Using geographical data and sonar to improve GPS localization for mobile robots

Karsten Bohlmann* Peter Biber† Andreas Zell*
**Computer Science Department, University of Tübingen, Tübingen, Germany* †*Robert Bosch GmbH, Schwieberdingen, Germany*

***Abstract*— Internet technology and the availability of large public knowledge bases should enable future autonomous systems to drastically improve their perceptual and cognitive capabilities with only inexpensive sensors. In this paper we investigate this aspect with respect to robot self-localization. We present a method to improve GPS-based localization of mobile robots using geographic data from a public database. From a cadastral map a basic map of a robot's working area is automatically created. A mobile robot is equipped with a low-cost GPS receiver and ultrasonic sensors. Then, a particle filter is used to fuse GPS position values and odometry data and to match sonar scan data with the a priori geodata map. The map is also updated with previously unknown environment features. The algorithm was tested in an outdoor environment with uneven terrain. Experimental results show considerable improvements in position estimation compared to using GPS alone.**

***Index Terms*— localization, geographical data, GPS, ultrasonic, rao-blackwellized particle filter**

I. Introduction and Related Work

The task of self-localization for autonomous robots in outdoor areas is more difficult than indoors for several reasons. The environment is unstructured, the underground cannot be assumed to be even, and odometry data are more imprecise. However, outdoor environments offer two advantages for robot localization: The GPS satellites provide a global system of active landmarks which can be used for absolute determination of the own position and, by exploiting the doppler effect of the received signals, even the own velocity and orientation [1]. The other advantage is the availability of preprocessed information about the potential robot workspaces in the form of geographic data. Geographic data (geodata) describe the positions and the characteristics of stationary objects and features on the earth surface. Large public geographic information systems have been created in recent times and data like cadastral maps are now available in electronic form for further processing.

In this paper we address the case of mobile robots operating in outdoor areas in the vicinity of buildings. In such spaces GPS accuracy typically suffers from multipath issues, with position errors up to 20m, or satellite reception can be disrupted completely. We use a prior map created from cadastral data of the area (Fig. 1). The main assumption is that buildings shown in this map always exist, but there might be other, previously unknown obstacles. As our objective is also to use low-cost sensory equipment, the presented approach is adapted to ultrasonic sensors and commercial-grade GPS receivers instead of laser scanners and differential GPS systems. We use the GPS receiver both as position and orientation sensor. The algorithm itself is based on Monte Carlo Localization as introduced by Dellaert et. al. in [2]. To reduce the number of particles needed for reliable localization we apply a Rao-Blackwellization [3, 4] on the particle filter by estimating the direction with subordinate Kalman filters. Performance and robustness of the filter are improved using Adaptive Resampling [5, 3] and Sensor Resetting Localization [6]. As applications for the resulting localization we present approaches to calculate the area covered by the robot's movements and an update of a probabilistic occupancy grid map.

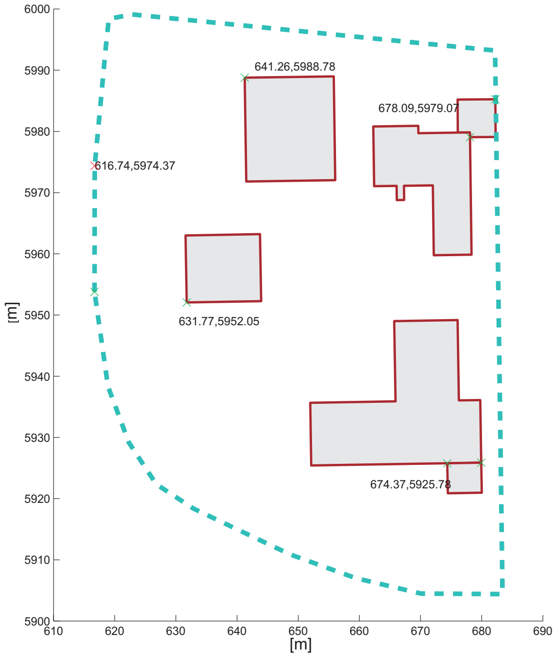


Fig. 1. The cadastral map of the testing area showing buildings, property border and coordinates in the Gauss-Krueger system (shortened)

A variety of techniques for improving GPS-based localization has been described over the past years. In [7] an extended Kalman filter is used to fuse GPS with inertia sensors and a laser scanner for matching with an incomplete map of the robot's surroundings. [8] uses an adaptive Kalman filter. [9] incorporates a map-based prior into a GPS-based localization. In [10] an approach using GPS in combination with a laser scanner and a map created from geographic data is presented.

II. Algorithm and sensor integration

A. Rao-Blackwellized Particle Filter

The core of the presented localization algorithm is a particle filter (PF) as introduced in [2] for robot localization. The distribution $p(s_t|z_{1:t}, u_{1:t})$ of the robot's pose s_t is estimated given the sequence of observations $z_{1:t}$ and odometry measurements $u_{1:t}$. The particle filter is implemented as sampling importance resampling filter (SIR) and approximates the pose distribution by a large set of M particles $s_t^{[i]}$, where each particle has an assigned importance weight $w_t^{[i]}$. The weights are calculated recursively, using the current observations $z_{j,t}$

$$w_t^{[i]} = \eta w_{t-1}^{[i]} \prod_{j=1:o} p(z_{j,t}|x_t^{[i]}) \tag{1}$$

where η is a normalization factor and o is the number of applied sensors.

A common problem of particle filters is their computional complexity, which increases with the number of particles and the dimension of the state vector. A technique to reduce the this complexity is the Rao-Blackwellization as described in [4]. The key concept here is to exploit dependencies in the state variables to reduce the dimension of the particle filter and therefore the complexity of the estimation task. The substates removed from the particle filter are determined analytically or using less costly filters. This approach can be applied if the variables inside the state vector s_t can be divided into two groups s_t^A and s_t^B so that

$$p(s_t|s_{t-1}) = p(s_t^B|s_{t-1:t}^A, s_{t-1}^B)p(s_t^A|s_{t-1}^A) \tag{2}$$

applies. Then the posterior turns according to [4] into

$$p(s_t^A, s_t^B|z_{1:t}, u_{1:t}) = p(s_t^B|z_{1:t}, u_{1:t}, s_{1:t}^A) \cdot p(s_t^A|z_{1:t}, u_{1:t}) \tag{3}$$

If (s^A, s^B) can be observed separately by (z^A, z^B) with the assumption $p(z_t^B|s_t^A) = p(z_t^B)$ follows with Bayes

$$\begin{aligned} p(s_t^B|s_t^A, z_{1:t}^B, u_{1:t}) \quad &= \\ = p(z_t^B|s^B, z_{1:t-1}^B, u_{1:t}, s_{1:t}^A) \quad & \cdot\frac{p(s_t^B|z_{1:t-1}^B, u_{1:t}, s_{1:t}^A)}{p(z_t|z_{1:t-1}, u_{1:t}, s_{1:t}^A)} \\ = \eta p(z_t^B|s^B, z_{1:t-1}^B, u_{1:t}) \quad & \cdot p(s_t^B|z_{1:t-1}^B, u_{1:t}, s_{1:t}^A) \end{aligned}$$

which leads to a Bayesian filter for s^B with the inputs (u, s^A).

We use this derivation to split the robot's pose state into orientation θ and position x. This is possible, as the motion model of a robot with differential drive is

$$\begin{aligned} \begin{pmatrix} x_t \\ y_t \\ \phi_t \end{pmatrix} &= \begin{pmatrix} x_{t-1} \\ y_{t-1} \\ \phi_{t-1} \end{pmatrix} \\ &+ \frac{1}{2} \begin{pmatrix} \cos(\theta_{t-1}) & \cos(\theta_{t-1}) \\ \sin(\theta_{t-1}) & \sin(\theta_{t-1}) \\ -2/L & 2/L \end{pmatrix} \begin{pmatrix} d_{l,t} \\ d_{r,t} \end{pmatrix} \end{aligned}$$

with L the wheel distance. Here, θ_t does not depend on (x_{t-1}, y_{t-1}) and therefore we can assume for the transition distribution

$$p(s_t|s_{t-1}) = p(\mathbf{x}_t|\theta_{t-1}, \mathbf{x}_{t-1})p(\theta_t|\theta_{t-1}).$$

This satisfies

$$\phi_{t+1} = \underbrace{1}_{A} \cdot\theta_t + \underbrace{\begin{pmatrix} -1/L & +1/L \end{pmatrix}}_{B} \cdot \begin{pmatrix} d_l \\ d_r \end{pmatrix}, \tag{4}$$

where θ is estimated using subordinate Kalman filters

The performance of the particle filter is improved using adaptive resampling. As shown in [3], the resampling step is only performed when the effective sample size, introduced by Liu [5] in the formulation by Doucet [4],

$$M_{eff} = \frac{1}{\sum_{i=1}^{M}(w^{[i]})^2} \tag{5}$$

falls below a threshold M_{th} . The argumentation here is that the larger standard deviation of the weights $w^{[i]}$ the poorer the approximation of the posterior is and the lower M_{eff}. [3] and [11] suggest as threshold a value of $0.5M$ and $0.6M$, which conforms to the experimental results of this work.

An issue with particle filters for localization is the possibility that the algorithm might fail entirely simply when due to bad previous sensor measurements no particles are left in the vicinity of the true pose. A simple measure against this behavior is the injection of new particles in the estimation. In [6] it is suggested to inject particles randomly in the entire working area or drawn from the inverse sensor distribution $p(s^{[i]}|z_t)$ based on the current observation z_t. The number of new particles M_{in} is determined, using the average particle weight $w_{avg} = \frac{1}{M}\Sigma_{i=1}^{M} w^{[i]}$, by

$$M_{in} = M \cdot \max(0, 1 - \frac{w_{avg}}{w_{th}})$$

with a heuristic threshold w_{th}. As in our setup we have an absolute sensor at our disposal, namely the GPS, this method avoids with $|\hat{s} - s| \leq |z_s - s|$ the divergence of the localization .

The structure of the complete localization algorithm is shown in Fig.2.

B. Sensor Modelling and Geographic Data

In our approach we apply a sensor fusion of data from a GPS receiver, ultrasonic sensors and odometry measurements. The GPS here is the only provider of absolute position information. A GPS measurement vector

$$\mathbf{z}_{gps} = (x, y, h, \theta, \sigma_{GPS}, v_x, v_y, v_z)$$

consists of position $\mathbf{x}_{gps} = (x, y)$ and altitude h, a direction θ, a velocity vector (v_x, v_y, v_z) and an error estimation value

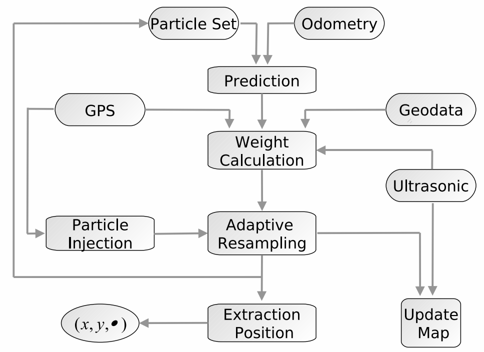


Fig. 2. Localization structure

σ_{GPS}. The position and velocity values are transformed into the coordinate system of the robot map. Our experiments showed that the GPS provides satisfactory direction values θ_{GPS} only for velocities $v_{bot} > 0.3m/s$. The covariance $\Sigma_{t,gps}$ is calculated for each measurement of the GPS by

$$\Sigma_{t,gps} = \begin{pmatrix} \sigma_t^2 & 0 \\ 0 & \sigma_t^2 \end{pmatrix}$$

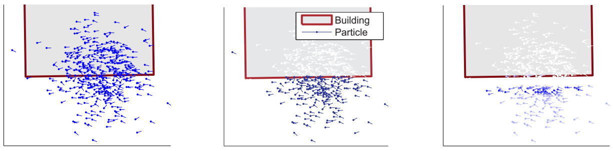
The likelihood $p_{GPS}(z_{gps}|\mathbf{x})$ for a particle position $\mathbf{x}$ by the GPS value $\mathbf{x}_{gps}$ is calculated using

$$p_{GPS}(z_{gps}|\mathbf{x}) = \exp(-\frac{1}{2}(\mathbf{x}-\mathbf{x}_{gps})^T\Sigma_{gps}^{-1}(\mathbf{x}-\mathbf{x}_{gps})) \quad (6)$$

In terms of the applied Bayesian localization the a priori geodata map is regarded as a sensor as well. Here the working area of the robot is represented using an occupancy grid map m. Each cell in the grid holds two values p_{occ} and p_{free} representing the current estimations for free space and obstacles. Buildings known from geodata have a probability as obstacle of $p_{occ} = 1$. The map m now works as a sensor which assigns weights to particles:

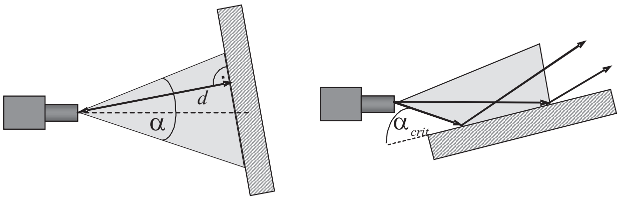
$$p(z_{geo}|\mathbf{x}, m) = 1 - p_{occ}(\mathbf{x})$$

The ultrasonic sensors are integrated in the localization using a *Likelihood Field* as suggested in [12] . The basic idea of this method is to determine the probability to detect an obstacle from a hypothetical robot pose (x, y, θ) at the distance z_d. The mechanics of the combined map and ultrasonic filter is illustrated in Fig. 3. At first the map filter sets the weights of all particles inside the building to zero. As the ultrasonic detects some object at a distance d, the robot cannot be closer to the building than this distance. However, it is possible that the robot encountered a previously unknown object with a probability $p_{unknown}$. Being in distance d from the building is still the best guess. Figure 4 shows the modelling of the ultrasonic sensor as a simple beam with opening angle α and a range r .



(a) Initial distribution (b) Map filter (c) Ultrasonic filter

Fig. 3. Weighting of particles using map and ultrasonic



(a) Distance measurement (b) Deflection

Fig. 4. Ultrasonic sensor modeling with regard to reflections away from sensor

III. Applications of the Localization Algorithm

A. Estimation of Robot Area Coverage

To test the results of the localization approach described above, two applications for robot self-localization were implemented. A typical task for robots in outdoor environments is to visit all reachable terrain in its assigned working area. Examples of these tasks include systematic exploration and many assignments in agriculture robotics.

The areas covered by the robot can easily be derived from the particle filter. Let $p_{xy}(t_{1:k})$ be the probability for the robot having covered the position $\mathbf{x} = (x, y)$ in the time period $[t_1..t_k]$ and $p_{xy}(t)$, the corresponding probabilities for t. The probability for never having been at (x, y) is

$$\overline{p}_{xy}(t_{1:k}) = \prod_{t=1:k} \overline{p}_{xy}(t) \quad (7)$$

From this and the relationship $p_{xy}(t) = 1 - \overline{p}_{xy}(t)$ the coverage probability can be formulated recursively as

$$p_{xy}(t_{1:k}) = (1 - p_{xy}(t_k))p_{xy}(t_{1:k-1}) + p_{xy}(t_k) \quad (8)$$

Based on the particle filter, $p_{xy}(t)$ is approximated as the sum of all weights of particles within a cell of width g in a grid map.

$$p_{xy}(t) \simeq \sum_i (w^{[i]}, s_t^{[i]} \in cell(x, y))$$

B. Mapping of Previously Unknown Objects

The same approach as described above could be used to update the map with previously unknown objects detected by the sonar scans. However, this would mean applying the sensor model of the ultrasonics to all particles at all times. Since this would be too time-consuming for online processing, we apply a different method to this task. Generally a particle filter delivers the state estimation as a set of possible states combined with weights. However, many applications of self-localization including map updates benefit from a concrete value $\hat{s}$ for the estimated position. To determine such a value several methods were suggested. A common method is to calculate the weighted mean of all particles

$$\hat{s} = \sum_{i=1}^{M} w^{[i]} s^{[i]} \tag{9}$$

This approach negates the main advantage of particle filter, namely the ability to deal with non-Gaussian distributions. In our experiments it often produced robot positions considered impossible, e.g. inside buildings. A simple method which showed very good results is to use the particle with the largest weight s^{max}. This method can be varied by using the weighted mean of the particles within a distance of less than ϵ from the particle with the highest weight:

$$\hat{s} = \sum_{i=1}^{M} w^{[i]} s^{[i]} : \left| s^{[i]} - s^{max} \right| \leq \epsilon \tag{10}$$

Based on these position estimations and the data from the environment scans, the area map of the robot can be updated. To every point of the path $\mathbf{x}_{t_n:t_m}$ of the best particle is a probability $p(\mathbf{x}_t)$, the confidence, by which the localization algorithm has estimated this position. If an obstacle at $\mathbf{x}_h$ is detected from the viewpoint $\mathbf{x}$, the quality of this measure depends on $p(\mathbf{x})$ being at this point. So the probability that an area is occupied is

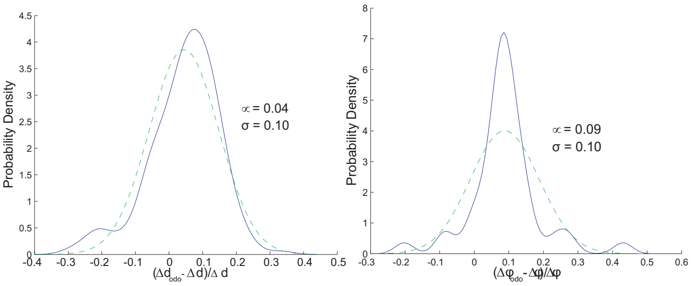
$$p_{occ}(\mathbf{x}_H) \sim \max_{t_n:t_m}(p(z_{occ}(\mathbf{x}_h)|\mathbf{x}_t)\, p(\mathbf{x}_t))$$

As $p(\mathbf{x}_t)$ the robust weight of the particle of the best path is used. The same applies to the free space $p_{free}(\mathbf{x})$. For the path the robot has driven is obviously $p(z_{free}(\mathbf{x}_t)|\mathbf{x}_t) = 1$. This approach is implemented using a grid map where each cell holds two values (p_{occ},p_{free}) as introduced in [13].

IV. Experimental Results

A. Sensor Characteristics

The presented algorithm was tested using a robot with differential drive. We first determined the characteristics of the used sensors. The testing area has uneven terrain and is irregularly covered with grass. To estimate the typical odometric errors on such a terrain we carried out test drives where the ground truth of the robot's position was provided by an infrared tracking system. Figure 5 shows the analysis of the gained data (104m straight drive and 42 turns in the range 45°-100°).



(a) Distance error

(b) Angular error

Fig. 5. Relative Odometry errors for a) straight drives b) turns. Continous lines show the distribution of measured errors, μ, σ are mean, standard deviation, the dashed line indicates the associated gaussian distribution

	Garmin 18 USB	Holux GR-213
Chipset	PhaseTrac12 (Garmin)	SiRF Star III (SiRF)
Frequency	1Hz	1Hz
Visible Satellites	7,9	8,6
Estimated Position Error (EPE)	14,8m	-
μ position error	2,1m	1,9m
σ position error	2,8m	1,2m
Max position error	12,5m	6,8m
Interface	Binary	Binary,NMEA
Remarks	No Output of raw GPS data	Altitude Hold Mode

TABLE I
Comparison of GPS devices, values are averaged from test series

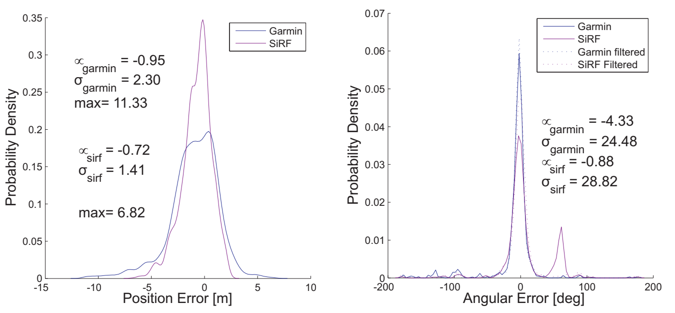
This illustrates the high influence of the uneven and partly slippery test terrain on the quality of the odometry measurements.

During this project we evaluated two commercial grade low-cost GPS receivers, a Garmin 18 USB and Holux GR-213. The chipset inside the Holux, a SiRF Star III, allows for setting an *Altitude Hold Mode* where an approximate altitude is specified in advance. The internal GPS algorithm of the device takes advantage of this a priori information to improve its position estimations. In our case, the altitudes of the test area are known from the geographical data. Both receivers are used *loosely coupled*, which means the localization uses the GPS position and orientation values estimated by the devices.

To collect GPS data, we mounted both devices on the robot. The robot was driven along different trajectories with markers on the ground for position reference. Test drives were made on 3 days to minimize the influence of GPS satellite constellations. The accuracy of the position reference is in the range $\pm 0.5m$.

Table I and Fig. 6 show the results of the GPS test series. For the orientation error we included only values taken at a robot speed $v_{bot} > 0.3m/s$, an experimentally determined threshold for usable orientation measurements by GPS.

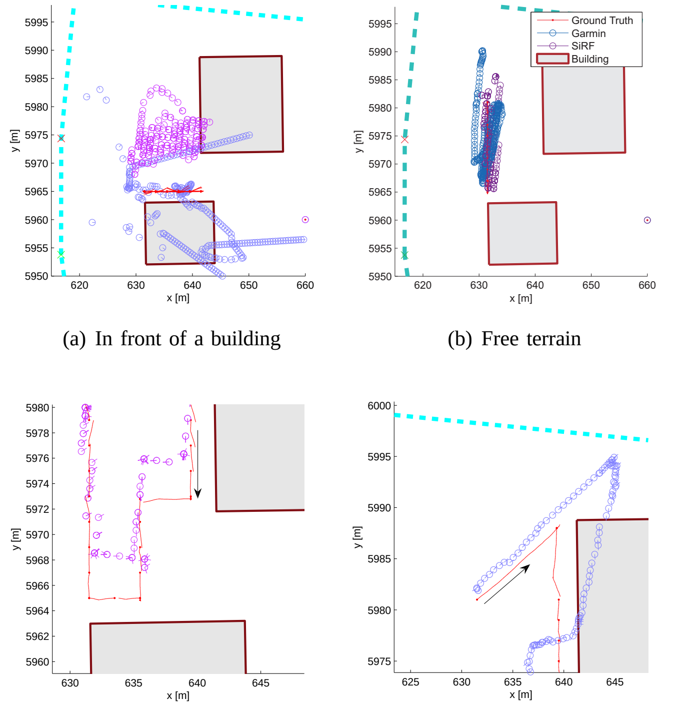
Figure 7 illustrates some of the typical difficulties when using GPS for more exact position estimates. Figure 7(a) shows the position error caused by reflections and shadowing in front of



(a) Error position

(b) Error orientation

Fig. 6. GPS Position and orientation errors



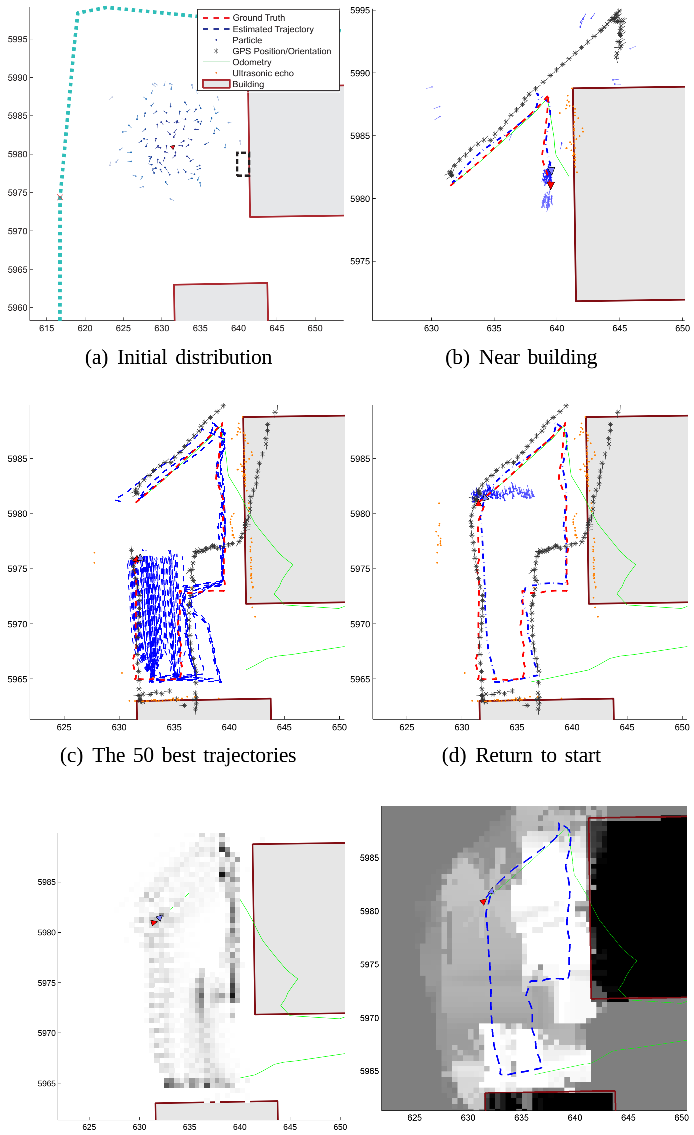
(a) In front of a building

(b) Free terrain

(c) Constant offset

(d) GPS receiver dynamics

Fig. 7. Typical GPS position errors near buildings



(a) Initial distribution

(b) Near building

(c) The 50 best trajectories

(d) Return to start

(e) Robot coverage map

(f) Detected probabilities for free space and obstacles

Fig. 8. Test drive with localization results. Dashed line: reference path, dashed: estimated trajectory, Stars: GPS positions.

buildings. Here the robot drove along a course 2m in front of a two-storied building. Figure 7(b) displays for comparison the GPS values in free terrain. This test run was executed right before Fig. 7(b), i.e. with a very similar satellite constellation. Although the GPS error is zero-mean in the long run (Fig. 6), GPS can exhibit a constant position error over a significant period of time. In Fig. 7c) an offset of 4m is visible. Fig. (d) points out the error caused by the estimation algorithm inside the GPS device itself: The robot (black arrow indicates direction of movement) stops and turns near a building, the GPS estimate follows the previous direction.

As environment sensors we used two ultrasonic modules of the type SRF08. This sensor has a range of 6m with a resolution of 1cm. The angle of the beam was measured with 60°. The modules were mounted on the robot in left and right orientation relative to the forward direction.

B. Localization results

To evaluate our localization algorithm, we performed multiple test drives. The robot was equipped with sensors and driven along different trajectories on the testing area. The logged sensor data were processed offline with the localization software. The reference path was determined using ground markers or an infrared tracking system. Fig. 8 illustrates such a test run. Figure 8a) shows the initial distribution of 100 particles around the first GPS value. Figure 8b) shows how the position estimation uses the building's wall as reference, although the GPS provides erroneous position estimates just because of the building. In the next figure (c) the trajectories of the 50 best particles are plotted, this shows that the algorithm

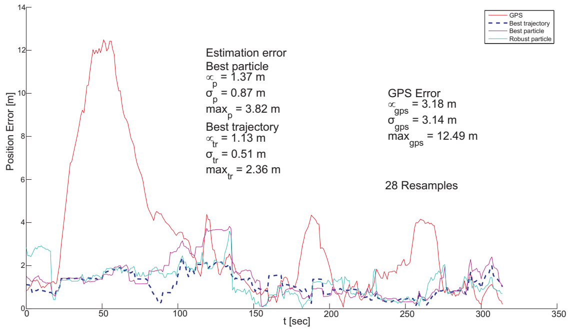


Fig. 9. GPS (red) and estimation errors (blue, dashed)

	particles	μ	σ	max
GPS	-	2.8	2.3	12.5
best trajectory	100	1.1	0.7	4.6
best particle	100	1.5	0.9	7.0
best trajectory	200	1.0	0.7	5.1
best particle	200	1.4	0.8	5.1
Rao-Blackwell disabled	100	1.9	1.2	9.5

TABLE II

ANALYSIS OF 3 TEST DRIVES, EACH RUN 5 TIMES THROUGH THE LOCALIZATION ALGORITHM. THE VALUES SHOW MEAN POSITION ERROR μ, STANDARD DEVIATION σ AND MAXIMAL ERROR, IN METERS)

has basically two assumptions for the robot's trajectory, which fan out after the right turn at the bottom building. As the GPS provided good results on free terrain, the estimation converged afterwards nearly exactly to the true position (Fig. (d)).

The results of the coverage algorithm and the map update illustrate the confidence of the algorithm in its own estimations. In e) a grid cell is the darker the higher the probability is for having visited this area. In the probability grid map in f) white indicates certain free space and black a certainly identify obstacle. Fields in 50% gray signify unknown or unsure. It is clearly visible that the path sections along the buildings are the best estimated parts of the robot trajectory. In the mid of the long building wall is a not previously known concrete block (marked dashed in Fig. 8a)), which was mapped correctly. Fig. 9 shows the GPS and estimation errors over time. Despite strong GPS errors (max. 12.5m, mean 3.2m) the estimated position stays in the range of $\pm 2m$ around the true position.

We tested our method on the collected data of two more test drives. The averaged results of the three test runs are presented in Table II. To verify the effectiveness of the Rao-Blackwellization, we implemented a plain particle filter without subordinate Kalman Filters. As can be seen in the last row of Table II, disabling the Rao-Blackwellization leads to significantly worse position errors.

V. CONCLUSION

In this paper we presented an efficient method to improve the precision of GPS-based localization for mobile robots. The method has been tested using inexpensive sensors and geographical data from a public database. Our experiments show that, using only two low-cost ultrasonic sensors, the resulting positioning accuracy is significantly larger than with GPS alone and that the algorithm is able to compensate for temporarily large GPS errors. The method takes advantage of the fact that the used sensors mutually supplement each other: Near buildings GPS often fails due to multi-path issues, there however the ultrasonic sensors detect environment features known from geographic data. We also showed that the number of particles, and thus the required CPU load, can be significantly reduced using the presented Rao-Blackwellization by estimating the orientation with subordinate Kalman Filters. Still problematic are the accuracy on open field and the robustness of the approach regarding large deviations from the basic map. In future work we will address these issues by integrating additional sensors and extending the algorithm.

REFERENCES

[1] E. D. Kaplan, ed., *Understanding GPS: Principles & Applications*. Artech House Publishers (February 1996), 1996.

[2] F. Dellaert, D. Fox, W. Burgard, and S. Thrun, "Monte carlo localization for mobile robots," in *Proceedings of the IEEE International Conference on Robotics and Automation*, vol. 2, pp. 1322–1328 vol.2, 1999.

[3] G. Grisetti, C. Stachniss, and W. Burgard, "Improved Techniques for Grid Mapping with Rao-Blackwellized Particle Filters," *IEEE Transactions on Robotics*, vol. 23, Issue: 1, no. 1, pp. 34–46, 2007.

[4] A. Doucet, N. de Freitas, K. Murphy, and S. Russell, "Rao-Blackwellised Particle Filtering for Dynamic Bayesian Networks," in *Proceedings of the Sixteenth Conference on Uncertainty in Artificial Intelligence*, pp. 176–183, Stanford, 2000.

[5] J. S. Liu and R. Chen, "Sequential monte carlo methods for dynamic systems," *Journal of the American Statistical Association*, vol. 93, pp. 1032–1044, 1998.

[6] S. Lenser and M. Veloso, "Sensor resetting localization for poorly modelled mobile robots," in *Proceedings of the IEEE International Conference on Robotics and Automation*, vol. 2, pp. 1225 – 1232, IEEE, 2000.

[7] S. Panzieri, F. Pascucci, and G. Ulivi, "An outdoor navigation system using gps and inertial platform," *IEEE/ASME Transactions on Mechatronics*, vol. 7, no. 2, pp. 134–142, 2002.

[8] G. Reina, A. Vargas, K. Nagatani, and K. Yoshida, "Adaptive Kalman Filtering for GPS-based Mobile Robot Localization," in *Proceedings of the IEEE International Workshop on Safety, Security and Rescue Robotics SSRR 2007*, pp. 1–6, 2007.

[9] S. M. Oh, S. Tariq, B. N. Walker, and F. Dellaert, "Map-based priors for localization," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, vol. 3, pp. 2179–2184 vol.3, 2004.

[10] M. Hentschel, O. Wulf, and B. Wagner, "A GPS and laser-based localization for urban and non-urban outdoor environments," in *Proceedings of the IEEE International Conference on Intelligent Robots and Systems IROS 2008*, pp. 149–154, 2008.

[11] M. Li, B. Hong, Z. Cai, and R. Luo, "Novel Rao-Blackwellized Particle Filter for Mobile Robot SLAM Using Monocular Vision," *International Journal of Signal Processing*, vol. Volume 3 Number 1, pp. 39–45, 2007.

[12] S. Thrun, W. Burgard, and D. Fox, *Probabilistic Robotics*. The MIT Press, 2005.

[13] A. Elfes, "Sonar-based real-world mapping and navigation," *IEEE Journal of Robotics and Automation*, vol. 3, no. 3, pp. 249–265, 1987.