

Indoor pedestrian navigation based on recursive filtering

Philipp Hasper¹ and Gabriele Bleser²

¹ `hasper@cs.uni-kl.de`

² `Gabriele.Bleser@dfki.de`

Abstract. While localization is most commonly associated with GPS, many use cases remain where satellite-based navigation is too inaccurate or fails completely. In this seminar, we will present techniques usable for indoor localization of pedestrians. We will introduce several approaches using Inertial Measurement Units attached to the subject.

Due to the strong drifting behavior of those units, several steps are necessary to provide feasible accuracy: the use of filter techniques and the use of Zero Velocity Updates. We will explain the required state-space model and its application in recursive Bayesian filters like the Extended Kalman Filter or the Particle Filter. The use of aiding techniques is discussed and a map-aided, WiFi-initialized Particle Filter is presented.

Keywords: Sensor fusion, Recursive Filtering, Kalman Filter, Particle Filter, ZUPT, EKF, MCL

1 Introduction

Computer-aided localization is helpful in many different domains, mainly navigation. While satellite systems like GPS are nowadays widely used both in industrial and private context, many use cases remain for which they are not applicable or the provided accuracy is too low. Examples would be navigation in space, undersea or indoors.

We will concentrate on the latter use case and discuss several methods for deriving accurate position information from Inertial Measurement Units (IMUs). Since computing linear displacement from IMU sensor data requires integrating the body acceleration twice, a naive approach would amplify errors due to sensor noise and the influence of gravity (cf. Section 3.1). To perform odometry correctly, IMU data therefore has to be post-processed. This can be achieved by filter techniques such as the Kalman Filter or the Particle Filter in combination with suitable models which we will discuss in this paper. We will concentrate on the use of such a system for localization of a walking person which could help firefighters and other emergency responders as well as soldiers and special forces.

Apart from that, indoor pedestrian tracking based on IMU is also useful in mobile Augmented Reality. Three possible tracking approaches could be:

1. **Solely IMU:** When AR overlays don't need to be pixel-accurate (e.g. description of POIs in an interactive tourist map).

2. **Correcting IMU with visual tracking:** Estimate the user’s position based on IMU data and correct the drift with image registration if more accurate augmentations or long-term stable positioning are required [10][14].
3. **Aiding visual tracking with IMU data:** Accelerate computer vision by limiting the visual search space (e.g. a database of known objects and features) according to the approximate location. An example would be the augmentation of a huge trade show: Instead of traversing through the whole database containing all objects of every booth, the approximate current location is derived from IMU data. Then, the object recognition can be narrowed down to only a few booths and their objects.

Paper structure: In Section 2 we will discuss several localization techniques. In Section 3, we give an overview about IMU-based tracking: We introduce the basic IMU localization pipeline (3.1), discuss the state-space model for filtering (3.2) and introduce Bayesian filtering (3.3). In the end of this section, we explain IMU drift correction (3.4) and report on the performance of IMU tracking approaches (3.5). The paper finishes with some additional aiding techniques (Section 4) and the conclusion (Section 5).

2 Localization techniques

A good overview of various localization techniques can be found in [13, Table 1]. We will organize our discussion by distinguishing between two subtypes: beacon-based localization and infrastructure-less localization.

2.1 Beacon-based

Beacon-based techniques rely on an infrastructure of active or passive landmarks, so-called beacons. The most popular representative of this category is GPS: By measuring the signal propagation delay to four or more satellites (these are the beacons), a GPS receiver can calculate its position. Similarly, position information can be derived by measuring the signal strength of several WLAN stations with known position [18] or using ultrasonic pulses [11]. An example for an approach with passive beacons would be marker-based visual tracking: visual patterns that are easily detectable by a computer vision algorithm are placed at known locations allowing to track/localize with a camera.

While beacon-based systems mostly have a good accuracy, they require a previous installation of infrastructure which is costly and not feasible in uncontrolled or unknown environments. Plus, the existence of active radiation sources can result in security problems (jamming and inferences) in critical applications such as military navigation [12].

2.2 Infrastructure-less

An infrastructure-less approach relying solely on the devices attached to the object to be localized often has a reduced accuracy or a higher computational

complexity compared to beacon-based approaches. However, it is more independent from the environment and does not require a previous installation of infrastructure.

Common optical solutions are often based on computer vision with a reference database. This is indeed infrastructure-less but requires knowledge about the environment's structure and thus does not work in unknown areas. However, there are approaches like SLAM (Simultaneous Localization And Mapping) which work without any prior knowledge of the environment.

Another technique for localization is RADAR which detects so-called landmarks which are assumed to be stationary. By observing the relative speed to those landmarks, heading and speed estimations can be made [4].

The technique we will concentrate on is pedestrian localization using IMUs.

3 IMU-based tracking

When an IMU is attached to an object, gyroscopic data, acceleration data and magnetometer data can be used to estimate orientation and position. The whole process chain is discussed in this chapter. Though publications like [12] state that IMUs are immune to jamming, magnetic disturbances e.g. from ferromagnetic materials have a high impact on performance [3].

3.1 IMU measurements

An IMU combines a gyroscope (measuring angular velocity) and an accelerometer with three degrees of freedom each. In our work, we will focus on those inertial sensors and omit the IMU's additional magnetometer.

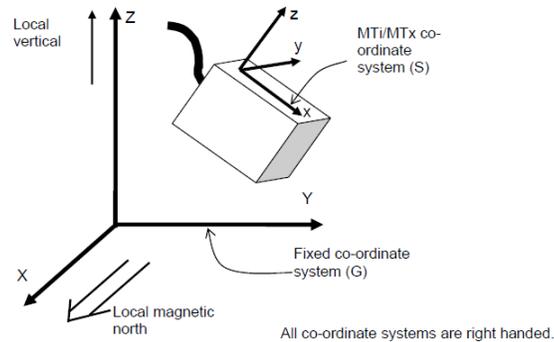


Fig. 1. Global coordinate system and sensor coordinate system. Taken from the Xsens MTi/MTx User Manual

As already mentioned, the device's position can be computed by double integration of the acceleration after removing the gravitational component. Since the

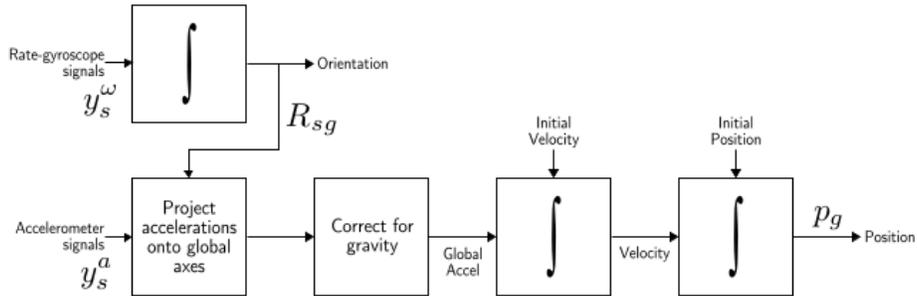


Fig. 2. Simple pipeline for IMU localization. The approximation of orientation introduces inaccuracies which are later amplified due to the double integration. Taken from [19] and annotated with the symbols used in Equations (1) and (2)

gravity correction requires knowledge about the IMU’s orientation in the global coordinate system (cf. Figure 1), the angular velocities are integrated once to approximate the orientation (cf. Figure 2).

This measurement model is expressed in Equations (1) and (2). A measurement is denoted y and the measurement noise (which is typically modeled as Gaussian) e . The subscript s refers to the sensor’s coordinate system and g to the global system. The superscript ω denotes angular velocity and a acceleration. The symbol a_g refers to the device’s acceleration and g_g is the gravitational acceleration: $\|g_g\| \approx 9.81 \frac{m}{s^2}$. The transformation between sensor and global coordinate system derived from the angular velocities is denoted by R_{sg} .

$$y_s^a = R_{sg}(a_g - g_g) + e_s^a \quad (1)$$

$$y_s^\omega = \omega_s^{gs} + e_s^\omega \quad (2)$$

Due to noise and sensor bias, both orientation and position estimates exhibit a strong drifting behavior (2005’s standard IMU: 100 degrees per hour [7], 2007’s military-grade IMU: 5 degrees per hour [12]). The orientation’s inaccuracy worsens the position estimate since it leads to a flawed gravity correction and those errors are amplified by the double integration. This means that a simple pipeline like the one discussed in Figure 2 will provide bad results.

Approaches which try to compensate for those problems are presented in the next sections, since IMUs still have several advantages: the low cost per unit, their little intrusiveness and their self-reliance (cf. Section 2).

3.2 State-space model for recursive filtering

To estimate a state based on a series of noisy measurements, one can use a recursive filter. Those typically require the estimation problem to be expressed in terms of a state-space model consisting of three components: (1) the state (2) the dynamic model, describing the unit’s state changes, and (3) the measurement model, describing how measured values relate to the state [4, Section 4.3.2]. Both

models are usually expressed as stochastic equations, i.e. they include a noise term denoted process noise and measurement noise, respectively.

The following sections are based on [4, Example 2.1] and [2, Chapter 3.4.1].

3.2.1 The state. The state x_t at time t of an inertial navigation system is most commonly structured like this:

$$x_t = \left(p_t^T \ v_t^T \ a_t^T \ q_t^T \ \omega_t^T \right)^T \quad (3)$$

with p being the position, v the velocity, a the acceleration, q the orientation expressed with quaternions and ω the angular velocity.

3.2.2 Dynamic Model. The dynamic model describes how the unit's states changes. The following model is based on the assumption of constant angular velocity ω and constant acceleration a between two timesteps. This assumption is feasible since the IMU sampling rate is quite high (usually around 100 Hz).

At time $t + T$ with T being the sampling time, the orientation q_{t+T} can be calculated iteratively by integrating the angular velocity once:

$$q_{t+T} \approx q_t + T \frac{1}{2} S'(q_t) \omega_t \ . \quad (4)$$

The term $\frac{1}{2} S'(q_t) \omega_t$ approximates how changes in angular velocity translate into quaternions [4, Appendix A4]. The velocity v can be calculated by integrating the acceleration once and the position p by integrating the acceleration twice:

$$v_{t+T} = v_t + T v_t \quad (5)$$

$$p_{t+T} = p_t + T v_t + \frac{T^2}{2} a_t \ . \quad (6)$$

When including the process noise w which accounts for possible violations of the assumption of constant a and ω between two timesteps, the whole dynamic model is represented with equation (7) which is an adapted version of [4, Equation 2.11] and [2, Equation 3.13]. I is the 3x3 identity matrix.

$$\underbrace{\begin{pmatrix} p_{t+T} \\ v_{t+T} \\ a_{t+T} \\ q_{t+T} \\ \omega_{t+T} \end{pmatrix}}_{x_{t+T}} = \begin{pmatrix} I & T I & \frac{T^2}{2} I & 0 & 0 \\ 0 & I & T I & 0 & 0 \\ 0 & 0 & I & 0 & 0 \\ 0 & 0 & 0 & I & \frac{T}{2} S'(q_t) \\ 0 & 0 & 0 & 0 & I \end{pmatrix} \underbrace{\begin{pmatrix} p_t \\ v_t \\ a_t \\ q_t \\ \omega_t \end{pmatrix}}_{x_t} + \underbrace{\begin{pmatrix} \frac{T^2}{2} I & 0 \\ T I & 0 \\ I & 0 \\ 0 & \frac{T}{2} S'(q_t) \\ 0 & I \end{pmatrix}}_{\text{Process noise}} \underbrace{\begin{pmatrix} w_a \\ w_\omega \end{pmatrix}}_{\text{Process noise}} \quad (7)$$

3.2.3 Measurement Model The measurement model describes the relationship between measurements and the state. It obviously depends on the available input types - in our case this is a and ω - but the integration of other inputs (e.g. position from GPS) is straightforward.

Equation (8) is just a time-dependent version of equations (1) and (2). The renaming (*) just illustrates the discussed problem of the orientation approximation q for gravity correction.

$$\begin{pmatrix} y_t^a \\ y_t^\omega \end{pmatrix} = \begin{pmatrix} R_{sg,t}(a_{g,t} - gg) + e_s^a \\ \omega_{s,t}^{gs} + e_s^\omega \end{pmatrix} \stackrel{(*)}{=} \begin{pmatrix} R_{q,t}(a_{g,t} - gg) + e_s^a \\ \omega_{s,t}^{gs} + e_s^\omega \end{pmatrix} \quad (8)$$

3.3 Recursive Bayesian filtering

Recursively estimating the state of a dynamic system based on noisy measurements can be done with filtering. A popular filter is the Bayes Filter with its so-called *Bayes update rule* which is explained in Appendix A.

Using Bayesian filtering for IMU localization is straightforward since the models discussed in Section 3.2 can be easily included in this update rule (also illustrated in Appendix A).

3.3.1 Extended Kalman Filter. The Extended Kalman Filter (EKF) is one implementation of the Bayes filter and is briefly explained in Appendix B. It can be applied if all probabilities are assumed Gaussian.

Since the original Kalman Filter can only cope with linear models, the EKF approximates nonlinear models with a first order Taylor approximation. This produces acceptable results if the overall uncertainty is low and the models don't exhibit high local non-linearity, as it is usually the case for IMU tracking.

3.3.2 Particle Filter. Another type of Bayesian Filters is the Particle Filter which uses Monte Carlo approximations. Its application for IMU localization as an aiding technique is explained in Section 4.1.

3.4 Drift correction

As previously mentioned, a major problem of IMUs is their drifting behavior. In this chapter, the so-called Zero Velocity Updates (ZUPT) are presented which allow drift correction by providing virtual measurements of zero velocity for the measurement model.

3.4.1 IMU positioning. Since IMUs are often placed at the origin of the reference system belonging to the object to be tracked (e.g. IMU placement for body tracking relates to the biomechanical model), the first idea would be to attach them at the waist or the head. However, the attachment on the pedestrian's foot (Figure 3) is advantageous since it allows for an important improvement: Zero Velocity Updates (Section 3.4.3).

The author of [4, Section 4.2.] also discusses various types of placement on the shoe: some may work for normal walking but produce bad results for running or crawling. This becomes important when considering firefighters or soldiers which often perform many different movement patterns aside from normal walking. The author showed that attaching the IMU close to the toes generally yields the best results. However, most approaches we discuss work with a sensor attached next to the shoelaces (cf. Figure 3) which is also feasible when the user does not run.

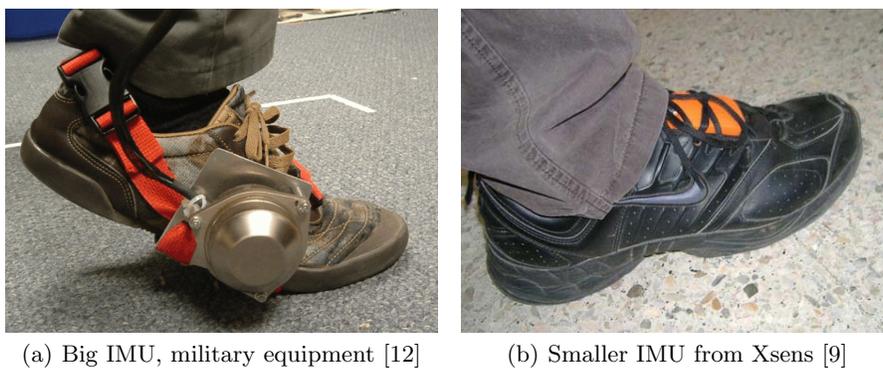


Fig. 3. Examples of used IMUs and their attachment to the user's shoe. Several authors plan the integration into the sole.

3.4.2 Step detection. Zero Velocity Updates (ZUPT) are a way to compensate for the IMU's drifting behavior and thus increase the accuracy of position tracking. The goal is to find a position where most of the IMU's true motion is known and to use the difference to the measured values for drift correction. This holds for the time when the foot is in contact to the ground during the walking cycle: all three accelerations and velocities of the foot are assumed to be zero (cf. Figure 4) which means that the norm of the IMU's acceleration should be equal to the gravitational acceleration and the angular velocities should be zero. This point in time is called stand still.

Hence, two subtasks have to be solved: (1) detecting the stand still and (2) integrating the observed difference between measurement and ground-truth into the model.

The former problem could be solved with a pressure sensor at the shoe's sole, but this would introduce a second sensor and would fail during movement patterns without contact between sole and ground, e.g. crawling [4]. We therefore discuss how to perform the whole ZUPT procedure with only IMU data available.

In the following, we will take a look on two different approaches for stand still detection: a threshold comparison and a probabilistic framework.

Thresholding [12]: The goal is to determine one arbitrary point in time T_s between midstance T_1 and terminal stance T_2 (cf. Figure 4) with velocity equal

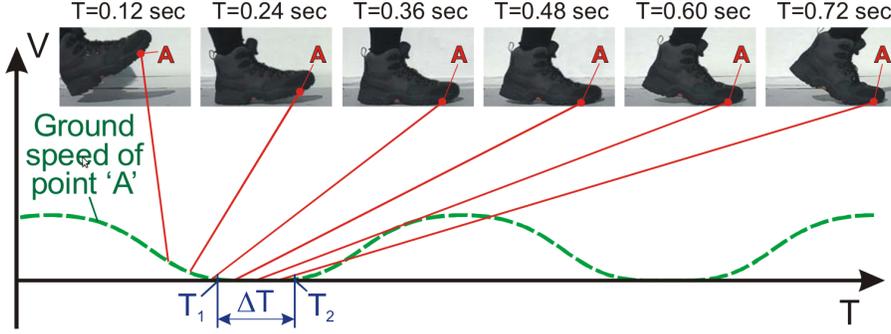


Fig. 4. Walking phases. Velocity and acceleration of A are zero during ΔT . T_1 is called midstance and T_2 terminal stance [1] (taken from [12]).

to zero. This is done by sampling segments of 0.5 seconds with 100 data points, computing the gyroscope's amplitude

$$w_i := \sqrt{w_{x,i}^2 + w_{y,i}^2 + w_{z,i}^2} \quad (9)$$

and then searching for the smallest w_i below a certain threshold. The time of this w_i is the wanted point T_s . As one can see, the correct identification of midstance or terminal stance is not needed for this approach since a minimum of w_i is assumed to occur between T_1 and T_2 .

Probabilistic framework [6]: A Hidden Markov Model is used to estimate the probabilities for standing still or motion. It has two hidden states (stand still and movement), a predefined state transition matrix

$$\Pi = \begin{bmatrix} 0.95 & 0.05 \\ 0.05 & 0.95 \end{bmatrix} \quad (10)$$

and an angular velocity test statistic for the observations

$$T_t^\omega = \frac{\|y_t^\omega\|^2}{\sigma_\omega^2} \quad (11)$$

with y^ω being the measured angular velocity and the variance σ_ω^2 being used as a noise parameter.

The authors of [6] claim that this probabilistic approach is more reliable than a simple threshold comparison. It reduces the rate of false positives (typically, stand stills can wrongly be found when the foot is in mid-air) which is crucial since they have a much worse impact on the performance than false negatives: A false reset with wrong sensor values is worse than a missed reset.

3.4.3 Zero Velocity Updates When a stand still is detected, zero velocity can be included into the model as virtual measurement [4][13][7].

The measurement model introduced in Section 3.2.3 can be extended with $0 = v_{g,t} + e_s^v$ (e_s^v representing the error in the ZUPT algorithm) whenever a stand still is detected:

$$\begin{pmatrix} y_t^a \\ y_t^\omega \\ 0 \end{pmatrix} = \begin{pmatrix} R_{sg,t}(a_{g,t} - g_g) + e_s^a \\ \omega_{s,t}^{gs} + e_s^\omega \\ v_{g,t} + e_g^v \end{pmatrix} \quad (12)$$

3.5 Evaluation of EKF-based approaches

Many publications use EKF and report feasible performance when using ZUPT for drift correction [7][13][4]. In a closed-loop experiment, [7] reports a total position drift of 0.3% of the path’s length (length: 118,5 m, time: 322 seconds) using the InertiaCube3 IMU. A closed-loop path of 194 m in around 170 seconds was traversed with a positioning error of around 4.4% of the path’s length by [13], using a Xsens MTi IMU.

Since the results highly depend on the used hardware and the filter parameters, the given numbers are not directly comparable. However, they clarify that the error grows not cubically over time as in pure inertial navigation but only linearly.

4 Aiding techniques

The IMU-based tracking can be improved further by integration of additional sensor information, for example GPS [7]. They use GPS measurements to correct the heading estimation.

A cooperative localization approach is presented in [13]: several agents constantly exchange position estimates together with a radio-based distance measurement.

Another approach is introduced in [18]. They utilize the discussed IMU-based tracking as a black box virtual sensor. Its position “measurements” are used together with a map of the environment as inputs for a Particle Filter (cf. Figure 5(a)). In the following, we will give a brief overview of this system.

4.1 Monte Carlo Localization in a known map

The Particle Filter represents arbitrary probability distributions via Monte Carlo approximations. This is why localization with Particle Filters is also called Monte Carlo Localization. A rough explanation of Particle Filters can be found in Appendix C. In summary, Particle Filter are advantageous over the Extended Kalman Filter when the probabilities are not Gaussian, the models are highly nonlinear or multiple hypothesis have to be tracked.

In the approach of [18], the map of the building to localize the pedestrian in is represented as a collection of planar floor polygons defining the areas where the pedestrian is able to stand. If an edge of a polygon does not have two adjacent polygons, it represents an impassable wall.

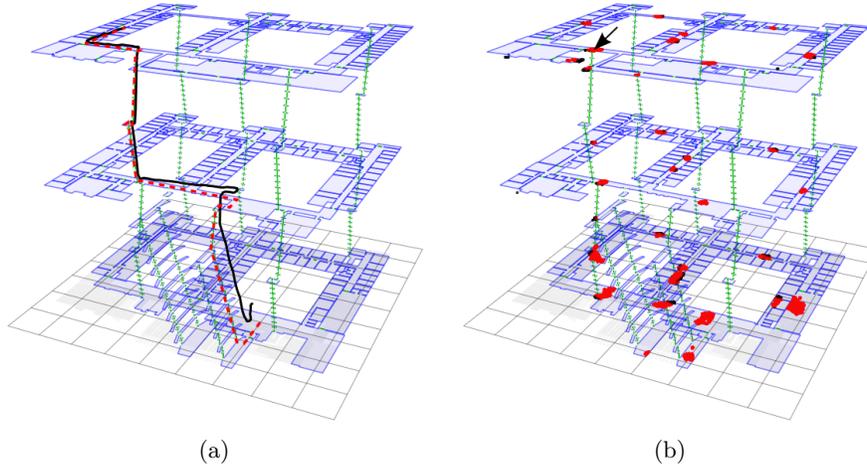


Fig. 5. IMU-based tracking with a Particle Filter and WiFi aiding in a three-storey building. (a) The pedestrians true path: dotted red line. The path tracked by the system: solid black line. (b) The particles representing the position guesses are displayed in red and the true position is indicated by the arrow (from [18]).

The **state** of a particle is

$$x_t = (p_t, \theta_t, poly_t) \quad (13)$$

with p being the position, θ the heading and $poly$ the floor polygon the particle is assigned to (cf. Figure 5(b)). The control input for the **dynamic model** consists of the horizontal position change Δp and heading change $\Delta\theta$ obtained from the black box IMU localization:

$$\theta_t = \theta_{t-1} + \Delta\theta \quad (14)$$

$$p_t = p_{t-1} + \begin{pmatrix} \Delta p \cos \theta_t \\ \Delta p \sin \theta_t \end{pmatrix}. \quad (15)$$

The **measurement model**, providing the particle weight, is

$$w_t = \begin{cases} 0 & , \text{ if the particle crossed a wall} \\ \mathcal{N}_{0,0.1}(\Delta z - \Delta z_{poly}) & \text{ else} \end{cases} \quad (16)$$

where $\mathcal{N}_{0,0.1}$ is the normal distribution with an empirically chosen variance of 0.1 m, Δz is the height change obtained from the black box IMU localization and Δz_{poly} is the height difference between the previous and the current polygon state. Since the particles move according to the dynamic model, their assigned floor polygons have to be adjusted constantly. This is done by intersecting the

vector resulting from the change in position with the edges of the assigned polygon.

1. If no intersection is found, the polygon stays the same.
2. If the vector intersects a wall, w_t is set to zero.
3. If the vector intersects with an edge connected to another polygon, this other polygon is assigned to the particle. The remaining part of the vector is intersected with the new polygon's edges.

4.2 Initialization with WiFi localization

To increase the accuracy by providing a feasible initialization (instead of just randomly distributing particles over the whole map in the beginning) and to resolve problems with symmetry in the environment, [18] also include WiFi localization. Their approach has two steps:

4.2.1 Offline WiFi cell decomposition: First, the map is subdivided into cells having no edge longer than 8 m. Every available WiFi access point throughout the building is attributed a set of cells in which it is visible.

4.2.2 Online WiFi localization: Before initializing the Particle Filter, the system collects the visible WiFi access points having a signal strength over a certain threshold. Then, all map cells are picked which occur in every set of those access points. For initializing the Particle Filter, the particles are randomly distributed mainly over those cells which according to the authors reduces the number of needed particles for the given building to less than 1/30.

4.3 Evaluation

The authors found that their map-aided, WiFi-initialized IMU tracking method provides position estimates with an accuracy of 0.73 m 95% of the time and 0.5 m 75% of the time. It appears that the positioning error is bounded and the drifting over time mostly eliminated, since the stated accuracy even holds in a 16 minute trace.

5 Conclusion

IMUs have several advantages for pedestrian localization, mainly their self-reliance, but require additional processing and aiding measurements to achieve long-term accuracy. We presented a state-space model which is usable for recursive filtering of IMU data and discussed several publications applying this method. In order to increase the accuracy further, we presented Zero-Velocity Updates as a way of constant automatic recalibration. In the end, aiding techniques like the use of a map of the environment and an additional Particle Filter were presented.

Many approaches showed feasible accuracy, though the evaluations are tough to compare since the used hardware, the test paths, the walking speed and several other factors differ. A very detailed evaluation was done by [12] but the used localization technique was not clearly defined (given the vague reference to [16], they probably used Kalman filtering) and was therefore excluded from our discussion.

As [18] already pointed out in their further work paragraph, it would be interesting to know how their map-aided system behaves when WiFi is not only used for initial localization but throughout the whole process. A way to use convex optimization for identification of good magnetometer data for robust heading estimation is discussed in [5] and could be integrated in the discussed state-space model.

The main problem remaining is the hardware setup: an IMU has to be attached to the user and has to be connected to a mobile computer. While some authors propose the integration of an IMU in the shoe sole, approaches adapted for smartphones may be interesting as well, since the hardware setup is very simple and the devices are widely spread. Instead of ZUPT, another way of drift correction would be needed for smartphones, e.g. correction by vision-based tracking.

Appendix

A Derivation of Bayes update rule

We will now derive the formula for calculating the so-called posterior or belief $Bel_t(x_t)$ for a state x_t based on [15]. The result is also known as *Bayes update rule* since it is based on the Bayes theorem

$$p(x|y, z) = \frac{p(y|x, z)p(x|z)}{p(y|z)} =: \eta p(y|x, z)p(x|z) . \quad (17)$$

In the following equation, x_t is a state, z_t a (noisy) measurement and u_t a control input at time t . $Bel_t(x_t)$ is the estimated probability of a state x at the time t .

$$\begin{aligned} Bel_t(x_t) &= p(x_t|z_{0:t}, u_{0:t-1}) \\ &\stackrel{Eq.(17)}{=} \frac{p(z_t|z_{0:t-1}, u_{0:t-1}, x_t) p(x_t|z_{0:t-1}, u_{0:t-1})}{p(z_t|z_{0:t-1}, u_{0:t-1})} \\ &\stackrel{Eq.(17)}{=} \eta_t p(z_t|z_{0:t-1}, u_{0:t-1}, x_t) p(x_t|z_{0:t-1}, u_{0:t-1}) \\ &\stackrel{Markov}{=} \eta_t p(z_t|x_t) p(x_t|z_{0:t-1}, u_{0:t-1}) \\ &\stackrel{(*)}{=} \eta_t p(z_t|x_t) \int p(x_t|z_{0:t-1}, u_{0:t-1}, x_{t-1}) p(x_{t-1}|z_{0:t-1}, u_{0:t-1}) dx_{t-1} \\ &\stackrel{Markov}{=} \eta_t \underbrace{p(z_t|x_t)}_{\text{Measurement}} \int \underbrace{p(x_t|u_{t-1}, x_{t-1})}_{\text{Dynamics}} Bel_{t-1}(x_{t-1}) dx_{t-1} \end{aligned} \quad (18)$$

(*) Here, the continuous marginalization rule respective its specialization “Law of total probability” is used:

$$p(x) = \int_y p(x|y) p(y) dy \quad (19)$$

Since we assume that the Markov assumption holds (i.e. a state can be derived knowing only its directly previous state) the Bayes update rule is also called Markov localization.

B The Extended Kalman Filter equations

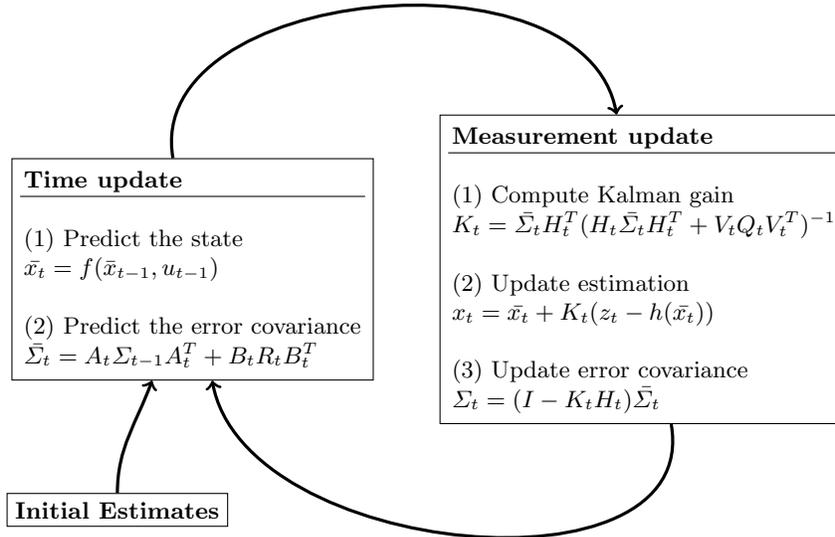


Fig. 6. The Extended Kalman Filter steps and their equations. Adapted from [17]

A detailed explanation of the formulas in Figure 6 is beyond the scope of this paper. Just a few remarks:

1. The functions $f(\cdot)$ and $h(\cdot)$ are nonlinear.
2. The matrices A_t and B_t are the Jacobian matrices used for linearization (Taylor approximation) of the dynamic model while R_t is the process noise covariance.
3. The Jacobian matrices for linearization of the measurement model are denoted by H_t and V_t . The measurement noise covariance is Q_t .

C Particle Filters

One major constraint of the EKF is the limitation to probability density functions (PDF) which are representable by Gaussians. To represent arbitrary PDFs, another Bayesian filtering approach has to be used. The Particle Filter uses a Monte Carlo approximation for this, i.e. the PDF is represented by weighted samples, called particles (cf. Figure 7). The Bayesian update rule (Equation 18) is implemented through an update of the particles.



Fig. 7. Position estimates as particles during localization of a moving robot [15].

The Particle Filter consists of three steps [8]:

1. **Drift:** The particles are copied according to their weights. Then, all particles get the same weight and are moved according to the dynamic model.
2. **Diffuse:** Noise is applied to spread the particles.
3. **Measure:** Evaluate the measurement model for each particle and represent this probability distribution with a new set of weighted particles.

References

1. Ed Ayyappa. Normal Human Locomotion, Part 1. *JPO Journal of Prosthetics and Orthotics*, 9(1):10–17, January 1997.
2. Gabriele Bleser. *Towards Visual-Inertial SLAM for Mobile Augmented Reality*. PhD thesis, Technical University Kaiserslautern, 2009.
3. Gabriele Bleser, Gustaf Hendeby, and Markus Miezel. Using egocentric vision to achieve robust inertial body tracking under magnetic disturbances. In *2011 10th IEEE International Symposium on Mixed and Augmented Reality*, pages 103–109. IEEE, October 2011.
4. Jonas Callmer. *Topics in Localization and Mapping*. Licentiate thesis, Linköping University, 2011.
5. Jonas Callmer and Fredrik Gustafsson. Robust Heading Estimation Indoors. *IEEE Transactions on Signal Processing*, 2013.
6. Jonas Callmer, David Tornqvist, and Fredrik Gustafsson. Probabilistic Stand Still Detection using Foot Mounted IMU. In *Information Fusion (FUSION), 2010 13th Conference on*, pages 1–7, 2010.
7. Eric Foxlin. Pedestrian Tracking with Shoe-Mounted Inertial Sensors. *IEEE Computer Graphics and Applications*, 25(6):38–46, November 2005.

8. Michael Isard and Andrew Blake. CONDENSATION - conditional density propagation for visual tracking. *International Journal of Computer Vision*, 29(1):5–28, 1998.
9. A.R. Jimenez, F Seco, C Prieto, and J Guevara. A comparison of Pedestrian Dead-Reckoning algorithms using a low-cost MEMS IMU. In *2009 IEEE International Symposium on Intelligent Signal Processing*, pages 37–42. IEEE, August 2009.
10. Masakatsu Kourogi and Takeshi Kuratta. A Wearable Augmented Reality System with Personal Positioning based on Walking Locomotion Analysis. In *ISMAR '03 Proceedings of the 2nd IEEE/ACM International Symposium on Mixed and Augmented Reality*, page 342, 2003.
11. J. Newman, D. Ingram, and A. Hopper. Augmented reality in a wide area sentient environment. In *Proceedings IEEE and ACM International Symposium on Augmented Reality*, pages 77–86. IEEE Comput. Soc, 2001.
12. Lauro Ojeda and Johann Borenstein. Non-GPS Navigation for Security Personnel and First Responders. *Journal of Navigation*, 60(03):391–407, August 2007.
13. Jouni Rantakokko, Joakim Rydell, Peter Strömbäck, Peter Händel, Jonas Callmer, David Törnqvist, Fredrik Gustafsson, Magnus Jobs, and Mathias Grudén. Accurate and reliable soldier and first responder indoor positioning: multisensor systems and cooperative localization. *IEEE Wireless Communications*, 18(2):10–18, April 2011.
14. Gerhard Schall, Daniel Wagner, Gerhard Reitmayr, Elise Taichmann, Manfred Wieser, Dieter Schmalstieg, and Bernhard Hofmann-Wellenhof. Global pose estimation using multi-sensor fusion for outdoor Augmented Reality. In *2009 8th IEEE International Symposium on Mixed and Augmented Reality*, pages 153–162. Ieee, October 2009.
15. Sebastian Thrun. Probabilistic Algorithms in Robotics. *AI Magazine*, 21(April):93–109, 2000.
16. D Titterton, J L Weston, and Institution of Electrical Engineers. *Strapdown Inertial Navigation Technology, 2nd Edition*. IEE radar, sonar, navigation, and avionics series. Institution of Engineering and Technology, 2004.
17. Greg Welch and Gary Bishop. An Introduction to the Kalman Filter. Technical report, University of North Carolina at Chapel Hill, 1995.
18. Oliver Woodman and Robert Harle. Pedestrian localisation for indoor environments. In *Proceedings of the 10th international conference on Ubiquitous computing - UbiComp '08*, pages 114–123, New York, New York, USA, 2008. ACM Press.
19. Oliver J Woodman. An introduction to inertial navigation. Technical Report UCAM-CLTR-696. Technical Report 696, University of Cambridge, Computer Laboratory, 2007.