Dual IMU Indoor Navigation with Particle Filter based Map-Matching on a Smartphone

C. Ascher*, C. Kessler*, M. Wankerl* and G.F. Trommer*

*Karlsruhe Institute of Technology, Institute of Systems Optimization, Germany. Email: christian.ascher@kit.edu

Abstract— In this paper an Indoor Navigation System with map-matching capabilities in real-time on a smart phone is presented. The basis of the system is an in-house development of an Integrated Pedestrian Navigation System, based on 2 low-cost IMUs, an electronic compass and an altimeter with a drifting navigation solution. Combining this system with an additional laser ranger and SLAM algorithms, we are able to build accurate maps of office buildings for already visited rooms in post processing.

This paper presents a map matching algorithm based on a new reduced particle filter in order to use these maps later for real-time applications without an expensive laser ranger but relying only on the dual inertial system. It can be used with both, pre-processed SLAM maps or with already available maps. Finally to smooth the resulting trajectory after particle filtering we propose the use of a new "balanced bubble band smoother" allowing the trajectory to optimally match to both, map and recorded IMU data. This new approach makes it possible to do map matching online on a smart phone.

***Index Terms*—Pedestrian Navigation, Indoor Navigation, Particle Filter, OpenMoko**

I. Motivation

Localization and navigation in indoor environments is a core issue for fire fighters, police task forces and first responders but also for the blind. In general it is necessary to provide navigation without any knowledge of the building when the emergency responders get in; knowledge of the building is often not available. But if a building plan is available, it is helpful to make use of the building information. Also a map that is created with laser or vision sensors and provided via internet could be used by another user in the same scenario. In any of the applications, aiding with maps would be desirable where available. In combination with an inertial based pedestrian navigation system, this will increase localization performance, navigation robustness and long-term stability. To make this system operable for mobile operators, it should be running on a small hand held device.

II. Dual IMU System

The sensor basis for the approach is an in-house development of a Dual IMU System [1], which takes advantage of Zero Velocity Updates of a foot mounted IMU and records the dynamics of a second, torso mounted unit (IMU, MAG, BARO). The torso setup can be extended with a laser or a camera sensor, see figure 1.

Figure 1: *Dual IMU system: with GPS, laser, camera, IMU, MAG, BARO and a foot IMU*

Due to the Dual IMU concept a tightly-coupled data fusion between torso and foot units is possible resulting in an only slightly drifting solution where mainly attitude errors remain in the system due to magnetic field anomalies in indoor scenarios.

But combined with a laser sensor, this allows constructing building maps determined by OrthoSLAM and Graphical SLAM algorithms [1], [2]. Figure 2 shows the results of the SLAM algorithms: The drifting IMU solution combined with the SLAM results yields a robust navigation solution and an estimated map. Good results and loop closure can be reproduced even in long corridors with little longitudinal laser aiding.

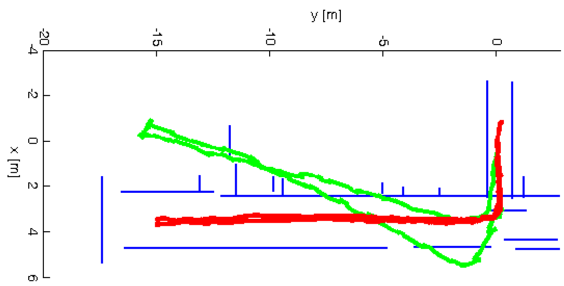

Figure 2: *OrthoSLAM: the drifting solution of the Dual IMU system (green) and the results of the tight coupled OrthoSLAM algorithm and generated map*

During a realistic application the laser or vision aiding sensors may be temporarily not available but the local map is already estimated. Then we require the calculated position is within the previously estimated map without using vision or laser sensors but based on inertial sensors and the map only. Another task for a map matching scenario: if maps are already available and the user is only wearing an inertial system, without vision or laser sensors.

III. Map Matching on a Smartphone

Processing the building information to solve the multi-modal problem of map-matching requires high computational effort. Often particle filters are used but those cause a high computational burden due to a drifting navigation solution. However, a portable system on a smart phone has not enough computational power to solve the problem with standard map matching algorithms based on particle filters.

Figure 3: Linux based *OpenMoko* smart phone

Therefore we propose a simplified map-matching particle filter which is running on a Linux based OpenMoko phone (see figure 3) using the compass aided navigation solution of the dual IMU system which is slightly drifting with time. This solution is calculated without the Laser SLAM capability but only inertial sensors are used. The connection between smart phone and sensors is realized as a USB-serial connection to the torso unit and a Bluetooth connection to the foot mounted IMU. The navigation result of the torso unit is sent to the map matching filter. The application is programmed in C++ using Qt libraries and is used to control the Dual IMU system and to display online results. With our approach, it is possible to match the slightly drifting Dual IMU based solution to a known map online on a smart phone with enough remaining calculation power to run a operating system with displaying functionality.

IV. Particle Filter

In this chapter the functionality of our new, reduced particle filter will be demonstrated based on a standard Bootstrap particle filter implementation. [3]. Particle filters are used to estimate the state of a system as a statistic state where the corresponding probability density function (PDF) is approximated numerically. This can be used, if the density function is not Gaussian for example for multi mode applications like map matching.

The system model may be given by:

$$x_k = f(x_{k-1}) + w_k \tag{1}$$

with a corresponding measurement model

$$y_k = h(x_{k-1}) + v_k \,. \tag{2}$$

System noise w_k and measurement noise v_k are assumed to be uncorrelated and white noise with their probability density function p_{wk} and p_{vk} which does not need to be Gaussian. The distribution of a given probability distribution $p(x_{k-1}|Y_{k-1})$ is approximated with a number of N particles x^i_{k-1} and the approximation error disappears for $N \to \infty$. The particles are randomly generated (operator: $\propto$) to approximate the given probability distribution function:

$$x^i_{k-1} \propto p(x_{k-1}|Y_{k-1}), \tag{3}$$

assuming a Markov process, with a density depending on the actual system state with regard to all available observations. [4] Then the probability density can be written as a weighted sum with the weight ω^i for each particle:

$$p(x_{k-1}|Y_{k-1}) = \sum_{i=1}^{N} \omega^i \cdot \delta(x_{k-1} - x^i_{k-1}) \tag{4}$$

where $\delta(\cdot)$ represents the Dirac function.

PROPAGATION:

To propagate the probability density in time with given system noise but moved with the mean value $f(x_{k-1})$, it can be written:

$$p(x_k|x_{k-1}) = p_{wk}(x_k - f(x_{k-1})) \tag{5}$$

and taken into account the Chapman Kolmogorov equation for Markov processes, the propagated density function can be written as:

$$p(x_k|Y_{k-1}) = \sum_{i=1}^{N} \omega^i \cdot p_{wk}(x_k - f(x_{k-1})) \tag{6}$$

This means, particles are updated with randomly generated numbers following the distribution p_{wk}, the weights ω^i are not updated during propagation.

ESTIMATION:

For given system observations or measurements Y_{k-1}, an update of the weights ω^i is calculated following measurement and measurement noise:

$$\omega^{i,+} = \omega^i \cdot p_{vk}(y_k - h(x^i_k)) \cdot c \tag{7}$$

where c normalizes the sum of all weights to 1.

RESAMPLING:

To avoid degeneration of the particle approximation, the weights must be resampled yielding the same weight for

978-1-4244-5864-6/10/$26.00 ©2010 IEEE

each particle. Due to a high calculation burden, resampling often is not done at each time step.

In the literature, often particle filters are used for map matching [5], [6], and [7]. An example for map matching in pedestrian navigation is given in a paper from [8], showing the capability of particle filters for indoor scenarios. The proposed backtracking method can go back in time, if a dead end room is found. But due to the implementation of the particle filter, the solution is not capable to be used online on our smart phone.

For our 2D map matching applications for indoor scenarios a particle filter comes into operation to estimate the position x and y with a reduced calculation burden. This is realized by reducing the particle weight to a binary weight:

$$\omega^i \in 0,1 \qquad (8)$$

so that only the *number of particles per area* describe the shape of the 2-dimensional density function. With this reduction, the high calculation burden for estimation and resampling steps can be reduced significantly. The following changes have to be realized:

- The estimation step is simplified by setting the weight to zero, if a particle is walking “through” a wall and it can be deleted.
- As a consequence, in the resampling step, a number of M new particles have to be generated to maintain a constant number of N particles. Therefore new samples are reproduced of the actual, reduced distribution $p(x_k|Y_{k-1})$ with the new weights $\omega^{i,+}$. The simplest way to do so is to randomly select out of the number of (N-M) particles and reproduce those.
- For the propagation step nothing changes: all particles are propagated with a generated random vector, covering the probability density function of the system noise.

The density function for the propagation step is demonstrated in figure 4, realized with 2 system noise parameters: step length noise and angle estimation noise. This yields a non Gaussian distribution but can easily be used in prediction steps in a particle filter.

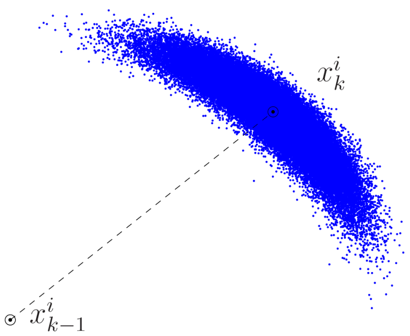

Figure 4: Density function for PF-prediction (one foot step)

V. PARTICLE FILTER RESULTS

In this chapter, simulation results and results with real data will be presented with our reduced particle filter algorithm: First a Matlab simulation in an office building with 9 rooms is evaluated.

- The simulated Dual IMU result is degraded by a drift in time resulting in a path through walls.
- At the beginning, a number of 1.5 particles per m² are distributed over the office building area.
- At each foot step a prediction of the old particle position is done with the non Gaussian distribution as described above.
- If a particle crosses a wall, it is deleted

After walking about half the way through the building from left to right (see figure 5), three clusters of particles are remaining - representing 3 possible areas where the user could be after having walked about 15 meters from right to left. After continuing on a unique path, the position algorithm will converge and the actual position of the user can be estimated without the knowledge of the starting point, see bottom of figure 5. Even the starting point can be calculated by reverse processing to reach convergence also at the starting phase, see figure 5 at the bottom right side.

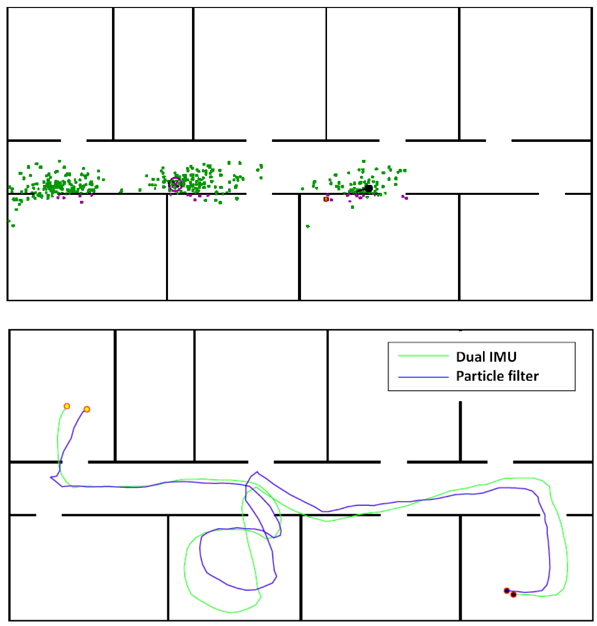

Figure 5: *Upper:* Position probability distribution after 15 m walk. *Lower:* Inertial drifting solution (green) and the PF result (blue). The starting point (right) also was found automatically by data reverse processing.

The same results have been evaluated with real Dual IMU data. Figure 6 shows the results of map matching of the real data Dual IMU solution to the given realistic map of the office building. At the beginning, the starting point is unknown. It can be seen, that real inertial data processing provides a long-term stable drift-less estimation of the walked path although, if the inertial system drifts over time, see figure 6. The particle filter solution is realized in C++ to be calculated online and shown on the smart phone display, see figure 3.

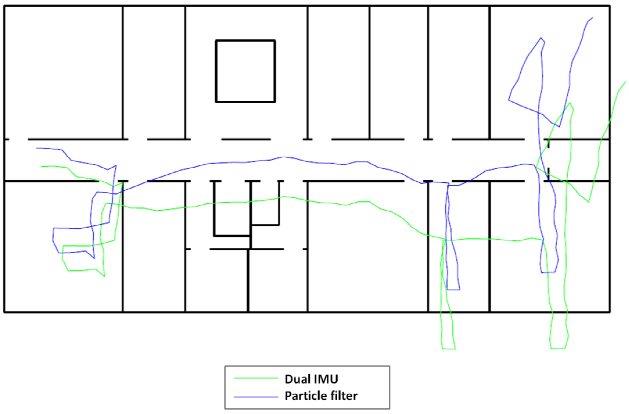

Figure 6: *Real data processing*, starting left: the drifting Dual IMU solution is matched to the map, long-term stable, the starting is estimated.

VI. MAP WITH UNKNOWN NORTH DIRECTION

By adding a third degree of freedom for the yaw angle estimation, it is even possible to estimate an unknown north direction of a map. This can be done by increasing the angle estimation noise to 360°, resulting in a circle distribution around the actual position, cf. figure 7.

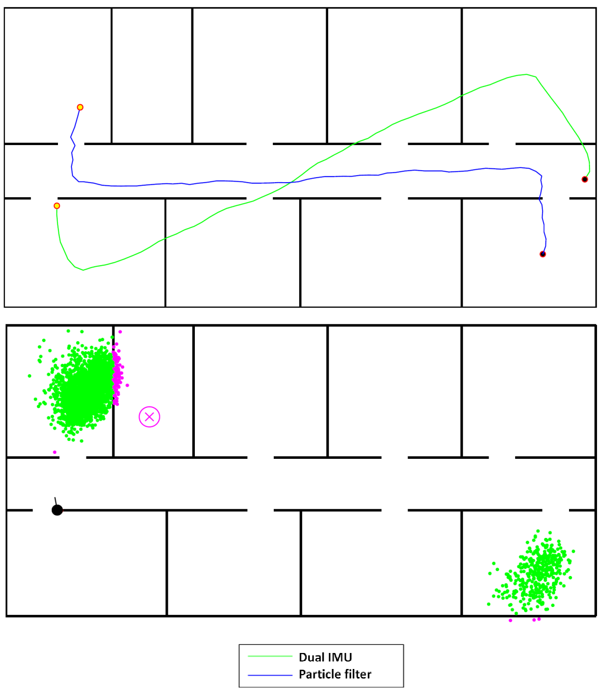

Figure 7: Estimation of Map north direction. *Upper:* A map north error of 30° degree can easily be estimate. *Lower*: if the path is not unique, then multiple solutions remain

Assuming a given map with a north error of about 30°, the resulting estimated Dual IMU trajectory would be rotated at -30° inside the map, see figure 7. If the 3-degree-of-freedom particle filter is used, it is even possible to match this path to the given map, if the path is unique. In this situation, the path still is not unique, and the particle filter has two convergence points. This can be solved with constraints or by a unique path, the filter works as expected. But as a consequence more particles are needed to cover the third dimension.

VII. BALANCED BUBBLE BAND SMOOTHER

Figure 8 shows results in detail of a particle filter solution with artifacts of the nonlinear particle filtering due to the nonlinear wall estimation steps.

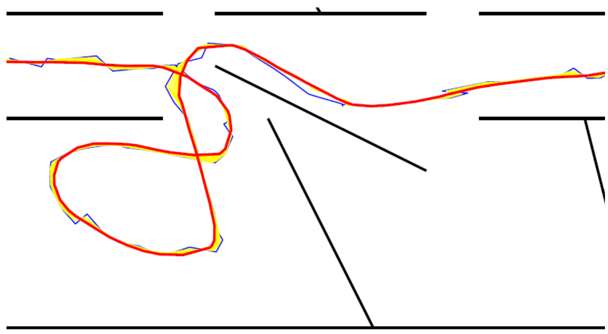

Figure 8: Particle filter results with many artifacts in detail after processing (blue) and "balanced bubble band smoothing": iterations (yellow) & balanced solution (red)

For further smoothing of this result, we propose the use of a new "balanced bubble band algorithm" which uses the smoothing force of a bubble band [9] but also holds the "balance" with the IMU solution to optimal match both, map and IMU data without contracting loops but still smoothed. Foot step position estimates of the particle filter solution are treated as bubbles that are connected with springs. The standard bubble band smoother consists of 2 forces; we extend it by a third force:

1. Force of contraction
2. Force of rejection
3. Force of inertial delta angle

For each bubble, the forces above can be calculated and the sum of these forces multiplied by a scale factor α yields the new position estimate:

$$\vec{b}_{new} = \vec{b}_{old} + \alpha \cdot \vec{f}_{1,2,3} \tag{9}$$

Force of contraction:
The first one is the force of attraction, equation showing the force for a given point, that it gains of its two neighbors:

$$\vec{f}_c = k_c \cdot \left(\frac{b_{i-1} - b_i}{\|b_{i-1} - b\|} + \frac{b_{i+1} - b_i}{\|b_{i+1} - b_i\|} \right) \tag{10}$$

The force of rejection approximates the behavior of the bubbles, which aim not to be too near to a wall. It should be used only near walls with the maximum distance to a wall ρ_0. The force of rejection is calculated in a way, that the actual bubble gains an increasing radius:

$$\vec{f}_r = \begin{cases} k_r(\rho_0-\rho)\dfrac{\partial\rho}{\partial b} & \rho<\rho_0 \\ 0 & \rho\geq\rho_0 \end{cases} \tag{11}$$

where k_r is the tuning factor for the rejection force and must be negative. For the radius variation, the following calculation can be used:

$$\frac{\partial\rho}{\partial\vec{b}} = \frac{1}{2h}\begin{bmatrix} \rho(b-hx)-\rho(b+hx) \\ \rho(b-hy)-\rho(b+hy) \end{bmatrix} \tag{12}$$

In [9], the scale factor h is proposed to be $\rho(b)$, but we propose the use of $h=\rho(b)/2$ resulting in a more realistic position shift.

As described above, these two forces would contract loops and curves to straight lines. This is why a third force comes into operation which – if in balance with the other two forces - spreads the solution, forcing the angles to be those, that were measured by the Dual IMU system. The force can be written as:

$$\vec{f}_t = k_t \begin{bmatrix} \cos(\psi) \\ \sin(\psi) \end{bmatrix} \tag{13}$$

This balanced bubble band algorithm can only be used for paths that do not cross walls. Therefore intersections first must be resolved, for example with an adapted intersection slide algorithm.

VIII. Outlook: Multi floor map matching

In the future, the proposed particle filter also will be used for multi floor map matching; stairs and ladders are also represented only in 2-dimensions because a person never flies through a room or a staircase. Then the height estimates can also be used to eliminate non-matching particles. First results are very promising, see figure 9.

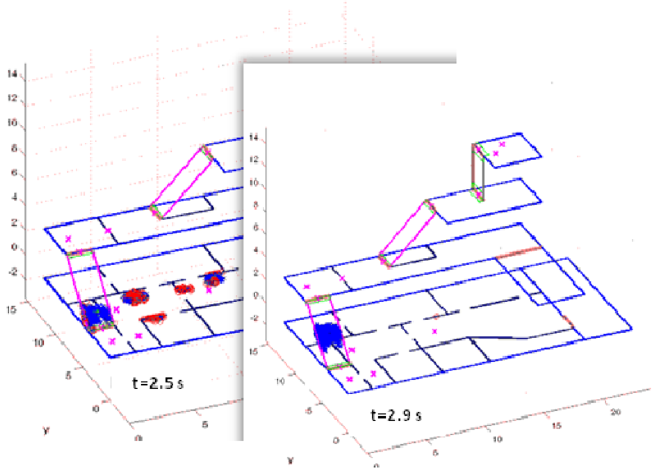

Figure 9: Multi floor map matching simulation with the presented reduced particle filter.

IX. Conclusion

It has been shown that a standard particle filter based map-matching algorithm can be adapted and simplified in order to be applicable on a smart phone. Online calculated results prove the efficiency of our approach. In the future we want to extend this approach for multi-floor map matching, first simulation results are very promising.

References

[1] C.Ascher, C.Kessler, M.Wankerl, G.F.Trommer. "Using OrthoSLAM and aiding techniques for precise pedestrian indoor navigation" ION GNSS 2009, September 22-25,2009 Savannah, Georgia, USA.

[2] Nguyen, V., Harati, A., and Siegwart, R. (2007). A lightweight SLAM algorithm using orthogonal planes for indoor mobile robotics. In Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems IROS 2007, pages 658–663

[3] Jan Wendel. *Integrierte Navigationssysteme*. Oldenbourg, 2007

[4] Uwe Kiencke. *Ereignisdiskrete Systeme - Modellierung und Steuerung verteilter Systeme*. Oldenbourg Wissenschaftsverlag GmbH, 2006.

[5] Birgit Graf, José Manuel Hostalet Wandosell, Christoph Schaeffer. Flexible path planning for nonholonomic mobile robots. In *Proceedings of The fourth European workshop on advanced mobile robots (EUROBOT'01)*, pages 199–206, 2001.

[6] Alain L. Kornhauser Christopher E. White, David Bernstein. Some map matching algorithms for personal navigation assistants. *Transportation Reseach*, Part C 8:pages 91 – 108, 2000.

[7] Kyung-soo Chon Jae-seok Yang, Seung-pil Kang. The map matching algorithm of gps data with relatively long polling time intervals. *Jounal of the Eastern Asia Society for Transportation Studies*, Vol. 6:pages 2561 – 2573, 2005.

[8] Widyawan, Martin Klepal, Stéphane Beauregard. A backtracking particle filter for fusing building plans with pdr displacement estimates. In *Proceedings of the 5th Workshopon Positioning, Navigation and Communication 2008 (WPNC'08)*, 2008.

[9] Khatib, Maher; Jaouni, H.; Chatila, R.; Laumond,J.P.:"Dynamic path modification for car-like nonholonomic mobile robots." In: IEEE International Conference on Robotics and Automation, Albuquerque, 1997, pp. 2920-2925.