INDOOR 3D NAVIGATION AND POSITIONING OF VEHICLES IN MULTI-STOREY PARKING GARAGES

J. Bojja, M. Kirkko-Jaakkola, J. Collin, and J. Takala

Tampere University of Technology, Tampere, Finland.

ABSTRACT

We propose a novel method for three-dimensional navigation and localization of a land vehicle in a multi-storey parking-garage. In order to navigate or localize in 3D space we also need height information in addition to 2D position. Conventionally, an altimeter is used to get the floor level/height information. The solution presented in this paper uses low cost gyro and odometer sensors, combined with a 3D map by means of particle filtering and collision detection techniques to localize the vehicle in a parking garage. This eliminates the necessity of an altimeter or other additional aiding sources such as radio signaling. Thus the proposed solution can be used without any additional infrastructure devices. Other sources of information, such as WLAN signals, can be used to complement the solution if and when available.

Index Terms— Particle filters, dead reckoning, land vehicles, sensor fusion, indoor environments.

1. INTRODUCTION

Current navigation solutions employ one or several of the techniques based on GNSS satellites and receivers, WLAN devices, inertial sensors and video [1]. For clear- sky out-door navigation, GNSS alone will suffice for an accurate navigation solution. However in environments where GNSS is totally unavailable, such as indoor parking garages; the solution involves GNSS or some other means to get initial location information (e.g., near the entrance of the parking garage), which is then propagated in time using external sources, such as wireless radio devices, motion sensors and altimeters [2] [3].

In this paper, we study 3D map-matching in parking garages, a scenario different from the common map-matching problem in various senses. Firstly, in parking garages, GNSS cannot be relied on due to the heavy attenuation of satellite signals when penetrating concrete structures; therefore, one has to resort to using on-board motion sensors such as the odometer of the vehicle. Secondly, vehicle heading is less constrained than on roads and streets, which poses additional challenges when a gyroscope is used for heading estimation. Thirdly, positioning in multi-storey parking garages requires the use of 3D maps and knowledge on the altitude of the vehicle.

This paper proposes methods for achieving low cost and effective solution to such GNSS denied indoor multi-storey parking garage navigation. Many existing devices in the market, such as smartphones, are equipped with gyros, and all modern land vehicles (e.g., cars) have odometers. Given the nonholonomic constraints of vehicle motion and the initial location of the vehicle with respect to the target indoor, these two sensors and a detailed 3D map are sufficient to obtain an indoor 3D positioning solution.

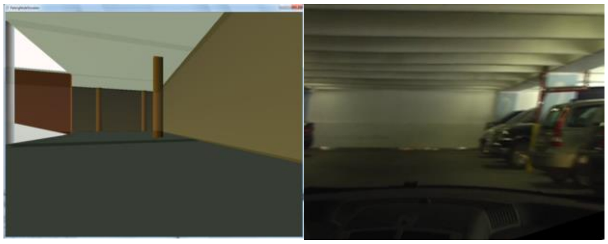
A 3D model as depicted in Fig. 1(a), representing the structural details of a real-world multi-storey parking garage shown in Fig. 1(b), is used as a 3D map and motion constraint in the solution which is based on particle filtering. In the filter, each particle is modelled as a separate 3D vehicle object which has approximately the same horizontal and vertical dimensions as a true vehicle.

The rest of the paper is organized as follows. Section 2 describes the related work while Sections 3 and 4 address the particle filtering and collision detection methods upon which the proposed navigation algorithm relies. Section 5 describes the measurement and experimental setup for testing and in Section 6 we demonstrate the approach with real-world sensor data obtained by driving a car in a parking garage. Finally, Section 7 concludes the paper.

2. RELATED WORK

Map-matching has been studied for decades, with the first implementations estimating the position of a vehicle along a known route [4]; an extensive description of the most common map-matching algorithms is given in [5].

Many solutions to the 3D indoor positioning problem have been proposed in the literature. Wagner et al. [6] used cascaded Kalman filters and “road link” matching for positioning vehicles in parking garages. Nowadays, a popular approach is to use a particle filter (PF); they are known to be well suited for positioning problems [7]. Fairfield et al. [8], and Kümmerle et al. [9] used the PF for simultaneous localization and mapping (SLAM) in a parking garage environment. However, the map information cannot be used as an efficient motion constraint in SLAM because the map is one of the unknowns. Leppäkoski



(a) (b)

Fig. 1. Multi-storey parking garage: (a) 3D model and (b) the real world garage.

et al. [10] proposed a pedestrian dead reckoning solution for indoor pedestrian navigation with detailed indoor maps as a motion constraint, inertial sensors as the primary source of information, and radio signals as assisting signals. This study showed that a very detailed 2D map, including even bookshelves, significantly improved the PF navigation solution in a 2D space.

Whereas we use, a detailed 3D map of the parking garage as a motion constraint to navigate in 3D space using on-board motion sensors. In our PF, we test each particle for collisions in a novel way, taking into account the dimensions of the vehicle which improves the accuracy of the map-matching. Instead of using altimeters or any other means for height estimation we use the 3D model ramps eliminating the necessity for external radio navigation updates, which are a common solution as in [11] when no map information is used.

3. PARTICLE FILTERING

Particle filtering is an approximation of the Bayesian filter where the posterior distribution $\mathbb{p}(x_n|y_{1,\ldots,n},x_0)$, with x_n denoting the state vector at time step n and $y_{1,\ldots,n}$ being the measurements, is characterized by a cloud of random samples, called particles, instead of, e.g., the moments of the distribution. The foremost benefit of this representation is the ability to operate on arbitrary distributions, thus making it possible to estimate, e.g., multimodal distributions which often cause divergence of Kalman-type and other filters that assume Gaussian distributions. Particle filtering is a Monte Carlo method and both its performance and computational complexity depend on the number of particles used. We use a PF variant called the bootstrap filter [12] where the so-called importance distribution is chosen to be the transitional prior distribution.

Suppose we have N particles x^i with nonnegative weights w^i, $i = 1, \ldots, N$. Each particle is a state vector containing the quantities that are to be estimated; in this study, the i^{th} particle is a 4×1 vector

$$x^i = \begin{bmatrix} H^i \\ E^i \\ N^i \\ U^i \end{bmatrix} \tag{1}$$

where H is the heading angle, E, N, and U denote East, North, and vertical coordinates, respectively. Particle filtering consists of two basic steps, i.e., prediction and updating. In the prediction phase, we draw particles from the transitional distribution. Assuming that nonholonomic constraints hold, the expected value of this distribution can be expressed as

$$\mathbb{E}(x_n^i|x_{n-1}^i) = \begin{bmatrix} H_{n-1}^i + \omega_n * \Delta t \\ E_{n-1}^i + \cos(H_n^i) * V_n * \Delta t \\ N_{n-1}^i - \sin(H_n^i) * V_n * \Delta t \\ U_{n-1}^i \pm \tan(P_n^i) * V_n * \Delta t \end{bmatrix} \tag{2}$$

where $\mathbb{E}$ is the expected value operator; P is the inclination angle of the surface on which the particle is located; V_n and ω_n are the measured speed and angular rate at the n^{th} time step, respectively; and Δt is the measurement interval. The covariance of the transitional distribution is determined based on e.g., the error characteristics of the motion sensors.

In the update step, the weights of the particles are modified according to the likelihood of a measurement given the state vector. In the case of the bootstrap filter, the update is done according to the simple proportion

$$w_n^i \propto \mathbb{p}(y_n|x_n^i) w_{n-1}^i. \tag{3}$$

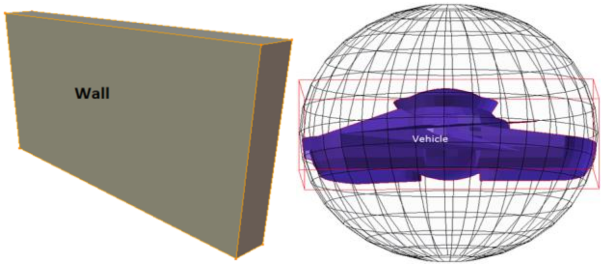
Due to the proportionality relation, the weights need to be normalized to sum to unity after updating. This way, it is straightforward to estimate the mean of the posterior distribution as the weighted average of the particles.

In this paper, we use the 3D map as a source of measurement updates according to the likelihood function

$$\mathbb{p}(y_n|x_n^i) = \begin{cases} 0 & \text{if the particle hit a wall} \\ 1 & \text{otherwise} \end{cases}. \tag{4}$$

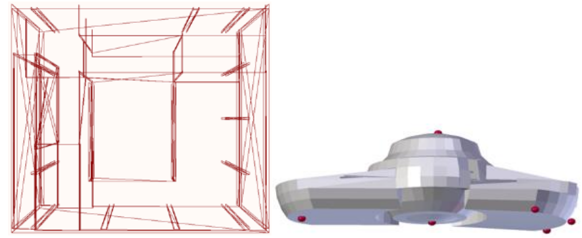
In other words, particles that collide are discarded. If other measurements are available, they can be incorporated into the estimation process by means of additional update steps with appropriate likelihood functions. It is obvious that discarding colliding particles leads to a situation where only a small fraction of the N particles are actually used for the state estimation; such a cloud of particles is obviously not a good approximation of a probability distribution and also causes a waste of computational resources if zero-weighted particles are propagated. This problem can be avoided by resampling the set of particles; in this procedure, a new set of N particles is drawn from the discrete probability distribution defined by the old particles and their respective weights, such that the newly obtained set of particles represent the same distribution as the old one, but with a full number of “alive” particles.

In this study, the map update likelihood is computed by modelling each particle as a vehicle with physical



(a) (b)

Fig. 2. (a) Target object with bounding box and (b) source object with bounding box and bounding sphere.



(a) (b)

Fig. 3. (a) Detailed structural model of a single floor parking garage and (b) vehicle model embeded with virtual sensor spheres (front, back, left, right, middle and top) used for collision detection.

dimensions and motion constraints, instead of a freely moving point mass. This significantly improves the accuracy of the map-matching because we can detect if one of the corners of the vehicle is touching a wall although the centre of mass is not. This modelling is discussed in detail in the following section

4. COLLISION DETECTION

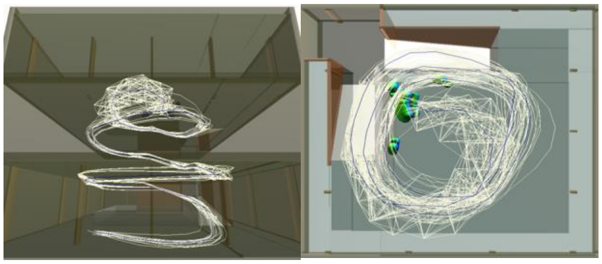
Collision detection is defined in this context as the ability to computationally detect if two or more objects are intersecting with each other. In general indoor parking garages consist of objects such as walls, railings, ramps, floors, roofs, and pillars that can be classified as target objects, and the vehicles which navigate in the garages as the source objects. Such object models can be obtained or designed easily with CAD software to reflect real-world parking garages and vehicles. The structural details of dimensions of the models play an important role, the more accurate the approximation of the model the better is the localization.

With the advances in computer aided design, programming tools, and computational capabilities of microprocessors, it has become feasible to represent and render such objects as software models on consumer devices. These software objects can be moved in a 3D space and collisions can be detected. Fig. 1(a) depicts our modelled garage space and Fig. 2 illustrates the wall and vehicle models. The modelled garage space forms our target space and objects in it are the target objects while the modelled vehicle object is our source object. When this source object is moved around in the target space, its movement is restricted to be on the path ways of the floor, ramp targets and it should not go through certain target objects such as walls. There arises a need to detect when source object is hitting such restricted objects to identify illegal moves. This is solved by collision detection methods [13]. In general, an intersection between a source and a target object is calculated using two simple boundaries encompassing the objects, namely the bounding box and the bounding sphere. These boundaries are as shown in Fig. 2. A sphere and a box intersect when any one of the points of the sphere falls within the range of points forming the bounding box. As can be seen from Fig. 2(b), box and sphere frames would not necessarily encompass the source object (vehicle model) as an exact fit. Therefore, for precise collision detection, in our implementation we have attached small spherical objects to the modelled source object, at different locations as depicted in Fig. 3(b). We will call these extra objects "virtual sensors" as they work similar to proximity sensors in real world. The number of such virtual sensors to be attached depends on the desired precision of required vehicle structure for collision detection and the trade-off between computational speeds and power consumed in such computations.

For every movement of the source object in the target space, the collision detection algorithm is applied between the source and the target objects. If a collision is detected, an appropriate action is taken, such as down-weighting the particle if it hits a wall or updating the position of the vehicle according to (2).

5. EXPERIMENTAL SETUP AND RESULTS

The proposed method was evaluated with field experiments. We measured the speed and heading rates of a passenger car by driving it in seven storey parking garage. A VTI SCC1300 MEMS gyro [14], a CarChip OBD II reader [15], and a DGPS system (NovAtel DL-4 Plus) [16] were used to measure the heading rate, speed data, and location information respectively. We have developed a software program that loads the modelled 3D replica of the garage and navigates the modelled vehicle within it. A separate Matlab script was used to process the sensor data samples and to feed them to the developed software. The software used this data to compute the successive locations, detect collisions, and move the modelled vehicle particles to the computed locations. The current location of the particles after the successive moves, and information about the target objects with which the particle has collided during successive moves are returned by the software to the Matlab script. Based on this output information appropriate action is taken by the script, e.g., adjusting the weight of a particle, or resampling when the count of the particles falls short.



(a) (b)

Fig. 4. Trajectories of the individual particles (white) and the resulting estimate of the path of the vehicle (blue): (a) side view and (b) top view.

(We used Matlab script as a faster development platform for the initial PF algorithm design and implementation, for real time navigation solution this script will be integrated with the software program).

The vehicle particle is propagated using a reference point located at the centre of mass of the vehicle. The overall structure is then defined by the virtual sensors attached around the vehicle. For checking the collision of the vehicle with the target objects, each virtual sensor is cross-checked for an intersection with the bounding frames of the target objects in the target space as depicted in Fig. 3 (a) and the collision detection information is used according to (4).

For detecting and navigating on ramps we have used a straightforward and simple method. The method assumes that the slope of the ramp is known beforehand and that a vehicle always touches the floors connecting the bottom edge and top edges of the ramp. Traversing a ramp always takes following sequence: once a vehicle reaches the floor connecting the start (end) of a ramp, collision detection is applied with that particular ramp and the vehicle; if a collision is detected, vehicle is tilted to the angle of the ramp and the vertical and horizontal position is altered accordingly in 3D space. This process is applied until the second floor connecting the end (start) of the ramp is reached, after which the tilt to the vehicle is removed and the vehicle will navigate in horizontal space. Ramp navigation can also be achieved without the knowledge of the slope of a ramp by using collision detection logic alone or using information from gyro measuring the tilt of the vehicle; nonetheless, we follow the approach described above.

Fig. 4 shows the trajectories of 60 particles with white lines and the resulting position estimate obtained as the mean of the particle distribution with a blue (dark) line. In Fig. 4(b) the green objects represent the vehicle structured particles. These particles were initialized with randomized heading and positions close to the initial position of the vehicle in the fourth floor, and they successfully tracked the vehicle to the seventh floor, the final vehicle parking place.

For 95% of the time, the mean of the particles was in the correct floor and with a maximum estimated position error

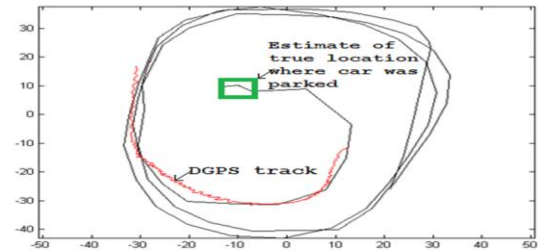


Fig. 5. Estimated trajectory of the vehicle projected into two dimensions. Trajectory of measured DGPS data on top floor, marked in red.

of 2.5 meters. 5% of the time the algorithm fails to find the actual path as particles move in wrong direction for uknown reasons during resampling stage; which will be addressed in future implementations. The performance could be improved by using a more accurate 3D map of the parking garage and enhancements to the algorithm; nevertheless, these results demonstrate the potential of the method. Fig. 5 shows the resulting position estimate projected into two dimensions; it can be clearly seen that the path on each floor is very much overlaping indicating the similarity of the floors and the accuracy of the solution. The shorter curve is the path on the last floor where the car was finally parked, the expected approximate true position is marked in green. For same last floor, track marked in red is the DGPS data track, measured during the return trip from top floor.

6. CONCLUSIONS

We have described a simple and efficient method for localizing a vehicle in a multi-storey parking garage. The method uses only gyro and odometer data, applies collision detection and particle filtering in 3D space. Unlike most of the existing techniques, external sources, such as radio signal beacons, altimeters are not needed. The proposed solution assumes that a detailed 3D structural model of the parking garage as shown in Fig. 1(a) or Fig. 3(a) is available in addition to initial position information when entering the garage. These assumptions are realistic and practical. Our experiments show that the solution works with reasonable accuracy using on-board sensors and the solution can be used to provide valuable assistance information to the driver, e.g., guidance to vacant parking spot.

The feasibility of the method was demonstrated in one type of parking garage. In the future, we plan to carry out more rigorous tests, with actual smartphone sensors, and extend the concept to tunnels and multi-level interchanges to enable accurate and seamless navigation services for drivers. The applicability of the proposed method is not limited to vehicular navigation; in fact, it could be used in any indoor environment with a navigating object capable of measuring its ground speed and angular rate, where navigation involves floor to floor navigation via ramps or staircases.

7. REFERENCES

[1] F. Chausse, J. Laneurit, and R. Chapuis, "Vehicle localization on a digital map using particles filtering," in *Proc. IEEE Intelligent Vehicles Symposium*, Las Vegas, NV, June 6-8 2005, pp. 243-248.

[2] J. Liu, R. Chen, Y. Chen, L. Pei, and L. Chen, "iParking: An Intelligent Indoor Location-Based Smartphone Parking Service," *Sensors*, vol. 12, no. 1, pp. 14612-14629, October 2012.

[3] C. Ascher, C. Kessler, M. Wankerl, and G.G. Trommer, "Dual IMU Indoor Navigation with particle filter based map-matching on a smartphone," in *Proc. International Conference on Indoor Positioning and Indoor Navigation*, Zurich, Switzerland, September 15-17 2010.

[4] R. French and G. Lang, "Automatic route control system," *IEEE Transactions on Vehicular Technology*, vol. 22, no. 2, pp. 36-41, May 1973.

[5] F. Gustafsson, U. Orguner, T.B. Schön, P Skoglar, and R. Karlsson, "Navigation and tracking of road-bound vehicles using map support," in A. Eskandarian, ed., *Handbook of Intelligent Vehicles*, Springer, London, pp. 397-434, 2012.

[6] J. Wagner, C. Isert, A. Purschwitz, and A. Kistner, "Improved vehicle positioning for indoor navigation in parking garages through commercially available maps," in *Proc. International Conference on Indoor Positioning and Indoor Navigation*, Zurich, Switzerland, September 15-17 2010.

[7] N. Yang, W.F. Tian, Z.H. Jin, and C.B. Zhang, "Particle filter for sensor fusion in a land vehicle navigation system," *Measurement Science and Technology*, vol. 16, no. 3, pp. 677, 2005.

[8] N. Fairfield, D. Wettergreen, and G. Kantor, "Segmented SLAM in three-dimensional environments," *Journal of Field Robotics*, vol. 27, no. 1, pp. 85-103, January 2010.

[9] R. Kümmerle, D. Hähnel, D. Dolgov, S. Thrun, and W. Burgard, "Autonomous driving in a multi-level parking structure," in *Proc. IEEE International Conference on Robotics and Automation*, Kobe, Japan, May 2009, pp. 3395-3400.

[10] H. Leppäkoski, J. Collin, and J. Takala, "Pedestrian navigation based on inertial sensors, indoor map, and WLAN signals," in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing*, Kyoto, Japan, March 25-30 2012, pp. 1569-1572.

[11] W. Chai, C. Chen, E. Edwan, J. Zhang, and O. Loffeld, "INS/Wi-Fi based indoor navigation using adaptive Kalman filtering and vehicle constraints," in *Proc. Workshop on Positioning Navigation and Communication*, Dresden, Germany, March 15-16 2012, pp. 36-41.

[12] N.J. Gordon, D.J. Salmond, and A.F.M. Smith, "Novel approach to nonlinear/non-Gaussian Bayesian state estimation," *Radar and Signal Processing, IEEE Proceedings-F*, vol. 140, no. 2, pp. 107-113, April 1993.

[13] T. Akenine-Möller, E. Haines, and N. Hoffman, *Real-Time Rendering*, third edition, A.K. Peters Ltd, Natick, MA, pp. 1045, 2008.

[14] VTI SCC1300 MEMS gyro, product information. [Online]. Available: http://www.muratamems.fi/products/gyroscopes/scc1300-combined-gyroscope-and-accelerometer

[15] CarChip OBD II reader, product information. [Online]. Available: http://www.carchip.com/Products/8226.asp

[16] DGPS system - NovAtel DL-4 Plus, product information. [Online]. Available: http://webone.novatel.ca/assets/Documents/Papers/DL4plus.pdf