2012 International Conference on Indoor Positioning and Indoor Navigation, 13-15th November 2012

A Particle Filter Approach to Indoor Navigation Using a Foot Mounted Inertial Navigation System and Heuristic Heading Information

James Pinchin
Horizon Digital Economy Research
University of Nottingham
Nottingham, UK
james.pinchin@nottingham.ac.uk

Chris Hide
Nottingham Geospatial Institute
University of Nottingham
Nottingham, UK
chris.hide@nottingham.ac.uk

Terry Moore
Nottingham Geospatial Institute
University of Nottingham
Nottingham, UK
terry.moore@nottingham.ac.uk

Abstract—**Foot mounted inertial navigation is an effective method for obtaining high quality pedestrian navigation solutions from MEMS sensors. Zero-Velocity information from stationary periods in the step-cycle can be used to regularly correct position drift and update estimates of the inertial sensor biases, hence dramatically improving the navigation solution.**

However the causes of heading error remain poorly observable and so foot mounted inertial navigation suffers from considerable drift over time. To address this problem the authors previously developed Cardinal Heading Aided Inertial Navigation (CHAIN). CHAIN makes use of the fact that when in a building, obstacles such as corridors and furniture constrain pedestrians to move in one of four directions parallel to the outside walls of the building. This knowledge is then appropriately weighted and used in an Extended Kalman Filter to improve error estimation.

Although the CHAIN method is very effective at improving the quality of the heading estimates, position errors still accumulate with time, and threshold tests are required to cope with periods of motion away from the cardinal headings. In this work we investigate the use of a building floor plan to further aid navigation. This is achieved using a particle filter approach whereby particles which cross walls are removed and those which navigate in open spaces are allowed to continue. Previously the particle filter approach has been computationally intensive process requiring many particles to effectively model the navigation errors.

In our work we recognise that heading is the primary source of navigation error and therefore incorporate heuristic heading information into the particle filter design. By weighting particles according to their heading we reduce the number of particles required to maintain a small failure rate and improve system performance in more open areas where there are few mapped walls to aid navigation.

This paper will describe the design of our particle filter and the heuristic heading approach. Results from a number of representative test walks using a MEMS IMU will be used to demonstrate the system performance. The use of CHAIN is shown to be capable of significantly reducing the filter failure rate from 44% to 14% when a small number of particles is used in the filter (250) and the initial position is poorly known.

Index Terms—**INS,Pedestrian,Particle Filter,Cardinal Heading**

I. Introduction

In 2005 the first work on foot mounted inertial navigation was reported [1]. This work showed that by using observations of zero velocity (ZUPTs) the navigation performance of a low cost MEMS IMU could be made to approach that of a much higher cost unit.

In a foot mounted INS zero velocity can be assumed each time the foot is in contact with the ground. This situation can be detected either using sensors independent of the IMU or by examination of the IMU output itself. A review of ZUPT detection methods can be found in [2].

However two key problems with foot mounted INS remain. Firstly the INS heading errors are poorly observable using ZUPTs. Secondly for stable filter performance, especially when using an Extended Kalman Filter (EKF), the INS must be initialised with an accurate position and attitude. The problem of initialisation is discussed further in [3].

In previous work the authors have proposed heuristic methods to solve the problem of heading error observability. The Cardinal Heading Aided Inertial Navigation (CHAIN) method uses heuristic knowledge of a building layout to obtain observations of the INS heading error [4]. This approach is discussed in more detail in section II of this paper. When used in an EKF the CHAIN observations have been shown to be capable of controlling heading drift over long periods of navigation, keeping position error below 5m over one hour [4]. The remaining drift results from small errors in the distance traveled estimate from the INS combined with residual heading errors.

Methods for relaxing the INS initialisation requirements have been proposed in the past. These include extending the INS states to include a large heading error term ([5]) and using an Unscented Kalman filter (UKF) ([6]). The authors have also proposed a technique for trajectory matching when using GNSS position updates ([3]). However all of these methods require multiple position observations before either the filter converges or the INS can be re-initialised, position observations of sufficient accuracy are unlikely to be available when indoors or near buildings.

In this work we use a Particle Filter to address the problems of both initialisation and position drift. We choose a Particle

978-1-4673-1954-6/12/$31.00 @2012 IEEE

Filters in order to allow *a priori* map data to be used in the navigation algorithm.

Previous work has shown that using map information to control the position drift of the INS allows high precision navigation to be maintained indefinitely if sufficient map constraints exist (e.g. [7]). By incorporating CHAIN into the particle filter we reduce the requirement for closely spaced map constraints and improve filter performance when navigating spaces with unmapped constraints on the pedestrians motion. It should be noted that the CHAIN method may also be used when complete mapping is unavailable. It is therefore suitable for use when navigating in areas with partial maps or with building footprints only.

Map data may also be used to solve the initialisation problem by using path constraints to converge from an unknown position to a uni-modal position solution. This method of initialisation often requires large number of particles in the filter and may be impractical for real time applications on single-board computers. For this reason we investigate a more realistic scenario whereby the initial position and orientation is roughly known and only requires refinement.

II. Background

In this section we introduce the principles of Cardinal Heading Aided Inertial Navigation (CHAIN). We also briefly describe the Extended Kalman Filter implementation and its use in forming the input to our particle filter. The principles of foot mounted inertial navigation have been well described elsewhere and are not reproduced here (e.g [1], [8]).

A. CHAIN

Cardinal Heading Aided Inertial Navigation, CHAIN, has previously been introduced by the authors as a method for identifying and correcting heading drift in a foot mounted INS (e.g [9], [10]).

It makes use of the observation that in many pedestrian environments, particularly indoors, a user is constrained to move in one of four directions termed the *cardinal headings*. This constraint may be put upon the pedestrian by corridors, furniture layout or simply desire lines linking points of interest. When indoors the cardinal headings usually relate to the directions of the outside walls of the building, in this case cardinal headings can be simply extracted from aerial imagery or building footprints.

CHAIN has been used in an Extended Kalman Filter in the form of an observation of heading error. The assumption is made that error in the INS 'course over ground' (COG) over one step is identical to the error in the INS heading state. A measurement of this error is obtained by differencing the INS COG with the cardinal headings. The minimum difference is taken as a measurement of heading error and weighted before being used as a filter input.

Weighting of the measurement is performed to reflect the degree to which the CHAIN assumption holds in a particular environment. In a tightly constrained environments (e.g. manhattan grid of corridors) the measurements are highly weighted while in a more loosely constrained environment (e.g. a car park) the observations can be given a lower weight.

Since heading is the primary source of position error in a foot mounted INS aided by zero-velocity observations, CHAIN is very effective at controlling position accuracy drift. Tests have demonstrated its ability to keep position error below 5m over a one hour test walk using an otherwise un-aided foot mounted INS [4].

B. Extended Kalman Filter

In this work an Extended Kalman Filter (EKF) is used to estimate the errors in the states of an INS using aiding measurements from zero-velocity observations (ZUPTs).

Extended Kalman Filters used for INS integration employ a linearised error model to describe the temporal evolution of the INS error terms in equation 1. The linearisation is usually achieved by perturbation analysis such as in [11] and [12].

$$x = (\delta p \ \delta v^n \ \delta\omega \ \delta g^b \ \delta a^b)^T \tag{1}$$

where δp is the vector of position errors ; δv^n is the vector of navigation frame velocity errors ; $\delta\omega$ is the vector of navigation frame attitude errors; δg^b is the vector of gyro bias errors and δa^b is the vector of accelerometer bias errors.

In this work the linearised inertial navigation model is the ϕ angle model ([13]). The effect of the linearisation on EKF initialisation is discussed in some more detail in [3]. However for the purposes of the present work this linearisation makes several assumptions including the small angle assumption which requires that errors in the initial attitude terms be small. This is often a challenging condition to fulfill, particularly in the case of heading error.

While many methods have been proposed to overcome the limitations of the small angle approximation, in this work we recognise that absolute position and orientation are unimportant for the current application of the EKF. In this work we simply seek to obtain step length and change in direction values for subsequent use in the particle filter (section III). This relaxation of the EKF output requirements allows us to initialise the EKF with approximate initial values for position and an arbitary heading value. Following [3] we keep the uncertainty in the initialisation parameters small.

III. Particle Filter

In this section the design of a particle filter for enforcing mapping constraints is described in the context of previous work. The filter is designed to take step length and change in direction as inputs. Thus it is suitable for use with other inertial pedestrian navigation systems including those based on step counting if the number of particles is kept high enough to model the uncertainty. At present the filter is designed to operate without feedback to the Extended Kalman Filter or the INS states.

A. Particle Filters for Foot Mounted INS

While much work has been done in the robotics community on localisation using particle filters (e.g. [14], [15]), they have only recently been applied to pedestrian navigation. Three pieces of previous work are considered especially relevant.

A 2008 work [16] investigated the use of partial mapping for indoor navigation. They describe the scenario of a first responder arriving at an incident with knowledge of only the building footprint. This work demonstrates that the use of only minimal constraints in a (backtracking) particle filter can be very useful. A simple weighting model was used in this work, simply eliminating particles which crossed walls. Further performance might have been obtained using a more subtle weighting approach.

In 2010 a thesis was published describing the use of full indoor mapping to aid a foot mounted INS [17]. The proposed particle filter constrained particles by preventing their movement over obstacles such as walls. This work investigated the use of these constraints for an unknown initialisation, using constraints on the path taken to converge to a uni-modal position solution. In this thesis large numbers of particles were used, especially at initialisation, with numbers ranging from $2.5\text{x}10^4$ to $1.4\text{x}10^6$. Adaptive resampling was used to vary the number of particles used depending on the complexity of the PDFs, thus fewer particles were used after initialisation and in the 'tracking' mode. In this work the filter weighting scheme was based upon the agreement of the height change from the INS with the height change obtained from the map data. In this work no attempt was made to investigate cases of incomplete or incorrect mapping.

In 2011 work was published describing the use of an angular probability density function for weighting particles within the particle filter [18]. In this work wall crossing constraints were not applied directly in the filter. Instead particles were weighted according to their direction with respect to an angular movement model, derived from complete mapping of a building. Particles which crossed walls were naturally de-weighted in this model. This work showed that use of weighting based on particles heading performs better than an equal particle weighting approach, especially when dealing with multiple particle groups. However this method assumes closely spaced walls when deriving the movement model. Its effectiveness may be limited in open areas or when using maps with no internal wall constraints.

B. Particle Filter Design

The particle filter uses a set of samples, the 'particles', to represent the probability density functions (PDFs) of a set of states. The PDF may be non-gaussian or multimodal. Weights are used to differentiate between particles, allowing a measure of 'quality' to be assigned. The number of particles used is critical - enough must be present to represent the PDF(s) while too many will make the computational cost of the method excessively high.

The 2.5D localisation filter used in this work is of a commonly used type: the bootstrap filter [19]. The particle states represent 2D position and heading. Each particle also has Room ID and floor number states which are naturally integer. Equations 2 and 3 give the state of a filter with N particles at time t and a state vector for a particle i, x_i.

$$S_t = \{\langle x_t^i, w_t^i\rangle \quad i = 1, \ldots, N\} \tag{2}$$

$$x^i = (\phi_i \ \lambda_i \ f_i \ \theta_i \ Rid_i)^T \tag{3}$$

Where ϕ_i and λ_i are the latitude and longitude of particle i. θ_i is the heading of particle i and f_i,Rid_i are the floor number and room ID of the particle.

The weights, w^i are normalised such that;

$$\sum_{i=1}^{N} w^i = 1 \tag{4}$$

As each new step occurs the particles are propagated according to their previous state and the new measurement. Their weights are adjusted according to a model in the 'update' step and finally they are 'resampled' in the final stage to remove low value particles and replicate high value particles.

We now consider each of these steps in more detail alongside other key aspects of the filter.

C. Mapping

Since the mapping data and the ability to query it has a very large influence on the particle filter we discuss it here in some detail. The purpose of this is not only to describe the process used for this work, but also to illustrate some of the practical difficulties encountered even when supposedly complete mapping is available.

For the purposes of this work a complete and accurate map of the building walls and doors is assumed. While this may not be possible in many real world scenarios, it does allow examination of the particle filter's performance and the effect of the CHAIN method under 'ideal' conditions. The CHAIN method is equally applicable in areas where incomplete mapping is available.

Mapping data for each building at the University of Nottingham has been made available to us. This takes the form of a well maintained dataset describing each building floor by floor. Since the primary purpose of the data is facilities management, the mapping data is not georeferenced. For the purposes of this work georeferencing has had to be undertaken manually for each set of building plans used.

In its raw form the mapping consists of a set of vectors describing each wall in a building. Doors are not mapped and walls are continuous across door openings. The first stage in our process is to separate each building floor into a set of continuous, planar, polygons. Each polygon describes one room.

A room is given a set of properties including but not limited to; Room ID, floor number, cardinal headings and associated doors. The doors were mapped manually and are held in a separate database, each door having an ID, a location and a record of the Room ID's it connects (including a Room ID for outside the building). A polygon is also created which details the building's footprint and any external doors. This allows efficient querying when particles enter the building from outside.

In our implementation a room may also allow or deny elevation changes. Thus a space which has stairs or an elevator will allow a particle to change floor number while a room without that capacity will restrict the particles to one level. Elevation changes are not discussed further in this work.

D. Particle Filter Stages

1) Initialisation: At system startup a fixed number of particles are initialised. Their state vectors are then populated with values drawn from suitable distributions.

In this work we assume that the initial position and heading are not completely unknown but instead is known to a low accuracy. This reflects a startup where the user has self declared their position and orientation or values have been taken from other sensors, eg WiFi and a magnetometer. This reduces the extent of the problem to be solved and reduces the chance of problems of building self symmetry which may be found in a more general localisation problem. Nevertheless sufficient particles must be used to represent the PDFs at this initial stage. The relationship of particle numbers to filter failure rate will be discussed later.

In our implementation particles are initialised with a gaussian distribution around the initial values for position and heading. For the purposes of this work all particles are taken to be on the same floor and their room number is determined by a sequential search of the rooms on that floor. The particles are initialised with equal weights, normalised according to equation 4.

2) Propagation: At each step the state vector of each particle is propagated according to the new step length (l)and change in direction measurement ($\Delta\theta$). For each particle the measurement is perturbed using a model of the measurement uncertainties (equations 5a and 5b).

$$l' = l + \delta l \tag{5a}$$

$$\Delta\theta' = \Delta\theta + \delta\Delta\theta \tag{5b}$$

Values for δl and $\delta\theta$ may come from any multimodal or non-gaussian model. In our case however they are drawn from gaussian distributions with constant variances defined at system initialisation.

The states are then propagated according to

$$\theta_t^i = \theta_{t-1}^i + \Delta\theta' \tag{6a}$$

$$dN = l' \cos(\theta_t^i) \tag{6b}$$

$$dE = l' \sin(\theta_t^i) \tag{6c}$$

The changes in position dN and dE are then converted to changes in latitude and longitude to be added to the particle's state vector. At this stage the Room ID and floor number states remain unchanged.

3) Update: In this step the map constraints and CHAIN method may be applied through the particles weight. A particle whose weight is set to zero is said to be 'killed'. This is because it is impossible for such a particle to survive the resampling step and be re-used in the following iteration. Conversely one whose weight is unchanged is said to 'live'.

- Wall Constraints

The most common way for a particle to be 'killed' is through a wall crossing. If a particle crosses a mapped wall it's weight is by default set to zero. The exception to this is if the wall crossing occurs within a threshold distance of a door. In this case the particle may survive and its Room ID is updated.

The process used in our filter is described in figure 1. Since this process is repeated for each particle at each step, along with the propagation step, our process has been designed to perform the minimum number of operations for each particle. For example only the walls of the current polygon are tested for crossings and only if a wall has a door is the distance of the crossing from the door tested.

The result of this process is a set of particles with weights either unchanged from the last step or set to zero. The process is depicted in figure 2a. In this figure, particles move towards a door with the position error increasing until they are either killed against a wall or allowed to pass through the door. On the far side of the door the size of the position distribution again increases. This is due to both the range of heading values

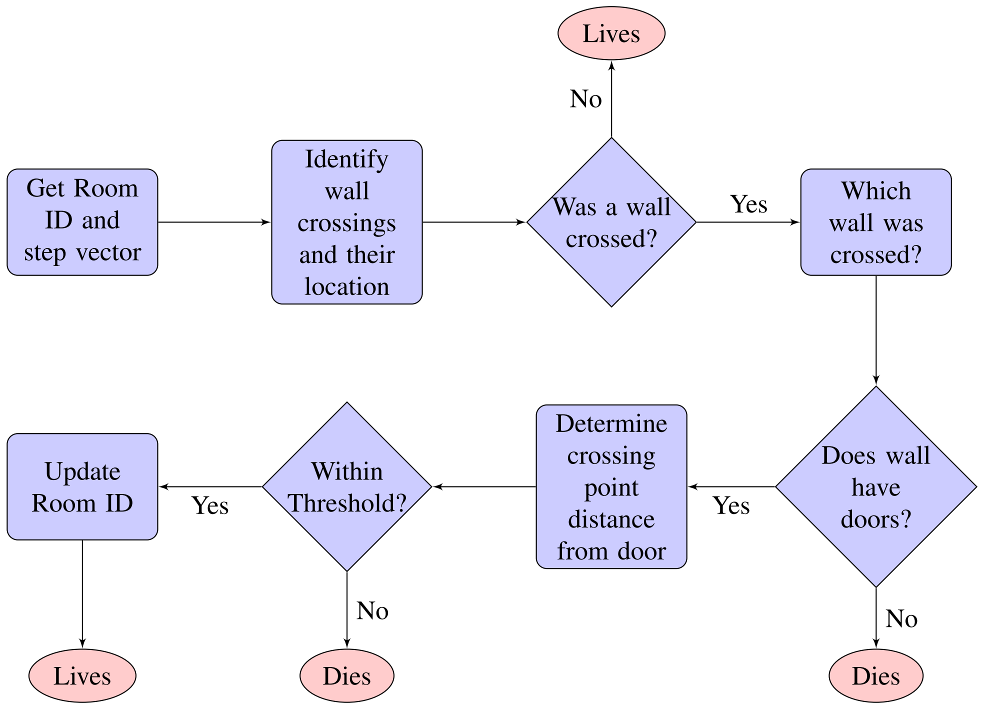


Fig. 1: Flow chart for deciding the fate of a particle at the update step of the particle filter.

allowed to pass through the door and the noise introduced in the propagation steps by the model of the measurement uncertainties (In this case the $\Delta\theta$ variance was 1° per step).

• Other Weighting Methods

Other, more subtle weighing methods may also be applied to the particles at this stage. If position aiding is available from GNSS or another sensor the the particles may be weighted according to their distance from the position update. Alternatively if WiFi fingerprinting is to be used then the particles can be weighted according the the degree to which their position and the measured signal strengths match those in the fingerprint database.

In this work CHAIN is included in this step to further refine the particles weights according to their heading. More details will be given in section IV. In cases where CHAIN or an alternative strategy is not used the surviving particles are simply left with equal, normalised, weights.

4) Resampling: In this work the straightforward multinomial resampling strategy has been adopted [20]. The input to the multinomial resampling algorithm is a set of particles with weights normalised according to equation 4.

The result of multinomial resampling is a set of particles equal in number to the input particles. Particles with zero weight are not resampled while those with higher weights are resampled (duplicated) multiple times to maintain the total number of particles.

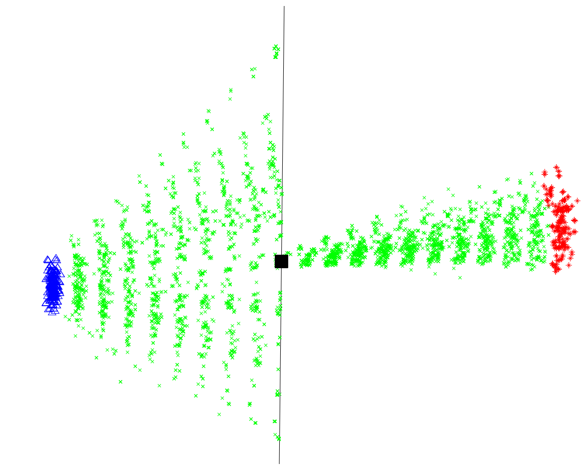
IV. CHAIN IN THE PARTICLE FILTER

The inclusion of the cardinal heading measurement in the particle filter process is straightforward. At the 'update' stage, the weight of each particle is set according to its previous weight plus some value depending on the difference between its heading and the closest cardinal heading. In this way particles such as those which cover a long distance at 45 degrees to one of the cardinal headings are gradually deweighted with respect to other particles which follow the cardinal headings.

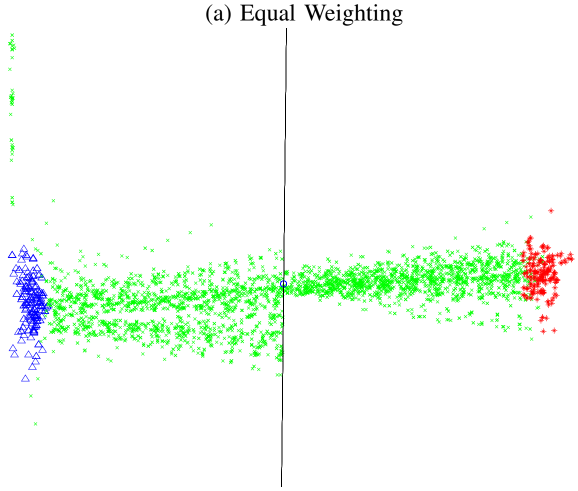
This is especially useful in more open spaces with unmapped features, such as open plan offices or supermarkets with unmapped shelves. If CHAIN was not used, the filter would gradually diverge until it represented an unknown position within the space. Even before this stage it is unlikely that a small number of particles would continue to be sufficient to represent the PDFs.

In our implementation the cardinal headings are a property of a room polygon. In the same way as described for the EKF (section II-B) a confidence is ascribed to the cardinal headings to reflect the degree to which the pedestrian is confined in that space. This confidence is the variance of a gaussian distribution, the PDF of which is used to obtain particle weights.

In this way if the CHAIN confidence is high (e.g. in a corridor), the variance is low and only particles with headings close to the cardinal headings will gain significant weight. Conversely if the CHAIN confidence is low, the variance will be set to a high value and all particles will get a similar gain



(a) Equal Weighting



(b) CHAIN

Fig. 2: Particles propagating through a door (left to right). The initial particle set is shown in blue while the final set is shown in red.

in weight, regardless of their heading.

In figure 2b the effect of using the CHAIN weighting model is depicted. The same initialisation conditions are used as in 2a. However in this case CHAIN is used with a confidence (variance) of 5 degrees. It can be seen in figure 2b that the size of the position uncertainty, as reflected by the particles distribution, is smaller when the particles reach the door. Particles whose headings agree with the cardinal headings have been selected, while others have been removed, resulting in the short lines of particles either side of the main column.

The size of the final particle cluster shown in figure 2b (red) is comparable to that in figure 2a. However the particles near the center of the cluster have a higher weight reflecting their heading being closer to the easterly cardinal heading. This weighting means that they are more likely to be resampled and survive in the long term.

Figure 2b was chosen to illustrate the dangers of using CHAIN when the particle filter is not converged or when it is initialised with a large heading uncertainty. In this case a small, but not insignificant, number of particles have been initialised with headings closer to the northern cardinal heading than the eastern. Since they appear to agree with the cardinal headings they have been highly weighted and have survived. In this case we must rely on the wall geometry killing off the stray particles.

V. Implementation & Data Collection

In this section the implementation of the particle filter and the hardware used for data collection are described. All data used in this work were collected and logged for post processing. A real-time implementation of the algorithms described in this work is currently underway. The real-time implementation is intended to be run on a low cost single board computer (*Raspberry Pi*) with the user interface supplied by a smart-phone. The development of the algorithms and testing has been done with this end implementation in mind.

A. Software

The particle filter used in this work was prototyped using MATLAB® software while the 'POINT' EKF and ZUPT detector are implimented in C++. POINT has been previously described in detail elsewhere, for example [21]. The processing chain is depicted as a flow chart in figure 3.

For the purposes of this work 'POINT' was used to post-process the raw IMU measurements and form an initial navigation solution. Since absolute position and orientation is not of interest in this work the EKF was initialised with an approximate initial position and arbitrary initial heading as described in section II. The entire dataset was processed using ZUPTs detected using a threshold on the total IMU acceleration. No other aiding measurements were used in the EKF at this stage.

The output of POINT was then segmented into 'step' observations for use in the particle filter. Segmentation was performed using the previously detected ZUPTs with each step having the properties of 'length' (l) and 'change in direction from the previous step' ($\Delta\theta$). Length was measured in 2D with the change in height estimated by the EKF being discarded for the purposes of this work.

Each step measurement was assigned a variance for use in the model of the measurement uncertainties in the particle filter. In this work each step was assigned a the same set of variances ($\sigma^2_{\Delta\theta} = 1°$, $\sigma^2_l = 0.04$m).

Finally, map data (polygons and door locations) were stored in a MySQL database to allow efficient querying and transfer between devices.

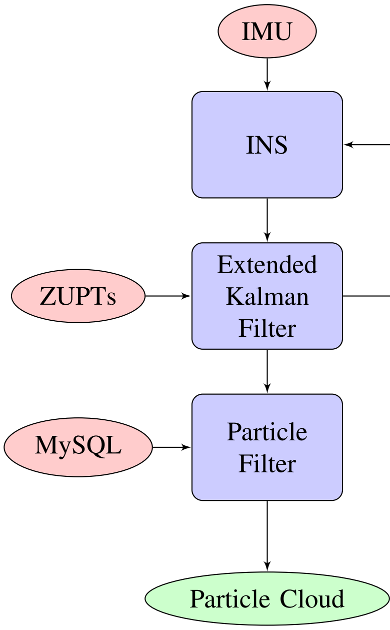


Fig. 3: Flow chart from raw IMU data to particle cloud output.

B. Hardware

Data for this work were collected using a Microstrain 3DMGX3-25 IMU ($\pm$16g model) operating at 100Hz. The IMU data was logged using a Samsung Galaxy Tab via an RS232 to Bluetooth connection. Since no data from other sensors was required, and the datasets were relatively short, counter values from the IMU internal clock were used to time stamp the data.

C. Data Collection

Results from two datasets are presented in this work. Each dataset is constrained to one floor of the Nottingham Geospatial Building. The first describes a trajectory on the ground floor of the building. The ground floor contains a large number of small offices and classrooms linked by corridors and foyer areas. There are many closely spaced doors on this level. This trajectory starts at the door of a classroom before traversing a second classroom, continuing around a corridor and leaving the building through the main door. The trajectory lasts around 2 minutes.

The second dataset was collected in a large ‘open plan’ area of the same building. The trajectory covers the entire space, constrained only by unmapped furniture. At no point does the trajectory leave the open plan space or pass through a door.

VI. Results

A. Ground Floor Trajectory

In this section we present the results of processing the first trajectory described in section V-C.

1) EKF Only: Figure 4 shows the results of processing the first trajectory with the EKF only. Mapping is shown for reference only, it was not used at this stage. The filter is initialised with accurate initial position and heading. No heuristic heading information is used at this stage.

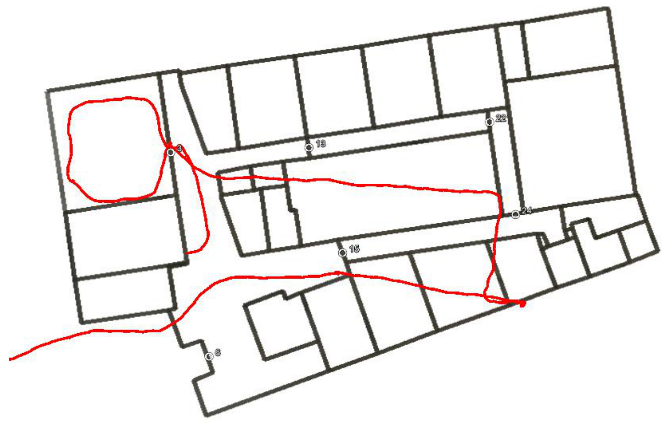


Fig. 4: Ground floor trajectory obtained using correctly initialised EKF.

2) Particle Filter: In figure 5 the results of using the particle filter are shown. The particle cloud from each step was written to the map to illustrate the change in the particle distribution over the course of the trajectory. The initial particle set is depicted as blue triangles.

A weighted mean of the particles was taken and depicted in red to give an example of a ’best guess’ selected from the particle cloud. This method for selecting the ‘best guess’ fails as the particle cloud leaves the classroom and briefly becomes bi-modal with one particle group in the corridor and another in a small cupboard.

It can be seen from these results that the particle filter effectively incorporates the map data to produce a representation of the state PDFs which are consistent with the mapping. The layout of this floor, with multiple closely spaced doors, often allows the particles to move into rooms which the user did not enter. Usually these particles are killed rapidly by wall constraints or more slowly by the cardinal heading weighting. Occasionally however the building geometry allows these ‘rogue’ particle groups to survive. An example of this can be seen in figure 5 when a group of particles enters the large rooms in the center of this floor and on the eastern side. The group of three closely spaced doors linking these rooms and the corridor allow these particles to survive until they are sufficiently de-weighted by the CHAIN process.

B. Open Plan Trajectory

We now present the results of processing the second trajectory described in section V-C.

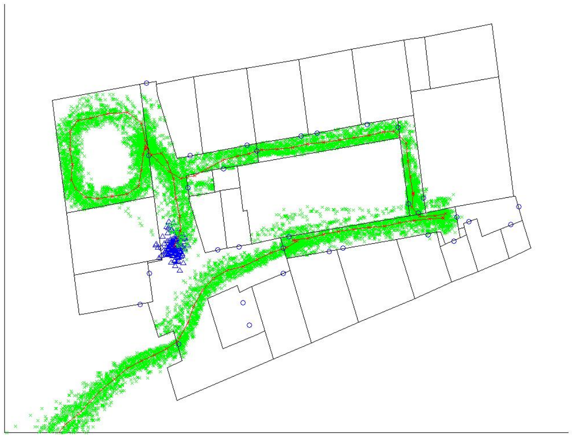


Fig. 5: The ground floor trajectory processed using map information. Green dots represent particles while the red crossed line is the weighted mean of the particle cloud at each step.

Figure 6a shows the results of processing the trajectory using the EKF only, with no heading aiding. Again it can be seen that heading drift rapidly makes the trajectory unusable for practical navigation even though the furniture layout of the room can be deduced by eye.

In figure 6b the results of processing the same data using the particle Filter are shown. For clarity only the 'best guess' (weighted mean) of the particle cloud is shown as a red crossed line. The final particle cloud is shown as a cluster of blue dots at the end of the trajectory.

In this case, while the trajectory is consistent with the mapping, heading drift is still causing some problems with the trajectory rotated with respect to the true furniture layout. The final particle distribution is shown in blue at the lower left hand corner of the room. Even after a short period of navigation in the open space (3mins) the cluster can be seen to be disintegrating into multiple small groupings.

Finally we present the results of processing the same trajectory using the full Particle Filter with CHAIN weighting (figure 6c). Again the 'best guess' or weighted mean of the particle cloud is shown. In this figure the trajectory is consistent with the mapping and the heading at each step is consistent with the furniture layout constraining the movement of the user. The final particle cloud (shown in blue) is more cohesive than that observed in figure 6b.

C. Success Rates

One effect of using CHAIN to weight the particles is a more precise, usually uni-modal, PDF of the position and heading states. As a result fewer particles are required to maintain a high success rate. Since the filter is initialised with particles drawn from a distribution representing the knowledge of the states at startup, the filter will not always perform in the same way for each navigation session.

The filter is considered to have failed if all of the particles have zero weight, for example they all impact a wall. This may occur due to mis-modeled measurement errors or insufficient particles being used to fully represent the state PDFs. In figure 7 we present the success rates obtained when processing the ground floor trajectory 500 times. To illustrate the effect of CHAIN we vary the initial position variance used in the filter. Similar effects are seen when the initial heading variance is changed.

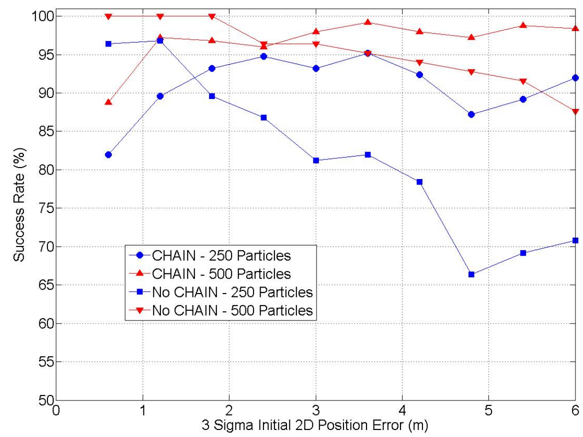
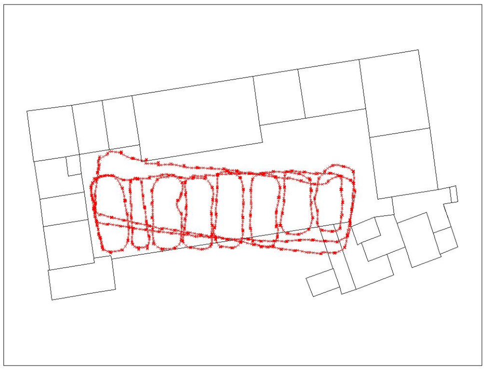


Fig. 7: Success rates for the Particle Filter run 500 times. Initial position standard deviation is varied from 0.2m to 2m.

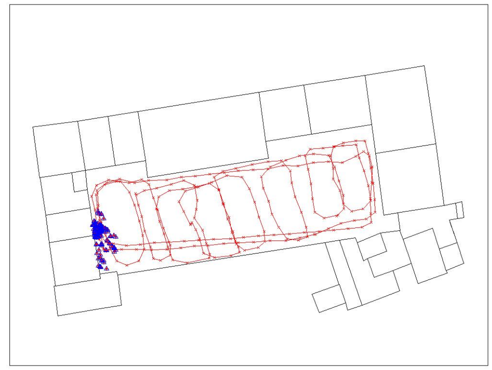
We begin by examining the filter performance without CHAIN weighting. In figure 7 these results are depicted in the blue line with squares (250 particles) and in the red line with downward pointing triangles (500 particles). It can be seen that for larger initial position variances the success rates drop when using either 250 or 500 particles. In the case of 500 particles the success rate drops to as low as 87% while for 250 particles it drops significantly to as low as 66%. For a precise initial position there is little difference between the 250 particle case and the 500 particle case.

Secondly we examine the use of CHAIN as a weighting model. In figure 7 these results are depicted in the blue line with circles (250 particles) and in the red line with upward pointing triangles (500 particles).In this case the reduction in the success rate is mitigated and little drop off is seen, even for very low precision initial position estimates. A significant improvement in the success rate is seen when 250 particles are used with the minimum success rate rising from 66% to 86% For most initial variances the performance of the 250 particle filter approaches that of the 500 particle filter.

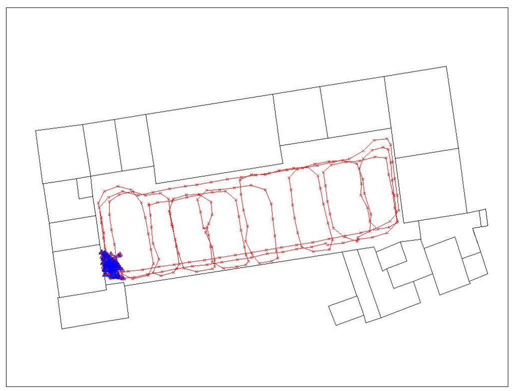
An anomaly can be seen in figure 7 when a highly precise initial position is used, CHAIN actually reduces the success rate compared to the equivalent filter without CHAIN. This is because the models no longer account for the errors in the initialisation parameters. When CHAIN is used, the heading



(a) EKF, No CHAIN



(b) Particle Filter, No CHAIN



(c) Trajectory two obtained using the Particle Filter with CHAIN applied.

Fig. 6: The open plan trajectory.

values remain so precise that they can no longer mitigate the effect of small errors in the starting conditions. This effect may be removed through tuning the filter, increasing the step measurement noise to compensate. However we have found that correctly modeling the precision of the initial parameters is preferable.

D. Processing Times

CHAIN is a computationally inexpensive way to weight particles. In this work a 'per particle' processing time was calculated over the course of 500 successful runs. This was done using both 500 particle and 250 particles and reflects the mean time taken to propagate, update and resample one particle at each step (the resampling time being split across all particles).

Data were processed on a laptop with an Intel®Core i3™ CPU running at 2.4GHz. Times were measured using the MATLAB® timer function.

We found that, as expected, the per-particle processing time is very similar regardless of the number of particles used. In our implementation we observed a mean, per particle, per step, processing time of 0.178ms without using CHAIN. When CHAIN was used we observed an increase in the processing time of 0.029 ms per particle, per step.

VII. Conclusion

In this work we have shown that the CHAIN method for heuristic heading aiding can be successfully applied to a pedestrian navigation system based upon a particle filter. We have described the design and implementation of such a filter and have performed tests in an environment where full and complete mapping is available.

CHAIN has been demonstrated to improve positioning performance in areas where few map constraints are present, or are unavailable. It has been shown to be capable of significantly reducing the filter failure rate from 44% to 14% when a small number of particles is used in the filter (250) and the initial position is poorly known. The CHAIN method requires little extra processing and requires only one heading to be stored per navigation area (polygon). In our implementation CHAIN has been observed to increase the processing time by around 15% when compared to the same particle filter with an equal weighting strategy.

CHAIN is equally applicable regardless of the fidelity & completeness of the mapping available and can be simply adjusted in the filter to reflect the confidence in its underlying assumptions for a particular space.

Future work will look at implementing the particle filter and CHAIN algorithm on a small single board computer for real-time navigation. We will also investigate implementing feedback from the particle filter to the EKF to aid observation of heading drift and therefore improve EKF only navigation performance when map data is unavailable.

ACKNOWLEDGMENT

This work was funded by the RCUK Horizon Digital Economy Research Hub grant, EP/G065802/1.

REFERENCES

[1] E. Foxlin. Pedestrian tracking with shoe-mounted inertial sensors. *IEEE Computer Graphics and Applications*, pages 38–46, 2005.

[2] I. Skog, J.O. Nilsson, and P. Handel. Evaluation of zero-velocity detectors for foot-mounted inertial navigation systems. In *Indoor Positioning and Indoor Navigation (IPIN), 2010 International Conference on*, pages 1–6. IEEE, 2010.

[3] J. Pinchin, C. Hide, and T. Moore. The use of high sensitivity gps for initialisation of a foot mounted inertial navigation system. In *Position Location and Navigation Symposium (PLANS), 2012 IEEE/ION*, pages 998–1007. IEEE, 2012.

[4] K. Abdulrahim, C. Hide, T. Moore, and C. Hill. Aiding low cost inertial navigation with building heading for pedestrian navigation. *Journal of Navigation*, 64(02):219–233, 2011.

[5] B.M. Scherzinger. Inertial navigator error models for large heading uncertainty. In *Position Location and Navigation Symposium, 1996., IEEE 1996*, pages 477–484. IEEE, 1996.

[6] E.H. Shin. Estimation techniques for low-cost inertial navigation. Master's thesis, 2005.

[7] O. Woodman and R. Harle. Rf-based initialisation for inertial pedestrian tracking. *Pervasive Computing*, pages 238–255, 2009.

[8] J. Bird and D. Arden. Indoor navigation with foot-mounted strapdown inertial navigation and magnetic sensors [emerging opportunities for localization and tracking]. *Wireless Communications, IEEE*, 18(2):28 –35, april 2011.

[9] K. Abdulrahim, C. Hide, T. Moore, and C. Hill. Aiding MEMS IMU with building heading for indoor pedestrian navigation. *Ubiquitous Positioning Indoor Navigation and Location Based Service (UPINLBS)*, 2010.

[10] J. Pinchin, C. Hide, K. Abdulrahim, T. Moore, and C. Hill. Integration of heading-aided mems imu with gps for pedestrian navigation. In *Proceedings of the 24th International Technical Meeting of The Satellite Division of the Institute of Navigation (ION GNSS 2011)*, pages 1346–1356, 2001.

[11] K.R. Britting. *Inertial navigation systems analysis*. John Wiley and Sons, New York, 1971.

[12] J. Farrell and M. Barth. *The global positioning system and inertial navigation*. McGraw-Hill Professional, 1999.

[13] B.M. Scherzinger and D.B. Reid. Modified strapdown inertial navigator error models. In *Position Location and Navigation Symposium, 1994., IEEE*, pages 426–430. IEEE, 1994.

[14] D. Fox, S. Thrun, W. Burgard, and F. Dellaert. Particle filters for mobile robot localization. *Sequential Monte Carlo Methods in Practice*, pages 401–428, 2001.

[15] I.M. Rekleitis. A particle filter tutorial for mobile robot localization. Technical Report TR-CIM-04-02, Centre for Intelligent Machines, McGill University, 2004.

[16] S. Beauregard, M. Klepal, et al. Indoor pdr performance enhancement using minimal map information and particle filters. In *Position, Location and Navigation Symposium, 2008 IEEE/ION*, pages 141–147. IEEE, 2008.

[17] O. Woodman. *Pedestrian Localisation for Indoor Environments*. PhD thesis, University of Cambridge, Computer Laboratory, 2010.

[18] S. Kaiser, M. Khider, and P. Robertson. A maps-based angular pdf for navigation systems in indoor and outdoor environments. In *Indoor Positioning and Indoor Navigation (IPIN), 2011 International Conference on*, pages 1–7. IEEE, 2011.

[19] N.J. Gordon, D.J. Salmond, and A.F.M. Smith. Novel approach to nonlinear/non-gaussian bayesian state estimation. In *Radar and Signal Processing, IEE Proceedings F*, volume 140, pages 107–113. IET, 1993.

[20] R. Douc and O. Cappé. Comparison of resampling schemes for particle filtering. In *Image and Signal Processing and Analysis, 2005. ISPA 2005. Proceedings of the 4th International Symposium on*, pages 64–69. Ieee, 2005.

[21] C. Hide, J. Pinchin, and D. Park. Development of a low cost multiple gps antenna attitude system. In *Proceedings of the 20th International Technical Meeting of the Satellite Division of The Institute of Navigation (ION GNSS 2007)*, pages 88–95, 2001.