hz or higher for the stride estimation and roughly at step-rate for the upper fusion layer. Based on experimental data it is shown that a moving pedestrian can be localized in a building just by using a foot-mounted 6DOF inertial platform and map matching without using any additional sensors and without the need to determine the pedestrian's initial position or heading in an alignment procedure. Furthermore, the experiment shows that due to the implicit map matching the uncertainty about the pedestrian's location decreases if the movement is suitable, which can lead to long-term stability in an indoor navigation scenario.

REFERENCES

[1] V. Gabaglio and B. Merminod, "Real-time calibration of length of steps with GPS and accelerometers," in *Proceedings of the ENC GNSS 1999*, Genova, Italia, Oct. 1999, pp. 599–605.

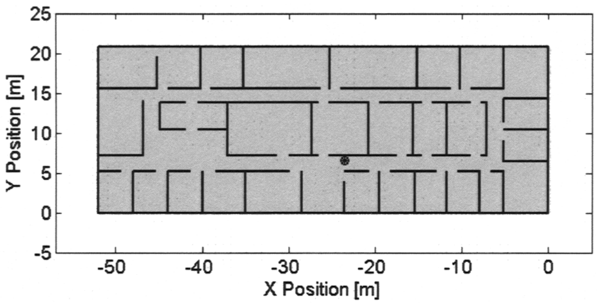
[2] Q. Ladetto, "On foot navigation: continuous step calibration using both complementary recursive prediction and adaptive Kalman filtering," in *Proceedings of the 13th International Technical Meeting of the Satellite Division of the Institute of Navigation ION GPS 2000*, 2000.

[3] J. Collin, O. Mezentsev, and G. Lachapelle, "Indoor positioning system using accelerometry and high accuracy heading sensors," in *Proceedings of the 16th International Technical Meeting of the Satellite Division of the Institute of Navigation ION GPS/GNSS 2003*, Portland, Orgeon, USA, Sept. 2003.

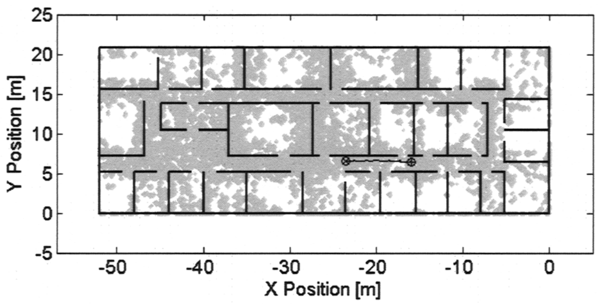
[4] D. Kubrak, C. Macabiau, and M. Monnerat, "Performance analysis of MEMS based pedestrian navigation systems," in *Proceedings of the 18th International Technical Meeting of the Satellite Division of the Institute of Navigation ION GNSS 2005*, Long Beach, California, USA, Sept. 2005.

[5] E. Foxlin, "Pedestrian tracking with shoe-mounted inertial sensors," *IEEE Computer Graphics and Applications*, vol. 25, no. 6, pp. 38–46, Nov. 2005.

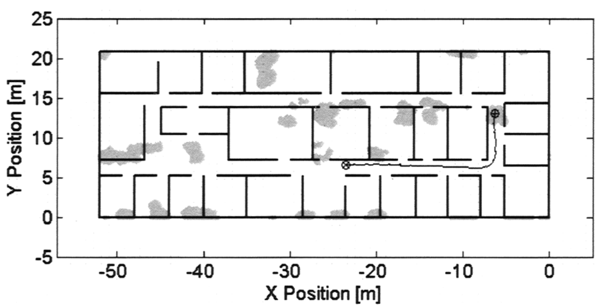
[6] S. Godha, G. Lachapelle, and M. E. Cannon, "Integrated GPS/INS system for pedestrian navigation in a signal degraded environment," in *Proceedings of the 19th International Technical Meeting of the Satellite Division of the Institute of Navigation ION GNSS 2006*, Fort Worth, Texas, USA, Sept. 2006.



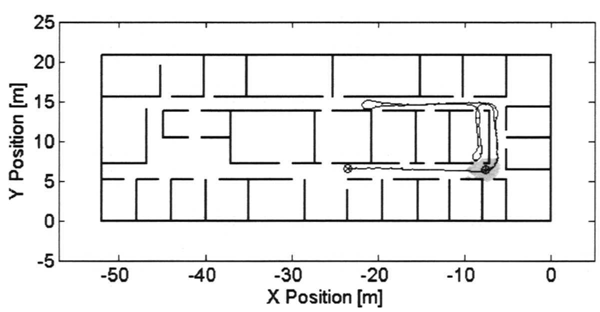
(a) Initial state



(b) After 10s



(c) After 25s



(d) After 80s

Fig. 6. Integration with map-matching in the upper particle filter: A pedestrian wearing the foot-mounted sensor walked the indicated track (black). At each figure the posterior position estimate (gray) becomes increasingly accurate, after 80s it is unimodal.

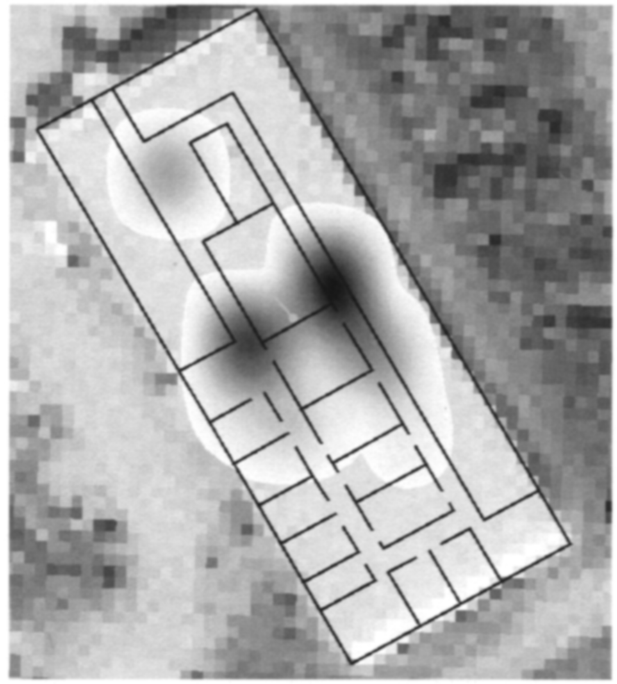


Fig. 7. An example of a tri-modal PDF as computed by the upper particle filter during an experiment with the real-time Java framework. Two main modes (and a minor mode) are still viable at this time.

[7] S. Beauregard, "Omnidirectional pedestrian navigation for first responders," in *Proceedings of the 4th Workshop on Positioning, Navigation and Communication, 2007, WPNC07*, Hannover, Germany, Mar. 2007, pp. 33–36.

[8] P. D. Groves, G. W. Pulford, C. A. Littlefield, D. L. J. Nash, and C. J. Mather, "Inertial navigation versus pedestrian dead reckoning: Optimizing the integration," in *Proceedings of the 20th International Technical Meeting of the Satellite Division of the Institute of Navigation ION GNSS 2007*, Fort Worth, Texas, USA, Sept. 2007, pp. 2043—2055.

[9] 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, no. 2, pp. 174–188, Feb. 2002.

[10] B. Ristic, S. Arulampalam, and N. Gordon, *Beyond the Kalman Filter - Particle Filters for Tracking Applications*. Boston-London: Artech House, 2004.

[11] A. Doucet, N. de Freitas, and N. Gordon, Eds., *Sequential Monte Carlo Methods in Practice*. New York: Springer, 2001.

[12] M. Angermann, J. Kammann, P. Robertson, A. Steingass, and T. Strang, "Software representation for heterogeneous location data sources using probability density functions," in *International Symposium on Location Based Services for Cellular Users, LOCELLUS 2001*, Munich, Germany, Feb. 2001, pp. 107–118.

[13] M. Grewal, L. Weill, and A. Andrews, *Global Positioning Systems, Inertial Navigation, and Integration*, 2nd ed., ser. IEE Radar, Navigation and Avionics Series. New York: John Wiley & Sons, Inc., 2001, vol. 5.

[14] D. H. Titterton and J. L. Weston, *Strapdown inertial navigation technology*, 2nd ed., ser. IEE Radar, Navigation and Avionics Series. London: Peter Peregrinus Ltd., 2004, vol. 5.

[15] "Advances in navigation sensors and integration technology," NATO RTO Eductional Notes, RTO-EN-SET-064, AC/323(SET-064)TP/43, NATO RTO, Feb. 2004.

[16] M. Khider, S. Kaiser, P. Robertson, and M. Angermann, "A novel movement model for pedestrians suitable for personal navigation," in *Proceedings of the Institute of Navigation National Technical Meeting 2008*, San Diego, California, USA, Jan. 2008.

[17] M. K. Pitt and N. Shephard, "Filtering via simulation: Auxiliary particle filters," *Journal of the American Statistical Association*, vol. 94, no. 446, 1999.