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

Multi-Floor Map Matching in Indoor Environments for Mobile Platforms

C. Ascher, C.Kessler, R.Weis, G.F.Trommer
Karlsruhe Institute of Technology, Germany
Institute of Systems Optimization
Karlsruhe, Germany
christian.ascher@kit.edu

***Abstract*— In this paper a map matching algorithm for multi floor indoor environments including guidance of a user is presented. Inertial sensor based pedestrian navigation systems are subject to drift. Maps are to eliminate that drift by matching the path to a map. In multi floor scenarios, we propose to use not only flat floor plans but also transitions like staircases as well as ladders and elevators. This is essential for an enhanced matching of a user path to a given 3D floor plan. Especially in industrial applications this is a crucial factor as also ladders and elevators exist. In contrast to other authors, this paper considers all of these objects. 3D position estimations from an Integrated Pedestrian Navigation System (IPNS) including an inertial system as well as a barometer and a magnetometer are used. Maps must be available, either from a prior Simultaneous Localization and Mapping (SLAM) survey or from facility maps.**

The new map representation is used to match real sensor data from the IPNS system to a map in a real time implementation. Furthermore an online guidance implementation is presented which is also based on the new 3D map representation.

Keywords: Pedestrian Navigation, Indoor Navigation, Map Matching, Particle Filters, Multi-floor

I. INTRODUCTION

In many applications, maps are known before the actual deployment. They greatly ease the navigation task for indoor pedestrian navigation because long term stability can be obtained. Map-Matching (MM) is used to fit an estimated path into these maps. It is often realized based on mathematical tools like Sequential Monte Carlo (SMC) methods, also referred to as particle filter: A large number of SMC particles is distributed over the digital map where rooms are represented as impenetrable walls. The particles approximate the probability density function of the user's position, moving into the direction of the estimated path and if a SMC particle collides with a wall, it is excluded from the Monte Carlo simulation.

On behalf of computation time, the degree of freedom of this formerly 3D problem often is reduced to 2D, so the height of a trajectory is not considered. Depending on the map and walked path, SMC map matching can completely eliminate estimation drift. Even an unknown starting point can be estimated after some time.

In the literature, often particle filters are used for map matching [1], [2], and [3]. Another example for map matching in pedestrian navigation is given in a paper from [4], showing the great 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.

In [5], the authors propose a map matching algorithm based on a particle filter, which is able to incorporate these nonlinear map-matching techniques. In that paper, the importance of having the map of the environment to reduce position drift of an inertial bases navigation system is pointed out again.

Our work is based on the 2D MM algorithm presented in [4], but we have reduced computation time with binary weights of each particle, to run it on a smart phone, [6].

To extend the 2D implementation to multi floor 3D MM, other research groups introduce additional virtual tracks, virtual floors or other approaches: In [7], a MM algorithm is presented using a virtual track. A virtual track is where all possible trajectories and paths in a building are assembled. This makes it very efficient when it is used with the pedestrian navigation system, but of course this works only good in areas, where a virtual track is available, for example in a corridor or on the street. For complex industrial areas, this approach seems not suitable. [8] is also using a particle filter based MM and proposes an intermediate virtual floor between staircases for 3D maps. The map must be known well, as the direction of the stairs helps the MM algorithm to find a user position inside the staircase, even the speed is adjusted in the staircase, depending on the user climbing up or down. User motion inside the staircase is represented by an extended Markov model. Finally the new floor level is set depending on the model based information. This seems to be enough for standard staircases but the height information is not used although it could increase the robustness of the floor transition estimation. The authors increase the robustness with movement models, see [8] for more details. But elevators or ladders for industrial facilities are not addressed.

[9] also proposes the use of MM based on particle filters. Multiple floors are taken into account by adding additional platforms for every step of a stairway. The differential height

information of the foot mounted pedestrian navigation system is used to watch the transition from one step to another. For very detailed maps, this approach seems to be a good solution. However, obtaining maps with that level of detail in a real-world mission might be a problem, as every step needs to be known; if one step is missing in the map, the proposed MM might fail in a staircase. Furthermore, ladders and elevators have not been addressed in the paper.

In this present paper, we propose a new map representation where rooms are represented as rectangles with additional information like doors or transitions. Staircases are represented as sloping rectangles and elevators and ladders as vertical rectangles that can be traversed by the user. The accurate height estimation from IMU and barometer measurements is gainfully exploited by imposing an additional constraint for each particle, finally matching the estimated trajectory to the multi-floor map. Even slightly inaccurate height profiles due to barometric drift lead to correct estimation results, which shows the robustness of our approach.

The new map representation is also very profitable for personal guidance in the new map. Our guidance approach will also be presented.

Results from simulation and real-world test runs will demonstrate the capabilities of our real-time implementation of map matching and guidance.

II. SCENARIO AND ENVIRONMENTAL CONDITIONS

A. Scenarios

One of the intended scenarios is the use in chemistry plants. The environment consists of multiple rooms, floors and platforms without reliable GPS availability. Many vertical connections such as stairs and ladders yield a very complex map. Typically, external workers are on-site for maintenance only for short times, and might be unfamiliar with the environment. Personal navigation systems support the user in finding quick and safe ways, and therefore reduce costs and risk. 3D Maps are available from construction plans.

Another scenario is the deployment of first responders on major events in complex buildings like sport arenas or official buildings. When using multi floor map matching combined with a personal dead reckoning system, first responders will find their way fast and easily.

B. Environmental and hardware conditions

For environments as described above, GNSS availability often cannot be guaranteed and multi path can mitigate the accuracy especially in an industrial outdoor environment or in indoor environments.

The position information from any inertial sensor based pedestrian navigation system is always subject to drift if no assumptions can be made about the environment. Beside position drift, another challenge is to obtain long-term stable heading information, so often an electronic compass is used but with the disadvantage of the vulnerability to iron effects at plants or indoors. The map matching algorithm eliminates this drift.

Another constraint is the mobile computing platform where all calculations have to be performed.

C. Specialities in pedestrian navigation

In pedestrian navigation, a user will always walk on the floor, so it's not necessary to model a true 3D environment with three full spatial degrees of freedom. Movement is only possible in rooms, on stairs, elevators and ladders. Even the height over ground can be neglected, knowing the exact height in a room doesn't provide more information for a user. But of course the transition to other floors must be detected when a multi floor scenario is present.

D. Maps

Currently, there is no common standard for indoor map data. In most practical cases, it is still necessary to compile a digital map for MM using different sources like 2D or 3D plans in a computer-aided process supervised by a human operator. In any case, when digitalizing a given map, meta information about transitions between rooms and room numbers must be provided to realize map matching and guidance. For unmapped buildings, laser based Simultaneous Localization and Mapping (SLAM) methods for pedestrian indoor navigation [4] can be used for 3D maps, but meta information must still be added in any case.

III. DUAL IMU SYSTEM

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

Fig. 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 unit 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.

IV. HARDWARE

Processing the building information to solve the multi-modal problem of map-matching requires computational effort. Often particle filters are used but those cause a high computational load. Especially when the user motion on a 3D map shall be tracked, special adaptations of the SMC estimation process are essential for achieving real-time performance on a portable system Therefore we propose a simplified map-matching particle filter using the compass aided navigation solution of the dual IMU system which is slightly drifting with time. The connection between computer and Dual IMU System 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 also used to control the Dual IMU system and to display online results.

V. PARTICLE FILTER

In this chapter the functionality of our reduced particle filter will be demonstrated based on a standard Bootstrap particle filter implementation [10]. 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_{k-1}^i and the approximation error disappears for $N \to \infty$. The particles are randomly generated (operator: $\propto$) to approximate the given probability distribution function:

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

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

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

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:

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

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 each particle.

ADAPTIONS

For our 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:

$$\omega^i \in 0,1 \tag{8}$$

so that only the number *of particles per area* (n/m^2) describe the shape of the 2-dimensional density function. With this reduction, the high calculation burden when calculating the the weights in equation (7) for estimation and resampling steps can be reduced significantly. The filter finally is realized as follows:

- 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 from the actual, reduced distribution $p(x_k|Y_{k-1})$ with the new weights $\omega^{i,+}$. simplest very fast way to do so is to randomly select from the number of (N-M) particles and reproduce those.

- the propagation step is unchanged: 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 shown in Fig. 2, realized with two 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. This particle implementation is realized for a 2D propagation of particles.

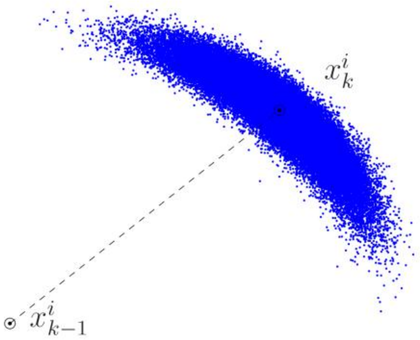

Fig. 2: Density function for PF-prediction (one foot step)

VI. 3D MAP REPRESENTATION AND PARTICLE PREDICTION

Based on our previous 2D particle filter work and in order to restrict the computational load when having a 3D map, we propose to exploit the fact that a user will always walk on the floor. Even if a user is going upstairs, he will stay on the stairs. So we propose to use a new representation of the map with three possible objects:

- Room: bounding walls, transitions to other objects
- Stairs: inclined planes with bounding lines and transitions to other classes
- Ladders/elevators: vertical rectangle with bounding lines
- Additional obstacles in a room (fine path planning)

An example of a building can be seen in Fig. 3 with rooms, stairs and a ladder. The transitions between the classes are marked in green representing the area where particles can change their class affiliation.

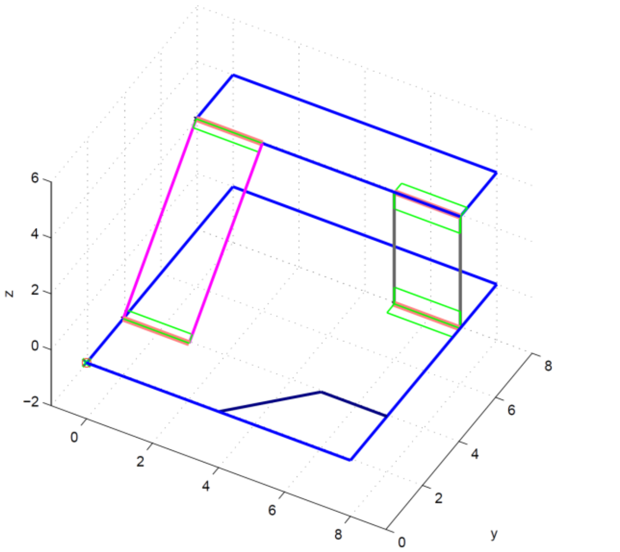

Fig. 3: *3D map with rooms, stairs ladders and additional obstacles in room*

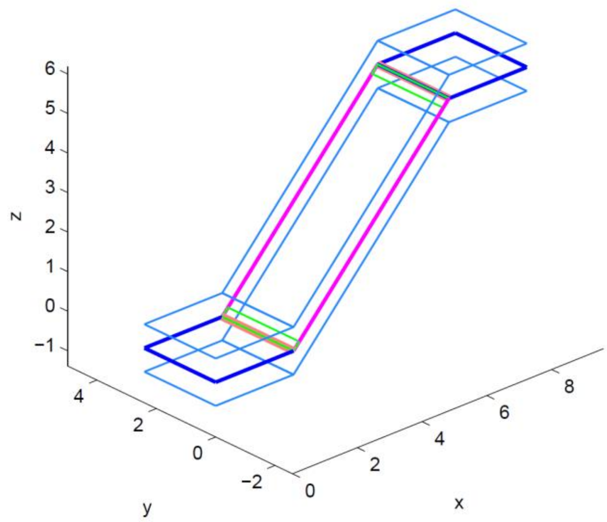

Fig. 4: *Tolerance regions around map objects*

To move particles in the map, the navigation solution of the pedestrian navigation system is used from each foot step (dx, dy, dz), so 2D position and the estimated height is given which is estimated by barometer and foot sensor. With this input, every particle in the map is moved in the given direction with the noise as described above. Now, the following constraints are given for each particle:

- A particle cannot move through a wall
- A particle cannot enter stairs or ladders from behind
- A particle cannot jump into the air, it can only stay near the floor. In the case of the stairs, the position must correspond to the given height.
- A particle can leave stairs and ladders only through the transitions

So for each particle and for every foot step, a collision test has to be performed. If a particle violates the constraint, then its probability will be set to zero, and it gets deleted. As described above, a particle will always stay near the floor, so a tolerance region has to be defined as the solution of the IPNS slightly drifts in the horizontal position as well as vertically. Fig. 4 presents an example for tolerance regions.

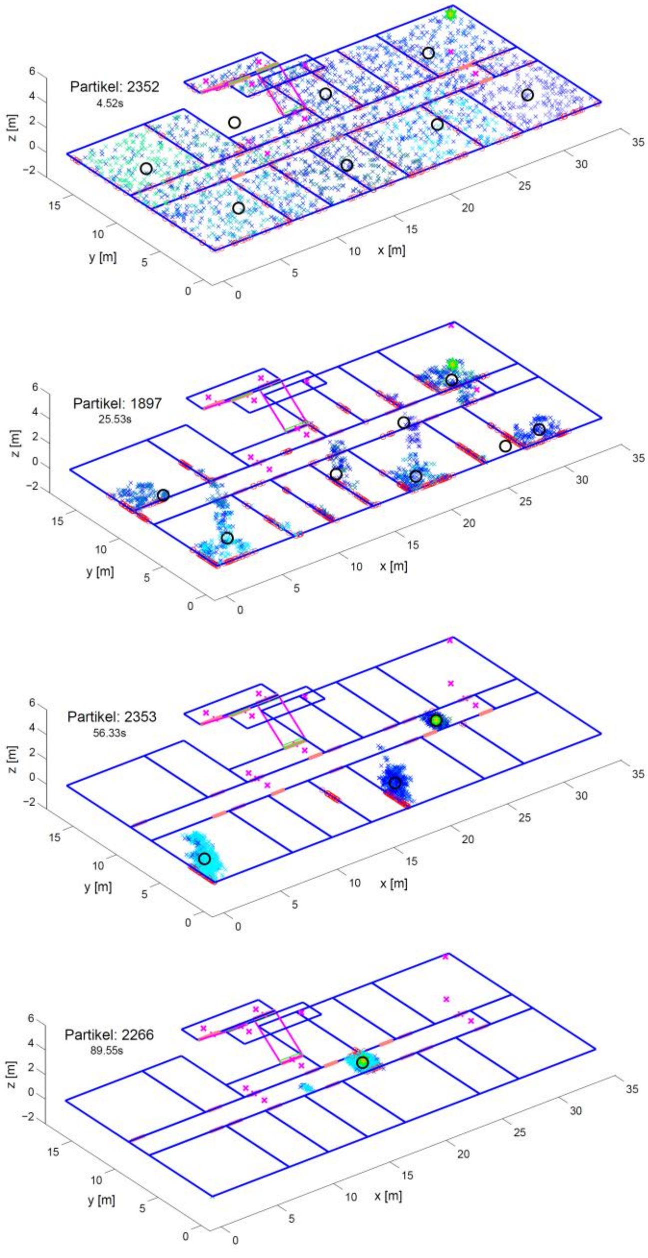

Fig. 5: *Simulation results of the map matching filter in an office building, particles are small, cyan crosses, clusters are black circles. The evolution of the particles is visualized from uniform distribution to the final plot which shows the convergence of the particles. Exact time stamps are indicated in each sub plot.*

VII. Positioning Based on Particle Clusters

The particles represent the probability of the actual position. The particles can be spread over a region and the distribution often is multimodal due to multiple possibilities at the beginning of a trajectory. To find the most likely position(s), the particle distribution is analyzed:

1. Initialization with a uniform distribution of particles over the map (position estimation not yet possible if starting point is unknown)
2. After some filter steps, several groups of particles may have been formed. At this point it is worth finding groups of particles using k-means cluster analysis. (multiple user positions are possible)
3. Finally, if only one cluster is left, a convergence point is reached and a good estimation of the position can be calculated using the center of gravity of the cluster. (one position found)

Fig. 5 shows simulation results of the map matching and the cluster analysis over time. The evolution of the particle cloud is shown with 4 subplots in time steps of about 25s, the exact time slot can be found at each subplot. The circles represent clusters, the blue crosses represent particles. After 89.55 seconds, a convergence point is reached as only one valid cluster is left. In the next section, the simulation is presented in detail.

VIII. Simulation Results

In this section results of the 3D map matching particle filter are presented. To see the theoretical positioning accuracy, a simple pedestrian navigation system simulation based on random step lengths on a given trajectory in an office building is implemented. The ground truth trajectory is calculated and after adding correlated noise in heading and step length a realistic simulation of a slightly drifting IPNS system is obtained. The drift of the pure IPNS solution is several meters after 100s. Now by using the proposed map matching algorithm, the error is not growing anymore, as soon as the estimation has converged (t=40s). Fig. 6 shows several simulation runs on the same trajectory. Of course the positioning error and convergence time heavily depend on the size of open rooms and the variety of the environment. In the present case, the long term accuracy is around 1.5m.

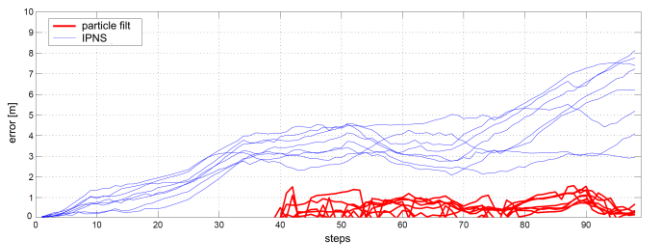

Fig. 6: Simulation results for several runs with ground truth: positioning errors from IPNS (blue) and map matching filter (red), convergence time around 40s

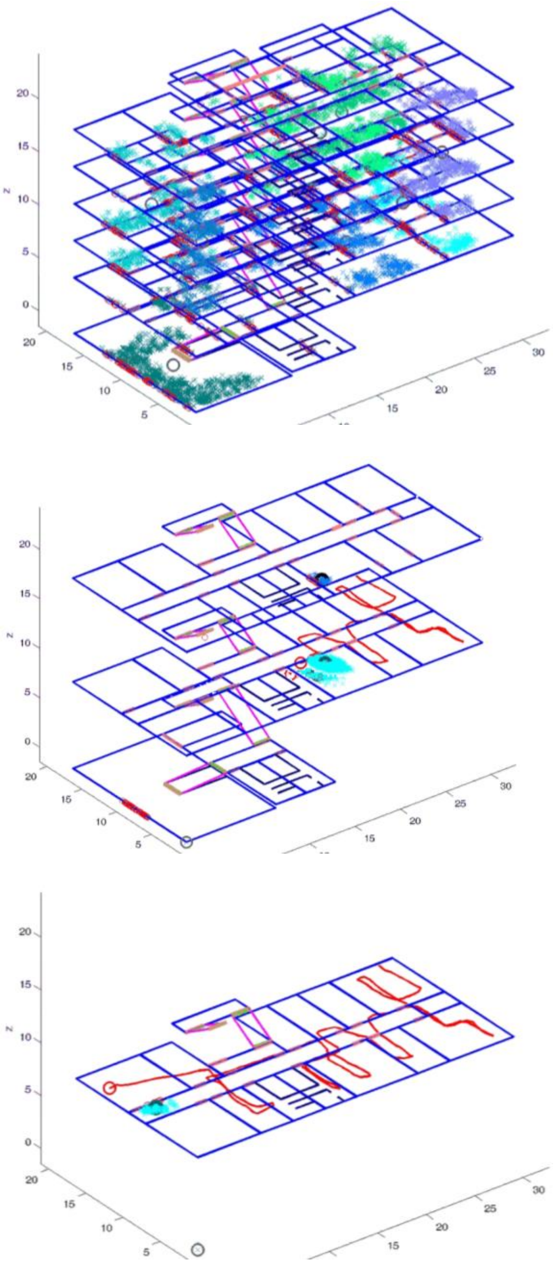

Fig.7: *Real data results of the IPNS (red) system and map matching particle filter (particles cyan, cluster black circle).The evolution over time is shown: At the beginning, the particles are distributed over the whole building on every floor. although the navigation solution(red) shows more and more drift, the track is successfully matched to the map (particle cloud).*

IX. REAL DATA RESULTS

In this section, the mapping results with real IPNS data will be presented. Fig. 7 shows the evolution of the drifting navigation solution of the IPNS system due to inertial drift in (red). The particle filter solution with the cluster point can be seen in black. Particles are plotted in cyan. At the beginning all floors of the building are possible, which is represented by the fact that the particles are distributed all over the building. After walking several corners, the walked path can only be realized on three remaining floors without collisions. When walking further, the final position is estimated as a point at the end of the corridor. Now the estimated IPNS trajectory is matched to the map successfully and without knowing on which floor the user has started, even the correct floor was found successfully. This map matching algorithm is now implemented on our real-time system and is calculated online. The computational burden is very low due to a fast implementation in C++: The particle filter runs 10 times faster than real time for the given office building.

X. PATH PLANING WITH THE 3D MAP

To show what else can be done with the presented 3D map, this section will propose how to guide a user inside a building from a known point to a known destination. Therefore an optimal path has to be planned with the lowest cost. When using the proposed 3D map from this paper, implementing an optimal path planner is a straight forward task.

For every object (room, stair, ladder) a dedicated cost per meter is saved. So when traversing objects, the sum of the cost can be calculated. If only a part of an object is crossed, the cost can also be estimated based on the percentage of the way through. Stairs and ladders for example have higher costs than rooms.

There are 3 classes of path planning [11]:

- Cell based path planning
- Roadmap based path planning
- Potential field path planning

We have chosen the roadmap based approach, because we have not too many points to watch in our maps as we are regarding only transitions. Furthermore, potential field path planning as well as a cell based approach need much more computation time because calculated grid needs to be small enough to find a way even in small rooms.

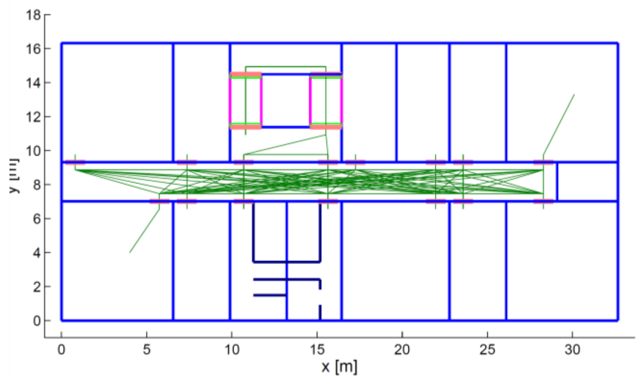

Fig. 8: *Global path planning graph, all door nodes are automatically connected with all others in a room. Path starts in the lower left room and stops in the upper right.*

We have decided to implement a road map based path planning [11]. We propose a 2 step graph:

- For *global path planning* between rooms a graph is established by automatically connecting each transition with all others (Fig.8)
- *Fine path planning* can be done in rooms that have additional obstacles (for example furniture or cabins, etc.). From this information, a visibility graph is established inside these rooms. (Fig. 9)

The global graph is established with nodes for every transition between two objects. Fig. 8 shows such a graph for one floor of an office building with an edge for every possible path through every object. The nodes are automatically introduced by connecting each transition with all other transitions in one room. For better visibility and representation the transitions are extended with 50cm lines inside and outside.

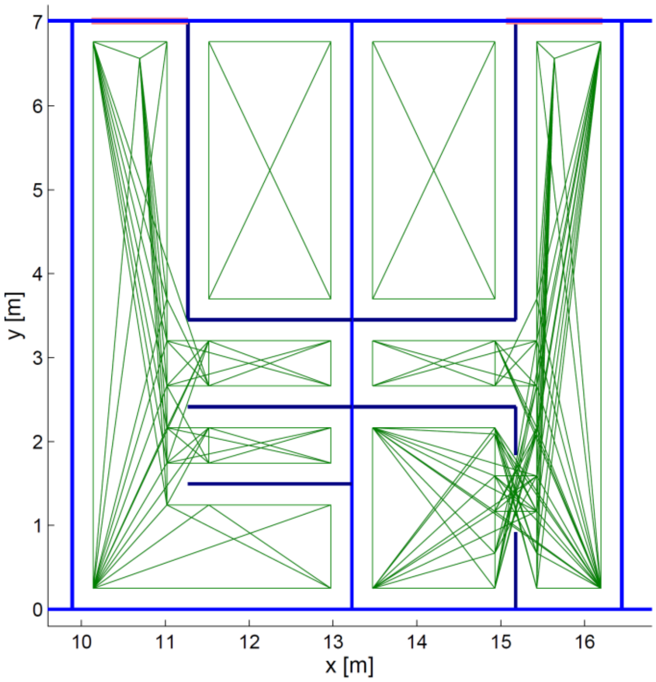

Fig. 9: *Fine path planning with a visibility graph, all obstacles(dark blue) are exposed as rectangles with 25cm clearance*

Often rooms have some obstacles which are not worth to be treated as an extra object but still have to be taken into account, for example toilet cabins or tables. Therefore we propose to use a fine path planning, based on a visibility graph, which is introduced in [11]. Fig. 9 shows the visibility graph of two rooms in an office building. All dark blue obstacle lines are treated as small rectangles with a clearance of 25cm so that the connections cannot cross them but they start at the corners of the rectangle. With this approach, it is possible to find a visibility graph for each room, that has additional obstacles. Inside this graph the shortest way out can be found without crossing obstacles. To reduce the computation burden, only if an object is used in the global path, then the fine planning is actuated.

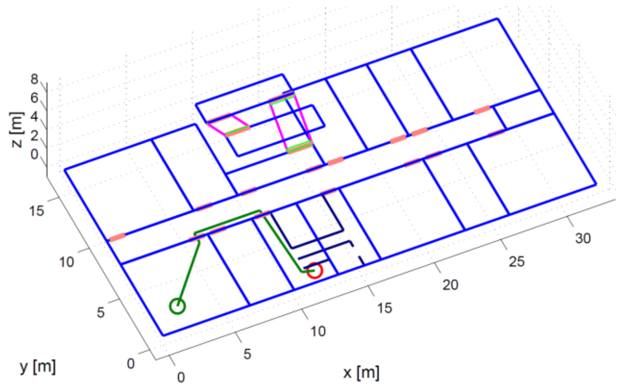

Fig. *10*: *Global and fine path planning in combination*

Finally to find the shortest path in both, global and fine graph we are using the A* algorithm [11] to save computation time. It finds one of the fastest ways but does not search the whole graph. Fig. 10 shows an example of a short path with global and fine planning in an office building: The global path planning starts inside a simple room, finds the way out through the only door to the corridor. Then the shortest way to the bath room is taken. Inside the bath room the fine path planning finds the destination which might be a toilet cabin without crossing cabin walls.

Fig. 11 shows another path planning example with multiple floors. The computational burden is mainly the calculation of the graph which must only be done once. The A* algorithm is implemented in C++ and the update rate needs not to be higher than 1Hz so even for large buildings, the calculation time is not a problem at all. For example for finding a path in the office building presented above from floor 1 to 5 over a distance of 200 m, a computation time of only 0.15s is needed.

This path planning is also used for online path planning and is updated at 1 Hz. But we do not have an audio user interface which tells the user where to go, but a graphical user interface is provided which is implemented in Qt.

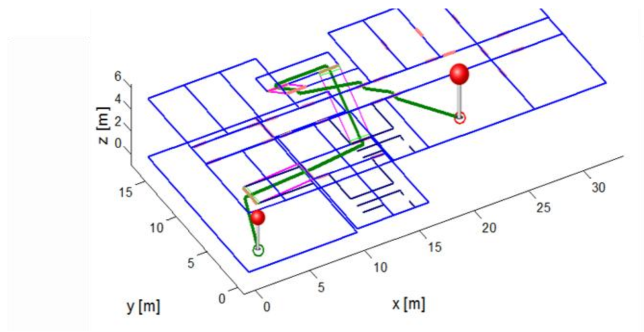

Fig. 11: Multi floor *path planning example. This is also used in our online implementation, the results are the same.*

XI. Conclusion

Map matching in a 3D environment like industrial plants or buildings can profit from a 3D map representation where ladders, elevators and stairs also can be taken into account and improve the map matching during the vertical movement. Especially vertical ladders have not been addressed in other papers. All these objects are all included in the presented new 3D map. It has been shown, that the presented particle filter based map matching algorithm can match a drifting user path into this 3D map without knowing a starting point. Simulation results as well as real data tests have been presented showing the benefit of the 3D map representation and the real time capability.

Furthermore, the 3D map can also be used for path planning and guidance inside a 3D building; this has also been demonstrated inside an office building and is implemented in our real time system.

For good map coverage also for unmapped buildings, we are currently working on the abstraction of our laser SLAM results [6] to extract rooms and staircases from the pure line segments from the laser readings.

References

[1] 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.

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

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

[4] 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.

[5] Krach, Bernhard, Roberston, Patrick, "Cascaded Estimation Architecture for Integration of Foot-Mounted Inertial Sensors," Proceedings of IEEE/ION PLANS 2008, Monterey, CA, May 2008, pp. 112-119.

[6] Kessler, C.; Ascher, C.; Trommer, G.: "Multi-Sensor Indoor Navigation System with Vision- and Laser-based Localisation and Mapping Capabilities", European Journal of Navigation Vol.9 Nr.3 December 2011.

[7] Shin, Seung Hyuck; Park, Chan Gook; Choi, Sangon. “New Map-Matching Algorithm Using Virtual Track for Pedestrian Dead Reckoning”. ETRI Journal, vol. 32, no. 6, pp. 891-900

[8] M. Khider, S. Kaiser, P. Robertson, M. Angermann, „ Maps and Floor Plans Enhanced 3D Movement Model for Pedestrian Navigation”. ION GNSS 2009.

[9] O. Woodman , R Harle : “Pedestrian Localisation for Indoor Environments”. Proceedings of the 10th international conference on Ubiquitous computing.

[10] Neil Gordon Branko Ristic, Sanjeev Arulampalam. “Beyond the Kalman filter : particle filters for tracking applications”. Artech House radar library. Artech House, Boston, Mass. [u.a.], 2004.

[11] J. Giesbrecht. “Global Path Planning for Unmanned Ground Vehicles”. Defence Research and Development Canada, 2004.