Improved GPS Sensor Model for Mobile Robots in Urban Terrain

Daniel Maier* and Alexander Kleiner*

Abstract— **Autonomous robot navigation in outdoor scenarios gains increasing importance in various growing application areas. Whereas in non-urban domains such as deserts the problem of successful GPS-based navigation appears to be almost solved, navigation in urban domains particularly in the close vicinity of buildings is still a challenging problem. In such situations GPS accuracy significantly drops down due to multiple signal reflections with larger objects causing the so-called multipath error. In this paper we contribute a novel approach for incorporating multipath errors into the conventional GPS sensor model by analyzing environmental structures from online generated point clouds. The approach has been validated by experimental results conducted with an all-terrain robot operating in scenarios requiring close-to-building navigation. Presented results show that positioning accuracy can significantly be improved within urban domains.**

I. Introduction

Autonomous robot navigation in outdoor scenarios gains increasing importance in various growing application domains, such as reconnaissance, urban search and rescue, bomb disposal, transportation of goods, assistance for disabled people, driving assistance systems, and many more. Successful systems for GPS-based outdoor navigation have been demonstrated in the past, as for example during the DARPA Grand Challenge, and DARPA Urban Challenge. In these domains perturbed GPS readings are comparably seldom due to sufficient clearance around the vehicle. In contrast, there are other domains requiring mobile robots to navigate arbitrarily close to buildings, as for example when the task involves entering an unknown building through a geo-referenced entrance. Depending on the complexity of the environment, and the robot's ability to recognize targets, the quality of the estimated pose has to be sufficiently high. However, particularly in close vicinity of buildings, GPS accuracy significantly drops down due to multiple signal reflection on larger objects causing the so-called multipath error.

A commonly deployed technique to compensate for these errors is to fuse multiple sensor readings weighted by their confidence with a filter. In case of GPS, confidence is gained from the HDOP (Horizontal Dilution of Precision) value, a measure computed according to the geometric constellation of GPS satellites. Since HDOP does not model multipath propagation, overconfident

*Department of Computer Science, University of Freiburg, Georges-Köhler-Allee 52, D-79110 Freiburg, Germany, {maierd,kleiner}@informatik.uni-freiburg.de

measurements are fused by the filter leading to highly inaccurate position estimates particularly in urban terrain. Figure 1 (a) illustrates the multipath effect between two close buildings, and Figure 1 (b) plots the corresponding GPS error (blue line) together with the HDOP-based 1σ bound (red line) over time. As can be seen, conventional error bounds do not hold in this particular situation.
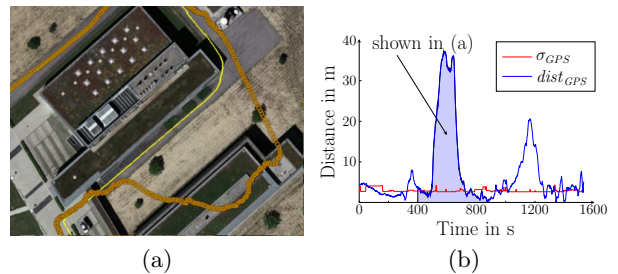

Fig. 1: (a) The effect of multipath when traveling between two buildings. The GPS measurements (orange dots) differ significantly from the true trajectory of the robot (yellow). (b) The corresponding GPS error (blue line) together with the HDOP-based 1σ bound (red line) over time.

In this paper we contribute a novel approach for incorporating multipath effects into the conventional GPS sensor model, and by this, increasing positioning accuracy within urban domains. This is carried out by generating online a 3D representation of the robot's surroundings, and detecting obstructed satellites by performing raytraces through this representation. The improved sensor model is also computed based on the HDOP, however, from non-obstructed satellites only. The approach has been applied with an Unscented Kalman filter (UKF) fusing data from odometry, IMU, and GPS, using the new uncertainty measure for improving pose estimation.

Experimental results presented in this paper were conducted with a telemax robot-based system that has been designed for the *TechX* challenge [1] hold 2008 in Singapore. Among other tasks, such as elevator handling, stair negotiation, and target detection, one critical task was about approaching the target building entrance via a stair, which required accurate pose estimation close to high building structures.

The remainder of this paper is structured as follows. In Section II related work is discussed. Section III describes the integration of GPS and odometry information, and Section IV introduces the improved sensor model. In Sec-

tion V experimental results are presented, and concluded in Section VI.

II. RELATED WORK

Common to most outdoor approaches is that they combine GPS and odometry or IMU measurements to mutually compensate the drawbacks of individual sensors. For instance, Panzieri et al. [2] fused GPS and IMU measurements using an Extended Kalman Filter (EKF) and surveyed the GPS performance. Sawabe et al. [3] compared the performance of EKF and a particle filter for outdoor GPS/IMU integration. Abbot and Powell [4] investigated the contribution of GPS, IMU, and odometry on the navigation performance of various land vehicles. They found that the overall position error is dominated by the quality of GPS. In this paper we focus on the assesment of the quality of GPS measurements for mobile robot localization.

Jones et al. [5] used two independent Kalman filters both incorporating GPS measurements. The outputs were merged to obtain a final position estimate. The GPS uncertainty was one of two constant values chosen by thresholding the HDOP value. We share the assumption that thresholding the GPS uncertainty is reasonable.

Several methods have been proposed to estimate the uncertainty of GPS measurements. One of the simplest ways to model GPS uncertainty is to assume a constant value, for instance used by Goel et al. [6]. They claim that it is not desirable to fuse GPS signals with gyroscope and odometry signals using an EKF as low GPS accuracy deteriorates the position estimate on small movements too much. Another method of modeling the GPS uncertainty is to dynamically adjust the covariance according to a parameter. For example, Thrapp et al. [7] used the number of available satellites.

Caron et al. [8] introduced contextual variables to define the validity of IMU and GPS. Fuzzy logic was used to weight sensor information in a Kalman filter. Capezio et al. [9] augmented the state space to include the non-zero mean bias of the GPS measurements in the estimation process. They assume that the bias changes slowly over time, which is not the case when multipath occurs.

Some approaches have been proposed to estimate or detect multipath errors. Giremus et al. [10] used a rao-blackwellized particle filter to detect abrupt changes in the state model. However, their approach was only evaluated with simulated measurements. Also, in reality multipath effects do not necessary reveal as abrupt jumps (see Figure 1). Brenneman et al. [11] used a statistical test on the signal of an array of GPS antennas to detect the presence of multipath. Common to approaches relying solely on the GPS receiver's output is that they cannot detect multipath presence if only the multipath replica but not the direct signal itself is propagated to the GPS receiver.

III. INTEGRATION OF GPS AND ODOMETRY

In this section the unscented Kalman filter for the integration of GPS and odometry data, and the conventional sensor model for GPS integration are described.

A. The Unscented Kalman Filter (UKF)

The UKF [12] allows the estimation of the state of a dynamic system given a sequence of observations and control inputs. Observations originating from different sensors are typically fused to obtain a more robust estimate of the state. Let x_t be the state, u_t the control input, and z_t the observation at time t. Furthermore, assume that state transitions are given by a function g and observations by a function h, both corrupted by Gaussian noise. That is,

$$x_t = g(x_{t-1}, u_{t-1}) + \epsilon_t, \qquad (1)$$
$$z_t = h(x_t) + \delta_t, \qquad (2)$$

where ϵ_t and δ_t are zero-mean Gaussian noise variables with covariance Q_t and R_t, respectively.

Because h and g are potentially non-linear, a Gaussian variable passed through either of them loses its Gaussian character. Hence, the UKF implements a stochastic linearization of g and h, called unscented transformation, around the pose estimate computed in the previous time step $t-1$. We choose the UKF over the Extended Kalman filter (EKF) because the latter can yield poor results in highly non-linear situations. The EKF computes a first-order Taylor approximation of the underlying models while the UKF has been shown to provide approximations that are at least as accurate as the second-order Taylor expansion [13]. Nonetheless, EKF and UKF share the same asymptotic complexity.

The results of the state estimation using the UKF depend strongly on covariances Q_t and R_t. If for instance R_t is very large, the residual of observation z_t and expected observation $\hat{z}_t$ are only marginally influencing the current state estimate. Furthermore, R_t can be regarded as weighting for different sensors when their measurements are fused to estimate a system's state. Therefore, a substantial challenge of designing a Kalman filter is to choose the covariance matrices appropriately.

B. Conventional GPS Sensor Model

The Global Positioning System (GPS) became a synonym for satellite-aided global localization systems. GPS currently consists of 31 satellites orbiting at about $20,000$ km providing global coverage with free access for civilian usage. Anywhere in the world at least six satellites are visible at all times. The signals of four satellites are necessary for a GPS receiver to estimate its position.

There are two major factors affecting the accuracy of GPS. a) The geometric constellation of the satellites represented by a numeric value termed Dilution of Precision (DOP). b) The errors in the pseudorange estimation, referred to as user-equivalent range error
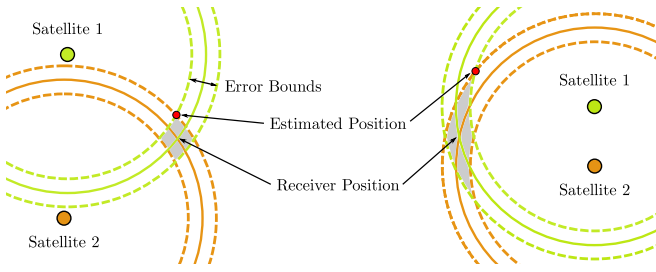
Fig. 2: Influence of satellite constellation on position estimate. For the right constellation the shaded area of potential receiver locations is larger than for the left one with the satellites further apart. Errors in the range estimate are magnified by poor constellations.

(UERE), which is typically expressed by its standard deviation σ_{UERE}. Therefore, the error of the position estimate is approximated by

$$\text{Positioning Error}(1\sigma) = \text{DOP} \cdot \sigma_{UERE}. \quad (3)$$

Standard deviation σ_{UERE} is affected by multiple sources. First, atmospheric effects that influence the propagation speed of the GPS signal. Second, transmitted positions of satellites (the *ephemeris*) can be inaccurate. Third, the satellites' internal clock may experience drift and noise and is subject to relativistic effects. Fourth, there are multipath effects in case reflected replica of the GPS signal are transmitted to the receiver. Whereas the first three error sources can mostly be compensated by using DGPS (Differential GPS), the fourth source particularly dominates the error in urban terrain.

The satellite constellation affects the accuracy of GPS position estimates. DOP is one approach to express the quality of the constellation. High DOP values indicate poor constellations while low values indicate desirable ones. Figure 2 illustrates a 2-dimensional example of the satellite geometry affecting the positioning error. The pseudoranges are given by the true distances (solid lines) to the corresponding satellites plus error bounds (dashed lines). The gray areas indicate the potential positions of the receiver depending on the error in the pseudorange estimation with the receiver being positioned at the intersection of the solid lines. Assume that the faulty pseudoranges were computed such that their intersection point is at the red dot, indicating the estimated receiver location. Although the distances to the satellites and the error bounds are equal for the left and the right constellation, their estimates for the receiver position differ considerably due to the difference of the constellation. Bad satellite constellations magnify errors in the pseudorange estimate and are indicated by high DOP values.

DOP values can be calculated for various levels according to the specific requirements of the task. For example, the Horizontal DOP ($HDOP$) captures the influence of

the satellite constellation on the position estimate in the horizontal plane, i.e., ignoring the vertical component, and is commonly used on unmanned ground vehicles (UGVs). Algorithm 1 describes the HDOP calculation according to [14]. From Equation 3 we obtain a model

---

DOP calculation for a set of n satellites:

1 $H = \begin{pmatrix} a_x^1 & a_y^1 & a_z^1 & 1 \\ \vdots & \vdots & \vdots & \vdots \\ a_x^n & a_y^n & a_z^n & 1 \end{pmatrix}$, where

$(a_x^i, a_y^i, a_z^i) = \texttt{sphere2cartesian}(\theta_i, \phi_i, 1)$

is a unit vector in direction of the i-th satellite

2 $M = (H^T H)^{-1}$, where M is symmetric with

$M = \begin{pmatrix} \sigma_x^2 & \sigma_{xy}^2 & \sigma_{xz}^2 & \sigma_{xt}^2 \\ \vdots & \sigma_y^2 & \sigma_{yz}^2 & \sigma_{yt}^2 \\ \vdots & \vdots & \sigma_z^2 & \sigma_{zt}^2 \\ \vdots & \vdots & \vdots & \sigma_t^2 \end{pmatrix}$

3 $HDOP = \sqrt{\sigma_x^2 + \sigma_y^2}$

Algorithm 1: Dilution of Precision

---

for determining the covariance of the GPS position in the horizontal plane. Therefore, DOP is replaced with the HDOP value and the whole expression is squared to obtain the covariance from the standard deviation. A slightly more sophisticated model reflecting both error amplitudes in East and North direction can be developed by:

$$R = \begin{pmatrix} \sigma_x^2 & \sigma_{xy}^2 \\ \sigma_{xy}^2 & \sigma_y^2 \end{pmatrix} \cdot \sigma_{UERE}^2, \quad (4)$$

where the σ-values are computed as in Algorithm 1 and denote the horizontal components of the error matrix M.

IV. Improved Sensor Model

The improved sensor model is computed from non-occluded satellites, where occlusions are detected by ray tracing on a local 3D map of the environment. This section describes the generation of local 3D maps, computation of the new HDOP model, and filtering of the covariance.

A. Map Generation

The presented method requires 3-dimensional point clouds in the local coordinate frame of the sensor. In general, these can be generated by any sensor, such as a 3D camera, stereo vision, or a rotating laser range finder, operating at a sufficient level of accuracy. In our implementation, a rotating laser range finder has been used (see Section V). These point clouds are then projected on the robot coordinate frame and accumulated to form a map of the environment. To preserve local consistency point clouds are aligned using the iterative closest point
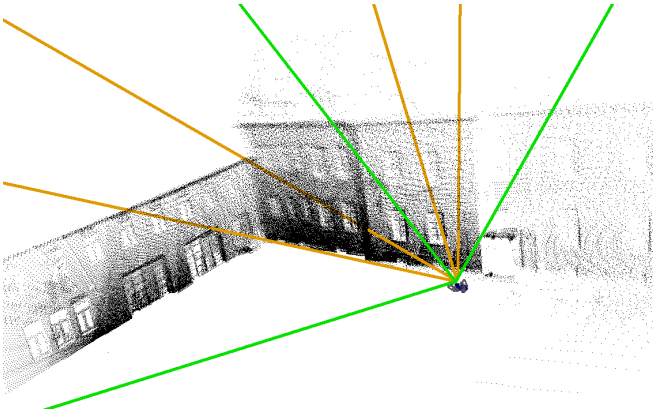
Fig. 3: A local map of the environment along with traces from the satellites to the robot. Green rays indicate free signal paths, orange rays indicate obstructed signal paths.

```
input  :  satellite position θ, φ (spherical),
          GPS antenna position (x, y, z), map M
          max ray length r_max, step size r_inc
output :  true if satellite's signal path is ob-
          structed, false otherwise
1  r = 0                      // Distance from antenna
2  while r < r_max do
3      (dx, dy, dz) ← sphere2cartesian(θ, φ, r)
4      q = (qx, qy, qz) ← (x, y, z) + (dx, dy, dz)
5      r_search ← determineSearchRadius(r)
6      Find neighbors N of q closer than r_search
7      if signal_path_obstructed(N,r) then
8          return true
9      r ← r + r_inc
10 return false
```

Algorithm 2: Ray tracing algorithm for detecting obstructed signal paths

(ICP) algorithm [15]. The projection is obtained as follows. Let P be a point cloud and ξ the 6-dimensional robot pose both at time t with $\xi = (x, y, z, \phi, \theta, \psi)^T$. $p \in P$ denotes a point in local coordinates $(p_x, p_y, p_z)^T$. Let $M_x(\alpha)$ denote the rotation around the robot's x-axis about α and analog for $M_y(\beta)$ and $M_z(\gamma)$. The projection w of p in the robot's coordinate frame is

$$w_p = (x, y, z)^T + M_x(\phi) \cdot M_y(\theta) \cdot M_z(\psi) \cdot p. \quad (5)$$

In an iterative process we obtain $W = \bigcup_{p \in P} w_p$. Figure 3 depicts a visualization of a thereby obtained map.

Given the point cloud representation and relative satellite locations, one can detect obstructed satellites if lines traced from the receiver towards satellites locations are blocked by objects. Note that relative satellite locations are computed with respect to the last global pose estimate of the robot.

Because ray tracing is a time consuming operation, the local map is maintained within a tree-like data structure accelerating nearest neighbor search. Therefore, point clouds are stored in ANN-trees (Approximate Nearest Neighbor [16]), which are approximated kd-trees allowing to tradeoff query speed and accuracy by a parameter $\epsilon \in \mathbf{R}$. For $\epsilon > 0$, ANN-trees approximate the nearest neighbor of a point p by a point q whose distance to p exceeds the distance to the true nearest neighbor by a factor of at most $(1 + \epsilon)$. The k-nearest neighbors of a query point can be enumerated in $O(k \log n)$, where n is the numbers of stored points.

B. New HDOP Calculation

With the local map represented as ANN-tree, we can easily identify blocked signal paths by using ray tracing as in Algorithm 2. The positions of satellites are reported by conventional GPS receivers in spherical coordinates θ and ϕ which are converted to unit vectors using the function spere2cartesian. The location of the antenna is obtained from the robot's position plus an offset.

The signal path of a GPS satellite is approximated by a set of query points in fixed intervals along a ray from the robot to the satellite. Obstruction is detected using a fixed-radius neighbor search around the query points. The search radius is determined by the function determineSearchRadius which increases with the distance to the robot to compensate for the decreasing resolution of the LRF. A bounded linear model was chosen to this end. After some experiments with the function signal_path_obstructed we finally found that a simple threshold on the number of neighbors yields reasonable results at the advantage of easy implementation and low error-proneness.

Once we know that the signal path of a satellite k is blocked, we remove it from the set of satellites. The remaining satellites form the set of visible satellites, S^{vis}. We use only the satellites from this set to calculate the uncertainty of a GPS measurement by applying Algorithm 1 to S^{vis}. Thereby, we obtain M_{vis} and $HDOP_{vis}$. These values can only be calculated, if $|S^{vis}| > 3$. Otherwise, we set the uncertainty to a high value R_{inf}. The obtained values indicate how desirable the visible satellites are distributed over the celestial sphere for a GPS position estimate. The UKF requires an areal measure of the uncertainty. Therefore, we define the uncertainty of the GPS measurements as

$$R_{Imp} = (M_{vis})_{2 \times 2} \cdot \sigma^2_{UERE}, \quad (6)$$

where $(M)_{2 \times 2}$ denotes the upper left 2×2 submatrix of matrix M. The standard deviation σ_{UERE} was experimentally determined as 3.75 m from collecting stationary measurements with good satellite constellation (low HDOP values) and free of multipath effects.

C. Covariance Filtering

According to Equation 6 covariance R_{Imp} grows as more as GPS signals are shadowed from multipath effects, and vice versa. However, due to an incomplete

mapping of the environment or larger vibrations of the 3D sensor, the structure of R_{Imp} can unsteadily change, e.g., causing a series of larger covariances interrupted by a series of smaller ones. This is clearly an undesired effect since in such situations overconfident measurements might be fused by the filter perturbing the pose estimate. Furthermore, we observed that receivers require recover time after they have been exposed to multipath situations. Hence, rapid changes in the covariance matrix have to be avoided, which is achieved by Algorithm 3.

In Algorithm 3 we apply two different thresholds,

> **input** : GPS uncertainty R
>
> **output**: improved GPS uncertainty R'
> classification GPS_{good}, GPS_{bad}
>
> **1** $\sigma_p \leftarrow \sqrt{R_{1,1} + R_{2,2}}$ `// ` $\equiv \sigma_{UERE}$ `hDOP`$_{vis}$
> **2** $R' \leftarrow R$
> **3** $R'(\sigma_p > \sigma_{thresh}) \leftarrow R_{inf}$ `// threshold ` R
> **4** **foreach** *interval iv with* $\sigma_p > \sigma_{smooth}$ **do**
> **5** $i \leftarrow iv(end) + 1$ `// apply smoothing`
> **6** $len \leftarrow min(\text{length}(iv) \cdot len_{scale}, len_{max})$
> **7** $R'(i : i + len) \leftarrow R_{inf}$ `// to successors`
> **8** $GPS_{bad} \leftarrow \text{find}(R' = R_{inf})$
> **9** $GPS_{good} \leftarrow \text{find}(R' < R_{inf})$
> **10** **return** R'

Algorithm 3: Filtering and inflation of R yielding R' and classification GPS_{good}, GPS_{bad}

---

where σ_{thresh} is the maximally accepted standard deviation of the GPS position error. If the error exceeds this value we set the uncertainty to a very high value R_{inf} to give the corresponding measurements virtually no weight in the filter. A second threshold affecting the standard deviation σ_p is σ_{smooth}, which activates smoothing. When observing a series of measurements with high uncertainties (exceeding σ_{smooth}), it is evident that the receiver is within a error-prone location. Then, uncertainties of the successor values are inflated to obtain smoothed uncertainties, and to respect the recovery time of the receiver. The amount of smoothing depends on the number of measurements with errors exceeding σ_{smooth}. During our experiments we set $R_{inf} = \text{diag}(600^2, 600^2)$, $\sigma_{thresh} = 6$, $\sigma_{smooth} = 9$, $len_{scale} = 0.2$, $len_{max} = 12$.

The result of Algorithm 3 can be seen as binary classification. GPS measurements with an R'_{Imp} value of R_{inf} are given virtually no weight in the state estimation process while the remainder of the measurements are considered as usable. To evaluate whether we inflated the right measurements we form two classes GPS_{good} and GPS_{bad}. In the experiments, we compare this classification to a reference classification based on the observed error.

V. EXPERIMENTAL RESULTS

Experiments presented in this section have been conducted with the all-terrain robot telemax from the *telerob* [17] company, designed for bomb disposal and reconnaissance missions. The robot is additionally equipped with a SICK LMS 200 scanner measuring at a field of view of 180° and a range of 80 m. The scanner is continuously rotated by a PowerCube device from *Schunk* to obtain 3D scans from the environment. The PowerCube rotates the LRF at constant rate of 90°/s around an axis parallel to the ground. Due to the high accuracy of the PowerCube device it is possible to generate point clouds by accumulating multiple range readings with respect to their corresponding scan and PowerCube angle. Accumulated scans from a half rotation of the PowerCube form a hemisphere with a radius of 80 m. As already mentioned above, point clouds taken at different time steps are consistently registered by the ICP algorithm taking odometry and IMU bearing as an initial guess. Odometry is obtained from the wheel encoders attached to the tracks of the robot. As IMU sensor an *Crossbow AHRS440* has been used. The telemax robot with sensor setup is shown in Figure 4 (a), and the PowerCube with laser scanner in Figure 4 (b).
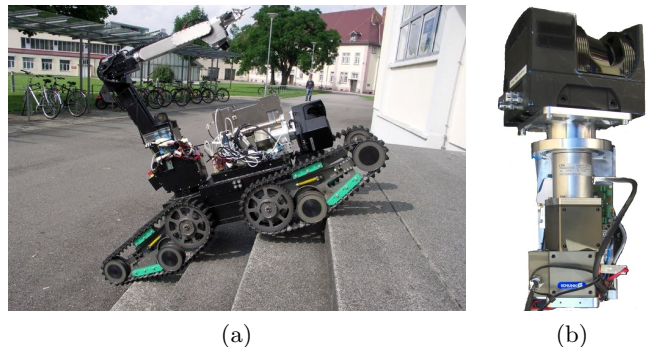

(a)   (b)

Fig. 4: (a) Telemax robot approaching a building entrance in an urban environment. (b) Rotating device mounted on the robot for taking 3D scans online during navigation.

Data was collected from odometry, IMU, the LRF, and the GPS module. The robot was manually controlled with a wireless controller. Most of the time, the robot was set to full speed which is ≈ 1 m/s. With this setup we recorded three logfiles which we refer to as *Log 014*, *Log 022*, and *Log 042*. In a time consuming process, we created ground truth by manually aligning data from a high-precision D-GPS receiver (Trimble GPS Pathfinder ProXT) with aerial images and the maps created from the LRF.

A. Classification

In the first experiment we evaluated the algorithm's ability to identify unusable GPS measurements caused by multipath effects. Therefore, we compared on the three logfiles the output of our algorithms with a reference classification. The reference classification was obtained by computing the distance to the ground truth $dist_{GPS}$ for every GPS measurement. A measurement was considered

Instances	Log 014		Log 042		Log 022	
Class. corr.	1178	79.3%	507	73.0%	1708	78.7%
Class. incor.	308	20.7%	188	27.0%	463	21.3%
#GPS Good	1102		538		1578	
#GPS Bad	384		157		593	
Total	1486		695		2171	
	Estimated as		Estimated as		Estimated as	
True Class	Good	Bad	Good	Bad	Good	Bad
GPS Good	0.79	0.21	0.69	0.31	0.84	0.16
GPS Bad	0.21	0.79	0.14	0.86	0.36	0.64

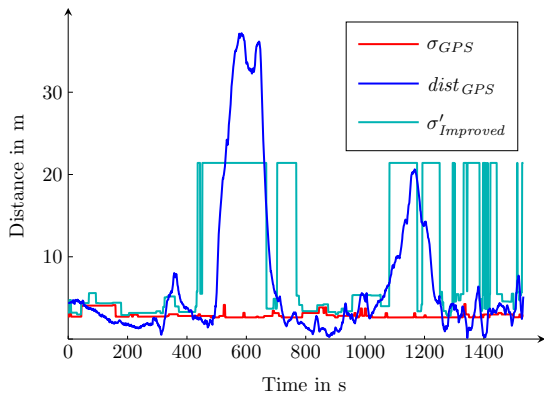TABLE I: Classification results including confusion matrix.

Fig. 5: Comparison of GPS error and uncertainties for *Log 014*. The improved model $\sigma'_{Improved}$ predicts the true error better than the conventional model σ_{GPS}. The $\sigma'_{Improved}$ has been cropped at 20 m.

unusable for navigational purpose if $dist_{GPS}$ exceeded the threshold σ_{thresh}, which we set to 6 m.

The comparison of the reference classification with the classification from our algorithm is summarized in Table I. We achieved a correct assignment for $70-80\%$ of the GPS measurements. The algorithm was able to identify at least 64% of the unusable GPS measurements.

Figure 5 illustrates the estimated GPS uncertainty for *Log 014* for the conventional uncertainty model (σ_{GPS}), and for the improved model ($\sigma'_{Improved}$) along with the true GPS error $dist_{GPS}$. The improved model clearly predicts the true GPS error more accurately than the default model.

Figure 6 visualizes the classification we obtained for *Log 042* and *Log 022*. GPS measurements assigned to GPS_{bad} by our algorithm are marked in red. All remaining measurements are marked in green. One can clearly see how the GPS measurements drift away from the yellow line indicating the ground truth. Our approach correctly identified those bad measurements. Once they converged back towards ground truth they were identified as good measurements again.

In Figure 6 the robot traveled from our campus (right)
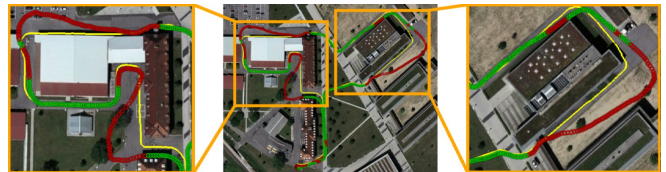
(a) *Log 042* Overview and Details
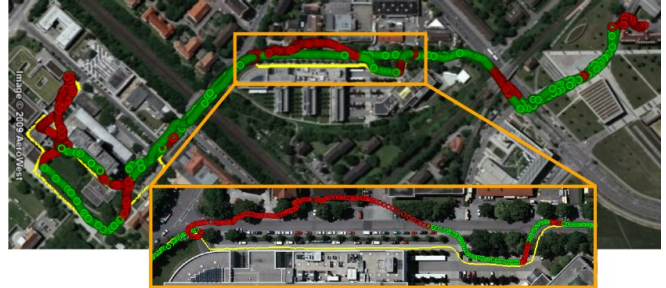
(b) *Log 022* Overview and Details

Fig. 6: GPS uncertainty classification. Red dots indicate GPS measurements classified as unusable for localization, whereas green dots indicate good measurements. The true robot trajectory is drawn in yellow.

to the university hospital's campus (left) and back again. During this experiment the robot traveled very close to buildings leading to GPS errors of up to 30 m due to multipath. Our method successfully identified these critical situations as shown in the figure.

B. Pose Error

In this section, we present an evaluation of applying the improved sensor model with an UKF filter. We compare the results from the conventional GPS sensor model with the results from the improved sensor model. Figure 7 shows the results from applying the filtering on the three logfiles. One situation worth mentioning is the difference in the position estimate for the multipath situation near the buildings on the upper right in Figure 7 (a) and (b). The robot drove on the pavement next to the top-most building in a counter-clockwise fashion. Pose tracking with the improved model (cyan line) approximated the actual trajectory of the robot comparably better than the tracking with the conventional model diverging up to 40 m away from the actual trajectory.

The results of the filtering are summarized in Table II. Most important to note is that the average and maximum errors in the position estimate as well as the standard deviation are significantly reduced by using the improved sensor model compared to the conventional model. The improvement of the average Euclidean distance error was between $21-38\%$, and the improvement of the orientation estimate between $10-27\%$.

VI. CONCLUSION

In this paper we presented a method of computing the uncertainty of GPS measurements with special attention to the multipath effect. The method is based on
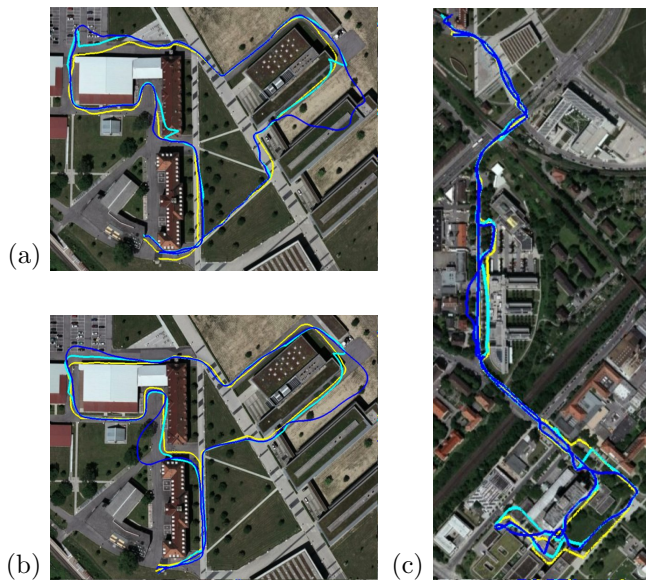
Fig. 7: Results from the UKF. Cyan lines visualize the results with the improved sensor model, blue lines with the conventional model. The true Robot trajectory is drawn in yellow.

Error	Odometry	Conv. Model	Impr. Model	Difference
		Log 014		
Position				
avg.	31.05 ± 17.49	7.09 ± 9.19	4.36 ± 3.76	38.5 %
max.	95.29	39.06	19.01	
Orientation				
avg.	0.35 ± 0.37	0.29 ± 0.35	0.21 ± 0.30	27.6 %
		Log 042		
Position				
avg.	55.24 ± 30.10	4.63 ± 4.98	2.90 ± 1.74	37.4 %
max.	116.90	22.21	10.70	
Orientation				
avg.	0.47 ± 0.53	0.34 ± 0.65	0.30 ± 0.64	11.8 %
		Log 022		
Position				
avg.	53.10 ± 34.14	5.90 ± 7.76	4.65 ± 4.77	21.2 %
max.	116.90	37.85	37.58	
Orientation				
avg.	0.28 ± 0.42	0.28 ± 0.45	0.25 ± 0.45	10.7 %

TABLE II: Errors for pose estimation. Units are m for positional errors and radian for orientational errors.

computing DOP values from only those satellites whose signal path is not obstructed. Obstruction of the signal path is determined using a 3D representation of the local environment which is generated online. The method was implemented and tested in realistic environments on the all-terrain robot platform telemax. Experimental results have shown that the improved sensor model outperforms the conventional model particular for tasks where mobile robots have to navigate close to buildings. We expect even better results when high precision IMUs, e.g., with fiber optics gyro, are used. In this case, an overconfident GPS error model has an even stronger impact on the resulting trajectory.

One difficulty we experienced while conducting exper-

iments is the influence of larger trees. They are detected correctly as objects in the scan. However, their influence on multipath effects seems to be less significant than the one of buildings. One solution to this problem is to distinguish trees from building structures and to treat them differently in the sensor model.

It seems to be obvious that there can be further improvements when utilizing the information about shadowed satellites on the receiver level directly, i.e., using pseudoranges only from satellites that are in line-of-sight for computing the GPS position. In future work we will consider to extend the approach towards such a tightly coupled Kalman filter, although it might then be applicable only to a few receivers which provide the necessary pseudorange information.

REFERENCES

[1] "DSTA TechX challenge," 2008, available: www.dsta.gov.sg/index.php/TechX-Challenge/.

[2] S. Panzieri, *et al.*, "An outdoor navigation system using gps and inertial platform," in *Advanced Intelligent Mechatronics, 2001. Proceedings. 2001 IEEE/ASME International Conference on*, vol. 2, 2001, pp. 1346–1351 vol.2.

[3] W. Sawabe, *et al.*, "Application of particle filter to autonomous navigation system for outdoor environment," in *SICE Annual Conference, 2008*, Aug. 2008, pp. 93–96.

[4] E. Abbott and D. Powell, "Land-vehicle navigation using gps," *Proceedings of the IEEE*, vol. 87, no. 1, pp. 145–162, Jan 1999.

[5] E. Jones, *et al.*, "Autonomous off-road driving in the darpa grand challenge," in *Position, Location, And Navigation Symposium, 2006 IEEE/ION*, 25-27, 2006, pp. 366–371.

[6] P. Goel, *et al.*, "Robust localization using relative and absolute position estimates," in *Intelligent Robots and Systems, 1999. IROS '99. Proceedings. 1999 IEEE/RSJ International Conference on*, vol. 2, 1999, pp. 1134–1140 vol.2.

[7] R. Thrapp, *et al.*, "Robust localization algorithms for an autonomous campus tour guide," in *ICRA*. IEEE, 2001, pp. 2065–2071.

[8] F. Caron, *et al.*, "Gps/imu data fusion using multisensor kalman filtering: introduction of contextual aspects," *Inf. Fusion*, vol. 7, no. 2, pp. 221–230, 2006.

[9] F. Capezio, *et al.*, "An augmented state vector approach to gps-based localization," in *Intelligent Robots and Systems, 2007. IROS 2007. IEEE/RSJ International Conference on*, 29 2007-Nov. 2 2007, pp. 2480–2485.

[10] A. Giremus, *et al.*, "A particle filtering approach for joint detection/estimation of multipath effects on gps measurements," *Signal Processing, IEEE Transactions on*, vol. 55, no. 4, pp. 1275–1285, April 2007.

[11] M. Brenneman, *et al.*, "An anova-based gps multipath detection algorithm using multi-channel software receivers," in *Position, Location and Navigation Symposium, 2008 IEEE/ION*, May 2008, pp. 1316–1323.

[12] S. J. Julier and J. K. Uhlmann, "A new extension of the kalman filter to nonlinear systems," in *Int. Symp. Aerospace/Defense Sensing, Simul. and Controls*, 1997, pp. 182–193.

[13] E. A. Wan and R. V. D. Merwe, "The unscented kalman filter for nonlinear estimation," in *Proceedings of Symposium 2000 on Adaptive Systems for Signal Processing, Communication and Control*, 2000, pp. 153–158.

[14] B. W. Parkinson and J. J. Spilker, *Global Positioning System: Theory and Applications*. The American Institute of Aeronautics and Astronautics, 1996, vol. 1, ch. GPS Error Analysis.

[15] P. J. Besl and N. D. McKay, "A method for registration of 3-d shapes," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 14, no. 2, pp. 239–256, 1992.

[16] S. Arya, *et al.*, "An optimal algorithm for approximate nearest neighbor searching," *J. ACM*, vol. 45, pp. 891–923, 1998.

[17] Telerob Gesellschaft für Fernhantierungstechnik mbH, available: http://www.telerob.de.