

Real-time Localization and Elevation Mapping within Urban Search and Rescue Scenarios

Alexander Kleiner and Christian Dornhege

Post Print

N.B.: When citing this work, cite the original article.

This is the authors' version of the following article:

Alexander Kleiner and Christian Dornhege, Real-time Localization and Elevation Mapping within Urban Search and Rescue Scenarios, 2007, Journal of Field Robotics, (24), 8-9, 723-745.

which has been published in final form at:

<http://dx.doi.org/10.1002/rob.20208>

Copyright: Wiley-Blackwell

<http://eu.wiley.com/WileyCDA/Brand/id-35.html>

Postprint available at: Linköping University Electronic Press

<http://urn.kb.se/resolve?urn=urn:nbn:se:liu:diva-72555>

Real-time Localization and Elevation Mapping within Urban Search and Rescue Scenarios

Alexander Kleiner

Institut für Informatik

University of Freiburg

79110 Freiburg, Germany

kleiner@informatik.uni-freiburg.de

Christian Dornhege

Institut für Informatik

University of Freiburg

79110 Freiburg, Germany

dornhege@informatik.uni-freiburg.de

Abstract

Urban Search And Rescue (USAR) is a time critical task. One goal in Rescue Robotics is to have a team of heterogeneous robots that explore autonomously the disaster area, while jointly creating a map of the terrain and to register victim locations, which can further be utilized by human task forces for rescue. Basically, the robots have to solve autonomously in real-time the problem of Simultaneous Localization and Mapping (SLAM), consisting of a continuous state estimation problem, and a discrete data association problem.

In this paper we contribute a novel method for efficient loop closure in harsh large-scale environments that utilizes RFID technology for data association and slippage-sensitive odometry for 2D pose tracking. Furthermore, we introduce a computational efficient method for building elevation maps by utilizing an Extended Kalman Filter for 3D pose tracking, which can be applied in real-time while navigating on rough terrain.

The proposed methods have been extensively evaluated within outdoor environments, as well as within USAR test arenas designed by the National Institute of Standards and Technology (NIST). Our results show that the proposed methods perform robustly and efficiently within the utilized benchmark scenarios.

1 Introduction

Urban Search And Rescue (USAR) is a time critical task. Rescue teams have to explore a large terrain within a short amount of time in order to locate survivors after a disaster. In this scenario, the number of survivors decreases drastically by each day due to hostile environmental conditions and injuries of victims. Therefore, the survival rate depends significantly on the efficiency of rescue teams. One goal in Rescue Robotics is to have a team of heterogeneous robots that explore autonomously, or partially guided by an incident commander, the disaster area. Their task is to jointly create a map of the terrain and to register victim locations, which can further be utilized by human task forces for rescue.

Basically, the robots have to solve autonomously in real-time the problem of Simultaneous Localization and Mapping (SLAM), consisting of a continuous state estimation problem and a discrete data association problem. Within disaster areas, these problems are extraordinarily challenging. On the one hand, state estimation is difficult due to the extremely unreliable odometry measurements usually found on robots operating on rough terrain. On the other hand, data association, i.e. to recognize locations from sensor data, is challenging due to the unstructured environment. Firemen of 9/11 reported that they had major difficulties to orientate themselves after leaving collapsed buildings. The arbitrary structure of the environment, and limited visibility conditions due to smoke, dust, and fire, prevent an easy distinction of different places. This problem is also relevant to standard approaches for SLAM, which recognize places by associating vision-based and LRF-based features. These extraordinary circumstances make it very hard to apply common techniques from robotics. Many of these techniques have been developed under strong assumptions, for example, they require polygonal structures, such as typically found in office-like environments (Gutmann and Schlegel, 1996; Grisetti et al., 2002) or depend on predictable covariance bounds from pose tracking for solving the data association problem by validation gating (Dissanayake et al., 2001).

In this paper we contribute a novel method for robust and efficient loop closure in large-scale environments that utilizes RFID technology for data association and slippage-sensitive odometry for 2D pose tracking. Furthermore, we introduce a computationally efficient method for building elevation maps by utilizing an Extended Kalman Filter for 3D pose tracking based on scan matching supported by visual odometry, which can be applied in real-time while navigating on rough terrain.

RFID tags have a worldwide unique number, and thus offer an elegant way to label and to recognize locations within harsh environments. Their size is already below $0.5mm$, as shown by the μ -chip from *Hitachi* (Hitachi, 2003), and their price is lower than 13 Cents (Alien-Technology, 2003). Passive RFID tags do not require to be equipped with a battery since they are powered by the reader if they are within a certain distance. Their reading and writing distance, which depends on the employed communication frequency, can be assumed to be within a range of meters.

Within the proposed approach, RFIDs are actively deployed by robots at adequate locations, as for example narrow passages that are likely to be passed. Displacements between these RFIDs are estimated by pose tracking, and utilized for globally optimizing the locations of the RFIDs by minimizing the Mahalanobis distance (Lu and Milios, 1997). Then, the whole robot trajectory is interpolated by using the globally corrected poses as constraint points. Pose tracking is carried out from wheel odometry and IMU (Inertial Measurement Unit) data. Since wheel odometry becomes arbitrarily inaccurate if robots navigate on slippery ground or have to overcome smaller obstacles, a method for slippage-sensitive odometry has been developed. The introduced method, which is designed for 4WD robot platforms with over-constrained odometry, i.e. four shaft-encoders instead of two, infers slippage of the wheels from differences in the measured wheel velocities. Inference is carried out by a decision tree that has been trained from labeled odometry data.

Besides the solution of the data association problem, the RFID-technology based approach

comes with three further advantages: First, in a multi-robot (Ziparo et al., 2007) or multi-human (Kleiner and Sun, 2007) exploration scenario, multiple maps can easily be merged into one consistent map by utilizing found correspondences from RFID tags registered on those maps. Furthermore, they can be utilized for a communication-free coordination of these robots (Kleiner et al., 2006; Ziparo et al., 2007). Second, RFID tags that have been put into the environment can be used in a straightforward manner by humans to follow routes towards victim locations, i.e. they do not need to localize themselves within a metric map. Third, RFID tags can be used by human task forces to store additional user data, such as the number of victims located in a room or information about hazardous areas.

Moreover, we propose an efficient method for building elevation maps in real-time, i.e. to map the environment while the robot is in motion. The method tracks the 3D pose of the robot by integrating the robot’s orientation, and the 2D pose generated from visual odometry and scan matching. Furthermore, the 3D pose is updated from height observations that have been registered on the map. Given the 3D pose, the height value of each map cell is estimated by a Kalman filter that integrates readings from a downwards tilted LRF. Due to the integration of the full 3D pose, the method allows to create elevation maps while the robot traverses rough terrain, as for example, while driving over ramps and stairs.

The introduced methods have been extensively evaluated within outdoor environments, as well as within USAR test arenas designed by the National Institute of Standards and Technology (NIST) (Jacoff et al., 2001). Our results show that the proposed methods perform robustly and efficiently within the utilized benchmark scenarios. Moreover, we show that RFID-SLAM is capable of closing large loops within a few seconds.

The remainder of this paper is structured as follows. In Section 2 we discuss related work. In Section 3 we introduce the experimental platforms utilized for the evaluation of the introduced methods. In the Section 4 we describe the approach for slippage-sensitive pose tracking on wheeled robots. The RFID technology-based SLAM approach is introduced in Section 5 and the real-time building of elevation maps is described in Section 6. Finally, we provide results from real world experiments in Section 7 and conclude in Section 8.

2 Related Work

Borenstein et al. introduced a method for improving the odometry on differential-drive robots (Borenstein and L., 1996). A method for odometry improvement and optimization of motor control algorithms on 4WD robots has been introduced by Ojeda et al. (Ojeda and Borenstein, 2004). They apply “Expert Rules” in order to infer the occurrence of wheel slip.

Inspired from the fundamental work by Smith et al. (Smith et al., 1988), early work on SLAM was mainly based on the Extended Kalman Filter (EKF) (Dissanayake et al., 2001), which updates the state vector after each measurement in $O(n^2)$. Based on the observation that landmark estimates are conditionally independent given the robot pose, Montemerlo et al. introduced *FastSLAM*, which reduces the computational complexity of EKF-based SLAM to $O(nk)$, whereas k is the number of robot trajectories considered at the same time (Montemerlo et al., 2002). The framework has been further extended to using evidence

grids (Haehnel et al., 2003; Grisetti et al., 2005). Thrun et al. introduced an approach following the idea of representing uncertainty with an information matrix instead of a covariance matrix (Thrun et al., 2004). By exploiting the sparsity of the information matrix the algorithm, called Sparse Extended Information Filter (SEIF), allows updates of the state vector in constant time. Another variant of SLAM, the *Treemap* algorithm, has been introduced by Frese (Frese, 2006). This method divides a map into local regions and subregions, whereas the landmarks of each region are stored at the according level of the tree hierarchy. Lu and Milios introduced a method for globally optimizing robot trajectories by building a constraint graph from LRF and odometry observations (Lu and Milios, 1997). Our method is closely related to their work, however, enables efficient route graph corrections by decomposing the problem into pose tracking, optimization, and interpolation. In contrast to incrementally full state updates performed by EKF-based methods after each observation, the decomposition reduces the computational requirements during runtime to a minimum, thus allowing the efficient optimization of even large-scale environments. Whereas existing methods typically rely on a high density of landmarks, the RFID-based approach is tailored for very sparse landmark distributions with reliable data association.

In connection with radio transmitters, the SLAM problem has mainly be addressed as "range-only" SLAM (Kehagias et al., 2006; Djugash et al., 2005; Kurth et al., 2003; Kantor and Singh, 2002) since the bearing of the radio signal cannot accurately be determined. RFIDs have already been successfully utilized for localizing mobile robots (Hähnel et al., 2004; Bohn and Mattern, 2004) and emergency responders (Kantor et al., 2003; Miller et al., 2006). Hähnel and colleagues (Hähnel et al., 2004) successfully utilized Markov localization for localizing a mobile robot in an office environment. Their approach deals with the problem of localization within a map previously learned from laser range data and known RFID positions, whereas the work presented in this paper describes a solution that performs RFID-based localization and mapping simultaneously while exploration. Also sensor networks-based Markov localization for emergency response has been studied (Kantor et al., 2003). In their work, existing sensor nodes in a building are utilized for both localization and computation of a temperature gradient from local sensor node measurements. Bohn and colleagues examined localization based on super-distributed RFID tag infrastructures (Bohn and Mattern, 2004). In their scenario tags are deployed beforehand in a highly redundant fashion over large areas, e.g. densely integrated into a carpet. They outline the application of a vacuum-cleaner robot following these tags. Miller and colleagues examined the usability of various RFID systems for the localization of first responders within different building classes (Miller et al., 2006). During their experiments, persons were tracked with a Dead Reckoning Module (DRM) while walking through a building. They showed that the trajectories can be improved by utilizing the positions of RFID tags detected within the building. While these map improvements have been carried out with only local consistency, the approach presented in this work yields a globally consistent map improvement.

Elevation maps are indispensable, particularly for robots operating within unstructured environments. They have been utilized on wheeled robot platforms (Pfaff et al., 2007; Wolf et al., 2005), on walking machines (Krotkov and Hoffman, 1994; Gassmann et al., 2003), and on car-like vehicles (Thrun et al., 2006; Ye and Borenstein, 2003). These methods differ in the way how range data is acquired. If data is acquired from a 3D scan (Pfaff et al., 2007; Krotkov and Hoffman, 1994), it usually suffices to employ standard error models, which

reflect uncertainty from the measured beam length. Data acquired from a 2D LRF, e.g. tilted downwards, requires more sophisticated error models, such as the compensation of pose uncertainty (Thrun et al., 2006), and handling of missing data by map smoothing (Ye and Borenstein, 2003). Furthermore, full 3D data processing is usually not possible in real-time since 3D data acquisition, as well as 3D data registration, is still time consuming, thus requiring interruptions of continuous navigation. In contrast to previous work, our approach deals with the problem of building elevation maps in real-time, allowing the robot continuous planning and navigation. Furthermore, we relax the assumption that the robot has to be situated on a flat surface while mapping rough terrain.

3 Experimental Platform

The work proposed in this paper has been extensively tested on two different robot platforms, a 4WD (four wheel drive) differentially steered robot for RFID technology-based exploration and mapping of large-scale environments (see Section 5), and a tracked robot for climbing and mapping of rough terrain (see Section 6). Figure 1(a) shows the tracked *Lurker* robot, which is based on the *Tarantula* R/C toy. Although based on a toy, this robot is capable of climbing difficult obstacles, such as stairs, ramps, and random stepfields, i.e. rough terrain simulated by an arrangement of vertically aligned blocks of wood. This is possible due to its tracks, which can operate independently on each side, and the “Flippers”, i.e. the four arms of the robot, which can be freely rotated at 360°. We modified the base in order to enable autonomous operation. First, we added a 360° freely turnable potentiometer to each of the two axes for measuring the angular position of the flippers. Second, we added touch sensors to each flipper, allowing to measure force when touching the ground or an object. Furthermore, the robot is equipped with a 3-DOF Inertial Measurement Unit (IMU) from *Xsens*, allowing drift-free measurements of the three Euler angles *yaw*, *roll*, and *pitch*, and two *Hokuyo URG-X004* Laser Range Finders (LRFs), one for scan matching, and one for elevation mapping, which can be vertically tilted within 90°. For feature tracking (Dornhege and Kleiner, 2006) we utilized a *Logitech QuickCam Pro 4000* web cam (Logitech, 2006).

Figure 1(b) shows the *Zerg* robot, a 4WD differentially steered platform, which has been completely hand-crafted. The 4WD drive provides more power to the robot and therefore allows to drive up ramps and to operate on rough terrain. Each wheel is driven by a *Pitman GM9434K332* motor with a 19.7:1 gear ratio and a shaft encoder. The redundancy given by four encoders allows to detect heavy slippage and situations in which the robot gets stuck (see Section 4). In order to reduce the large odometry orientation error that naturally arises from a four-wheeled platform, the robot is also equipped with an IMU from *XSens*. Moreover, the robot is equipped with a *Thermal-Eye* infrared thermo camera for victim detection, and also with a *Hokuyo URG-X004* LRF.

The active distribution of RFID tags is carried out by a custom-built actuator based on a metal slider that can be moved by a conventional servo (Figure 1(e)). The slider is connected with a magazine that maximally holds around 50 tags. Each time the mechanism is triggered, the slider moves back and forth while dropping a single tag from the magazine. The device is constructed in a way that for each trigger signal only one tag is released. A software module triggers the device at adequate locations, which are determined according to the

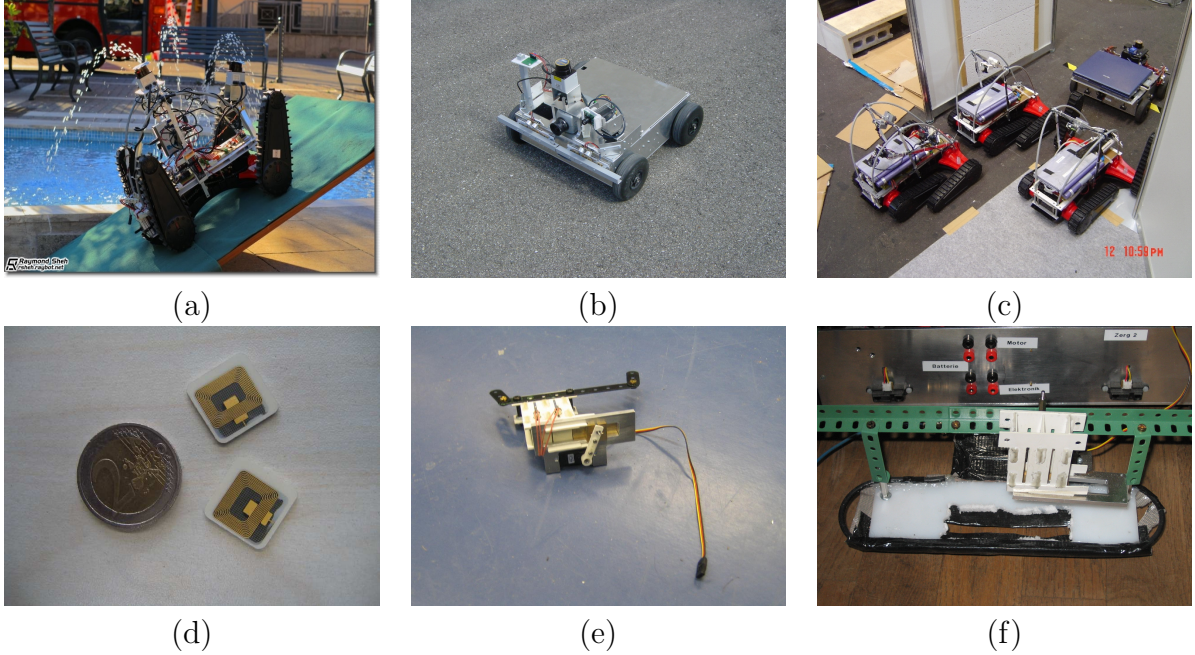


Figure 1. (a-c) Robots designed for rescue scenarios: (a) The *Lurker* robot, and (b) the *Zerg* robot. (c) The team of robots waiting for the mission start during the RoboCup competition in Osaka 2005 (Pictures a,c were taken by Raymond Sheh, and Adam Jacoff, respectively). (d-f) A novel mechanism for the active distribution of RFID tags. (d) The utilized RFID tags. (e) The mechanism with servo. (f) The mechanism, together with an antenna, mounted on a *Zerg* robot.

existing density of RFIDs, i.e. maintaining a maximal defined density of RFIDs, and also, if operating in indoor scenarios, with respect to the structure of the environment. For example, narrow passages, such as doorways and corridors, are likely to be passed by the robot. Hence, RFIDs are deployed with higher probability within these kind of environmental structures. The width of the free space surrounding the robot is computed from the distance between the obstacles located most left and most right to the robot, found on a line which goes through the center of the robot, and which is orthogonal to the robot's orientation.

The antenna of the reader is mounted in parallel to the ground, allowing to detect RFIDs lying beneath the robot. In order to enable the robot to perceive the deployment of a RFID, the deploy device has been directly mounted above the RFID antenna, forcing deployed RFIDs to pass through the antenna directly. We utilized Ario RFID chips from *Tagsys* (Figure 1 (d)) with a size of $1.4 \times 1.4\text{cm}$, 2048Bit RAM, and a response frequency of 13.56MHz. For the reading and writing of these tags, we employed a Medio S002 reader, likewise from *Tagsys*, which operates within a range of approximately 30cm while consuming less than 200mA. Figure 1 (f) shows the complete construction consisting of deploy device and antenna mounted on the *Zerg* robot.

4 Slippage-sensitive Wheel Odometry

In this section, we introduce a method for slippage-sensitive odometry that has been developed for increasing the reliability of pose tracking on wheeled robots. The two-dimensional

pose of the robot can be represented by the vector $l = (x, y, \theta)^T$. In order to represent uncertainties, the pose is modeled by a Gaussian distribution $N(\mu_l, \Sigma_l)$, where μ_l is the mean and Σ_l a 3×3 covariance matrix (Maybeck, 1979). Robot motion is measured by the odometry and given by the traveled distance d and angle α , likewise modeled by a Gaussian distribution $N(u, \Sigma_u)$, where $u = (d, \alpha)$ and Σ_u is a 2×2 covariance matrix expressing odometry errors. The pose at time t can be updated from input u_t as follows:

$$l_t = F(l_{t-1}, u_t) = \begin{pmatrix} x_{t-1} + \cos(\theta_{t-1})d_t \\ y_{t-1} + \sin(\theta_{t-1})d_t \\ \theta_{t-1} + \alpha_t \end{pmatrix}, \quad (1)$$

$$\Sigma_{l_t} = \nabla F_l \Sigma_{l_{t-1}} \nabla F_l^T + \nabla F_u \Sigma_u \nabla F_u^T, \quad (2)$$

$$\text{where } \Sigma_u = \begin{pmatrix} d\sigma_d^2 & 0 \\ 0 & \alpha\sigma_\alpha^2 \end{pmatrix} \quad (3)$$

and ∇F_l and ∇F_u are partial matrices of the Jacobian matrix ∇F_{lu} .

If the robot operates on varying ground, as for example concrete or steel sporadically covered with newspapers and cardboard, or if it is very likely that the robot gets stuck on obstacles, odometry errors are not linearly growing anymore, but are dependent on the particular situation. Therefore, we designed the *Zerg* robot with an over-constrained odometry for

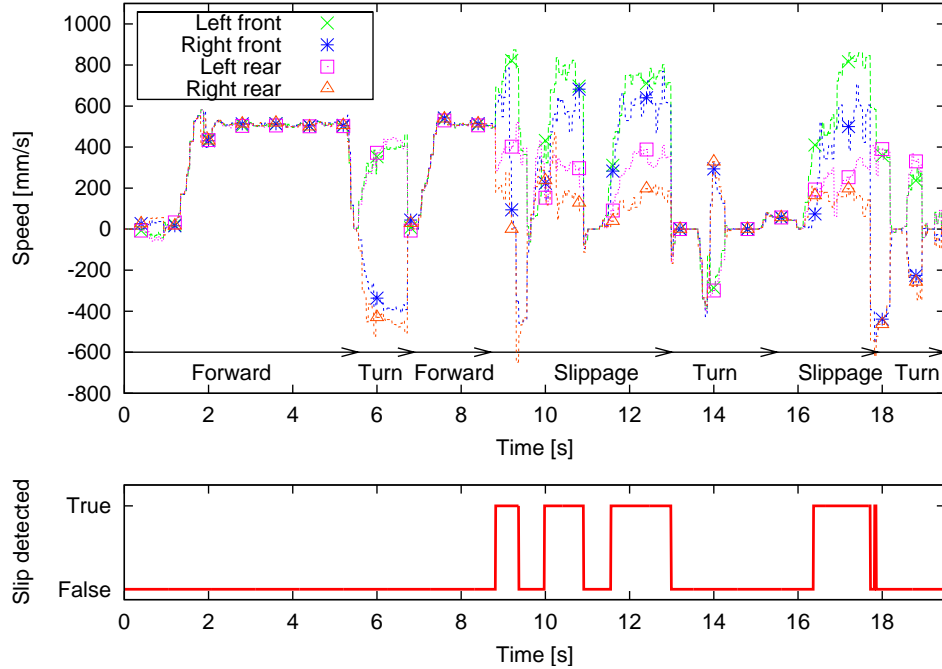


Figure 2. Slip detection on a 4WD robot: each line in the upper graph corresponds to the velocity measurement of one of the four wheels by shaft encoders. The black arrows indicate the true situation, e.g. driving forward, slippage, etc., and the lower graph depicts the automatic slip detection by the decision tree classifier, given the velocities as input.

the detection of slippage of the wheels by utilizing four shaft-encoders, one for each wheel. From these four encoders, we recorded data while the robot was driving on varying ground, and labeled the data sets with the classes $C = (slippage, normal)$. This data was taken to learn a decision tree (Quinlan, 2003) with the inputs $I = (\Delta v_{Left}, \Delta v_{Right}, \Delta v_{Front}, \Delta v_{Rear})$,

representing the velocity differences of the four wheels, respectively. Figure 2 depicts the slippage detection from velocity differences by the classifier.

Given the detection of slippage, the traveled distance d is computed from the minimum wheel velocity, e.g. $v_t = \min(v_{LeftFront}, v_{RightFront}, v_{LeftRear}, v_{RightRear})$, and the robot’s pose is updated according to Equation 3, however, with $\sigma_{d_{slip}}^2$, within covariance matrix Σ_u , in order to increase uncertainty in translation. Note that the rotation update needs not to be modified since the traveled angle α is measured by the IMU, which is not affected by wheel slippage. The values for σ_d^2 and $\sigma_{d_{slip}}^2$ have been determined experimentally. During extensive runs with slippage events, we recorded the true traveled distance, determined with scan matching, and the distance estimated by the odometry. The data set was labeled by the slippage detection and then has been utilized for computing the Root Mean Square (RMS) error for determining the variances σ_d^2 and $\sigma_{d_{slip}}^2$. We finally determined $\sigma_d = 0.816 \frac{cm}{m}$ and $\sigma_{d_{slip}} = 24.72 \frac{cm}{m}$. As we will show in Section 7, the improved odometry reduces the error significantly, while maintaining appropriate covariance bounds.

5 RFID Technology-based SLAM

In this section we introduce a method for globally optimizing robot trajectories based on RFID observations (Kleiner et al., 2006). This method requires as input the noisy odometry trajectory, and the set of RFIDs that have been observed on this trajectory. The uncorrected trajectory can be generated, for example, from slippage-sensitive pose tracking, as described in Section 4.

RFID tags have a world-wide unique encoded number, which significantly simplifies the data association problem in SLAM. When utilizing RFID tags, data association can reliably be achieved even if landmarks are only sparsely distributed in the environment. Therefore, it is possible to track the noisy robot pose over comparably large distances without landmark observation. RFIDs can be placed manually into the environment or actively distributed by the robots, e.g. by the deploy device described in Section 3.

The basic idea of the proposed approach is to compute local displacements between RFIDs by Kalman-based dead reckoning (see Section 4). From the correspondences of observed RFID tags and the estimated displacements, a globally consistent map is calculated by minimizing the Mahalanobis distance (Lu and Milios, 1997). The optimization method can be illustrated by considering the analogy to a spring-mass system. Consider the locations of RFIDs as masses and the measured distances between them as springs, whereas the uncertainty of a measurement corresponds to the hardness of the spring. Then, finding a globally consistent map is equivalent to finding an arrangement of these masses that requires minimal energy. The introduced method has the following advantages: On the one hand, optimized RFID locations are globally consistent given the estimated displacements between them. On the other hand, utilizing RFIDs as constraint points rather than odometry poses enables efficient and robust route graph optimization since the number of elements in the joint state vector is drastically reduced.

5.1 RFID graph optimization

The proposed method successively builds a graph $G = (V, E)$ consisting of vertices V and edges E , where each vertex represents an RFID tag, and each edge $(V_i, V_j) \in E$ represents a measurement \hat{d}_{ij} of the relative displacement $(\Delta x, \Delta y, \Delta \theta)^T$ with covariance matrix $\Sigma_{(\Delta x, \Delta y, \Delta \theta)}$ between the two RFID tags associated with the two vertices V_i and V_j , respectively. The relative displacement between two tags is estimated by a Kalman filter, integrating data from slippage-sensitive wheel odometry and the IMU (Section 4). If the robot passes a tag, the Kalman Filter is reset in order to estimate the *relative* distance \hat{d}_{ij} to the subsequent tag on the robot's trajectory.

We denote the true pose vectors of $n + 1$ RFID nodes with x_0, x_1, \dots, x_n , and the function calculating the true distance between a pair of nodes (x_i, x_j) as measurement function d_{ij} . The noisy measurement of the distance between two nodes (x_i, x_j) is denoted by $\hat{d}_{ij} = d_{ij} + \Delta d_{ij}$. We assume that the error Δd_{ij} is normally distributed and thus can be modeled by a Gaussian distribution with zero mean and covariance matrix Σ_{ij} . Loops on the trajectory are detected if the same RFID has been observed twice. A detected loop is modeled by a *pseudo* edge between the same RFID node, which will be described further in Section 5.3.

Our goal is to find the true locations of the x_{ij} given the set of measurements \hat{d}_{ij} and covariance matrices Σ_{ij} . This can be achieved with the maximum likelihood concept by minimizing the following Mahalanobis-distance:

$$\mathbf{x} = \arg \min_{\mathbf{x}} \sum_{i,j} (d_{ij} - \hat{d}_{ij})^T \Sigma_{ij}^{-1} (d_{ij} - \hat{d}_{ij}), \quad (4)$$

where \mathbf{x} denotes the concatenation of poses x_0, x_1, \dots, x_n . Moreover, we consider the graph as fully connected, and if there does not exist a measurement between two nodes, the inverse covariance matrix Σ_{ij}^{-1} is set to zero. If the robot's pose is modeled without orientation θ , e.g. because measurements from the IMU are sufficiently accurate, the optimization problem can be solved linearly by inserting $d_{ij} = x_i - x_j$ in Equation 4:

$$\mathbf{x} = \arg \min_{\mathbf{x}} \sum_{i,j} (x_i - x_j - \hat{d}_{ij})^T \Sigma_{ij}^{-1} (x_i - x_j - \hat{d}_{ij}). \quad (5)$$

Since measurements are taken relatively, we assume without loss of generality that $x_0 = 0$ and x_1, \dots, x_n are relative to x_0 . In order to solve the minimization problem analytically, Equation 5 can be rewritten in matrix form:

$$\mathbf{x} = \arg \min_{\mathbf{x}} (\hat{\mathbf{d}} - \mathbf{h}\mathbf{x})^T \mathbf{\Sigma}^{-1} (\hat{\mathbf{d}} - \mathbf{h}\mathbf{x}), \quad (6)$$

where $\mathbf{h}\mathbf{x}$ denotes the measurement function in matrix form with \mathbf{h} as an index function whose elements are $\{1, -1, 0\}$ and \mathbf{x} as the concatenation of pose vectors. Furthermore, $\hat{\mathbf{d}}$ denotes the concatenation of observations \hat{d}_{ij} , and $\mathbf{\Sigma}^{-1}$ denotes the inverse covariance matrix of \hat{d}_{ij} , consisting of the inverse sub-matrices Σ_{ij} . Finally, the minimization problem can be solved by:

$$\mathbf{x} = (\mathbf{h}^T \mathbf{\Sigma}^{-1} \mathbf{h})^{-1} \mathbf{h}^T \mathbf{\Sigma}^{-1} \hat{\mathbf{d}}. \quad (7)$$

and covariance of \mathbf{x} can be calculated by:

$$\mathbf{c}_{\mathbf{x}} = (\mathbf{h}^T \mathbf{\Sigma}^{-1} \mathbf{h})^{-1} \quad (8)$$

Equation 7 can be solved in $O(n^3)$ if the covariances Σ_{ij} are invertible. In practice, we assume that measurements are independent from each other, consequently the Σ_{ij} are given as diagonal matrices. Moreover, since many nodes in the graph are unconnected, most Σ_{ij}^{-1} are set to zero. Therefore, Σ is a sparse matrix and can in general be inverted efficiently. In order to utilize Equation 7 for the correction of the orientation angle θ , measurement equation d_{ij} has to be linearized by a Taylor expansion (Lu and Milios, 1997). Since the linearization leads to errors, the procedure has to be applied iteratively. We noticed during our practical experiments that five to six iterations are sufficient.

5.2 Trajectory interpolation

The corrected RFID network can be used as a basis for correcting the odometry trajectory. This is carried out by utilizing the corrected RFID locations as constraint points for the correction of the trajectory. Since these constraint points are already globally consistent according to the input data, it is not necessary to optimize the trajectory, augmented with these constraints, globally again. It turns out to be much more efficient to perform a local interpolation of the trajectory poses between the corrected RFIDs.

Given a sequence of corrected RFID locations, denoted by $r_{1:t}$, and an uncorrected trajectory, denoted by $x_{1:t}$, the corrected trajectory, denoted by $y_{1:t}$, is computed by interpolating each pose $x_k \in x_{1:t}$ between its next preceding and succeeding RFID location, respectively. In order to correct each pose x_k of the trajectory, we first determine the corrected locations r_i and r_j of the two closest RFIDs before and after the pose, whereas $j \geq k \geq i$, and the uncorrected poses x_i and x_j at corresponding time, respectively. Finally, the corrected pose is computed by:

$$y_k = x_k - \frac{(w_1 (x_i - r_i) + w_2 (x_j - r_j))}{w_1 + w_2}, \quad (9)$$

whereas weights w_1 and w_2 are computed by $w_1 = |r_j - x_k|$ and $w_2 = |r_i - x_k|$. Experimental results in Section 7 will show that this method together with the RFID graph optimization allows to efficiently and robustly correct large trajectories.

5.3 RFID observation model

Each time a loop has been detected on the trajectory, i.e. a RFID has been observed twice, a *pseudo* edge is added to the corresponding RFID node. We model this edge by accounting for the spatial expansion of the utilized RFID antenna. During our experiments, we utilized an antenna with a rectangular expansion mounted parallel to the ground, allowing to successfully detect RFID tags lying within this expansion beneath the robot. Unfortunately, it is not possible to tell the exact position of the detected RFID within this expansion. Hence, it is assumed that, in the average case, RFIDs are detected within the antenna's center, and that RFID detections can occur at arbitrary orientations of the robot. Therefore, we model the distance between identical tags by $\hat{d}_{ii} = (0, 0, \Delta\theta)$ and covariance matrix Σ_{ii} , whereas $\Delta\theta$ denotes the angle difference between the two pose estimates at the same RFID. The

covariance matrix Σ_{ii} is modeled in the following way:

$$\Sigma_{ii} = \begin{pmatrix} \sigma_{ant}^2 & 0 & 0 \\ 0 & \sigma_{ant}^2 & 0 \\ 0 & 0 & \sigma_{\Delta\theta}^2 \end{pmatrix}, \quad (10)$$

whereas σ_{ant}^2 reflects the size of the antenna, and $\sigma_{\Delta\theta}^2$ the uncertainty of the angular difference, which has to be linearized. In our implementation, we have chosen, according to the size of the antenna, $\sigma_{ant} = 15cm$.

6 Building Elevation Maps In Real-Time

In this section we describe a Kalman filter-based approach for building elevation maps by integrating range measurements from a downwards tilted LRF, whereas the map is incrementally build in real-time while the mobile robot explores an uneven surface. The motivation of the presented approach is to provide a basis for enabling the robot to continuously plan and execute skills on rough terrain. Therefore, the method has to be computational efficient and capable of building elevation maps that are sufficiently accurate for structure classification and behavior planning, as we have already demonstrated within another work (Dornhege and Kleiner, 2007). We argue that building globally consistent elevation maps, e.g. by loop closure, is computationally hard within large-scale environments since it requires to maintain the whole map in memory. Therefore, the goal of the proposed method is to build maps that are locally consistent within the vicinity of the robot for continuous planning and navigation. In a further processing step, e.g. offline, a globally consistent map can be generated by merging locally consistent map patches according to a globally consistent route graph, as for example, computed by the RFID-SLAM method introduced in Section 5. Nevertheless, in the following we will denote poses as global in order to distinguish between the local and global coordinate frame of the robot.

An elevation map is represented by a two-dimensional array storing for each global location (x^g, y^g) a height value h with variance σ_h^2 . In order to determine the height for each location, endpoints from the LRF readings are transformed from robot-relative distance measurements to global locations, with respect to the robot's global pose, and the pitch (tilt) angle of the LRF (see Figure 3). This section is structured as follows. In Section 6.1 we describe the update of single cell values relative to the location of the robot, in Section 6.2 we show the filtering of the map with a convolution kernel and in Section 6.3 we describe an algorithm for the estimation of the robot's 3D pose from dead reckoning and map observations.

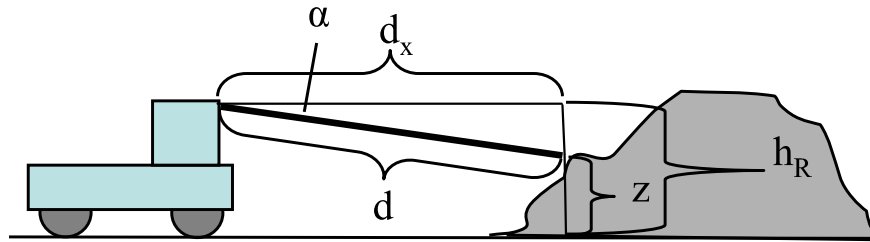


Figure 3. Transforming range measurements to height values.

6.1 Single cell update from sensor readings

Our goal is to determine the height estimate for a single cell of the elevation map with a Kalman filter (Maybeck, 1979), given all height observations of this cell in the past. We model height observations z_t by a Gaussian distribution $N(z_t, \sigma_{z_t}^2)$, as well as the current estimate $N(\hat{h}(t), \sigma_{\hat{h}(t)}^2)$ of each height value. Note that the height of cells cannot be observed directly, and thus has to be computed from the measured distance d and LRF pitch angle α . Measurements from the LRF are mainly noisy due to two error sources. First, the returned distance depends on the reflection property of the material, ranging from very good reflections, e.g. white sheet of paper, to nearly no reflections, e.g. black sheet of paper. Second, in our specific setting, the robot acquires scans while navigating on rough terrain. This will lead to strong vibrations on the LRF, causing an oscillation of the laser around the servo-controlled pitch angle. Consequently, we represent measurements from the LRF by two normal distributions, one for the measured distance $N(\mu_d, \sigma_d)$, and one for the pitch angle $N(\mu_\alpha, \sigma_\alpha)$.

The measurements from the LRF are transformed to robot-relative locations (x^r, y^r) . First, we compute the relative distance d_x and the height z of each measurement according to the following equation (see Figure 3):

$$\begin{pmatrix} d_x \\ z \end{pmatrix} = F_{d\alpha} \begin{pmatrix} d \\ \alpha \end{pmatrix} = \begin{pmatrix} d \cos \alpha \\ h_R - d \sin \alpha \end{pmatrix}, \quad (11)$$

where h_R denotes the height of the LRF mounted on the robot. Second, from distance d_x and the horizontal angle β of the laser beam, the relative cell location (x^r, y^r) of each cell can be calculated by:

$$x^r = d_x \cos \beta \quad (12)$$

$$y^r = d_x \sin \beta \quad (13)$$

Equation 11 can be utilized for computing the normal distributed distance $N(\mu_{d_x}, \sigma_{d_x})$, and height $N(\mu_z, \sigma_z)$, respectively. However, since this transformation is non-linear, $F_{d\alpha}$ has to be linearized by a Taylor expansion at μ_{d_x}, μ_z :

$$\begin{pmatrix} \mu_{d_x} \\ \mu_z \end{pmatrix} = F_{d\alpha} \begin{pmatrix} d \\ \alpha \end{pmatrix} \quad (14)$$

$$\Sigma_{d_x z} = \nabla F_{d\alpha} \Sigma_{d\alpha} \nabla F_{d\alpha}^T \quad (15)$$

$$\text{with } \nabla F_{d\alpha} = \begin{pmatrix} \cos \alpha & -d \sin \alpha \\ -\sin \alpha & -d \cos \alpha \end{pmatrix} \quad (16)$$

$$\text{and } \Sigma_{d\alpha} = \begin{pmatrix} \sigma_d^2 & 0 \\ 0 & \sigma_\alpha^2 \end{pmatrix} \quad (17)$$

Then, the height estimate \hat{h} can be updated from observation z_t , taken at time t , with the following Kalman filter:

$$\hat{h}(t) = \frac{1}{\sigma_{z_t}^2 + \sigma_{\hat{h}(t-1)}^2} \left(\sigma_{z_t}^2 \hat{h}(t-1) + \sigma_{\hat{h}(t-1)}^2 z_t \right) \quad (18)$$

$$\sigma_{\hat{h}(t)}^2 = \frac{1}{\frac{1}{\sigma_{\hat{h}(t-1)}^2} + \frac{1}{\sigma_{z_t}^2}}, \quad (19)$$

Equation 18 cannot be applied if the tilted LRF scans vertical structures since they lead to different height measurements for the same map location. For example, close to a wall the robot measures the upper part, far away from the wall the robot measures the lower part. We restrict the application of the Kalman Filter by the Mahalanobis distance. If the Mahalanobis distance between the estimate and the new observation is below a threshold c , the observation is considered to be within the same height. We use $c = 1$, which has the effect that all observations with a distance to the estimate that is below the standard deviation $\sigma_{\hat{h}}$, are merged. Furthermore, we are mainly interested in the maximum height of a cell, since this is exactly what elevation maps represent. These constraints lead to the following update rules for cell height values:

$$\hat{h}(t) = \begin{cases} z_t & \text{if } z_t > \hat{h}(t) \wedge d_M(z_t, \hat{h}(t)) > c \\ \hat{h}(t-1) & \text{if } z_t < \hat{h}(t) \wedge d_M(z_t, \hat{h}(t)) > c \\ \frac{1}{\sigma_{z_t}^2 + \sigma_{\hat{h}(t-1)}^2} (\sigma_{z_t}^2 \hat{h}(t-1) + \sigma_{\hat{h}(t-1)}^2 z_t) & \text{else,} \end{cases} \quad (20)$$

and variance $\sigma_{\hat{h}(t)}^2$ with:

$$\sigma_{\hat{h}(t)}^2 = \begin{cases} \sigma_{z_t}^2 & \text{if } z_t > \hat{h}(t) \wedge d_M(z_t, \hat{h}(t)) > c \\ \sigma_{\hat{h}(t-1)}^2 & \text{if } z_t < \hat{h}(t) \wedge d_M(z_t, \hat{h}(t)) > c \\ \frac{1}{\frac{1}{\sigma_{\hat{h}(t-1)}^2} + \frac{1}{\sigma_{z_t}^2}} & \text{else,} \end{cases} \quad (21)$$

where d_M denotes the Mahalanobis distance, defined by:

$$d_M(z_t, \hat{h}(t)) = \sqrt{\frac{(z_t - \hat{h}(t))^2}{\sigma_{\hat{h}(t)}^2}}. \quad (22)$$

The cell update introduced so far assumes perfect information on the global pose of the robot. However, since we integrate measurements from the robot while moving in the environment in real-time without loop-closure¹, we have to account for positioning errors from pose tracking that do accumulate over time. Continuous Kalman updates without regarding pose uncertainty due to robot motion in between update steps will successively reduce the cell's variance leading to variance estimates which highly underestimate the actual uncertainty about a cell's height. Thus we increase the variance in the Kalman propagation step based on the robot's motion to reflect the true uncertainty in the variance estimate. The height itself is not changed as the old estimate still gives the best possible estimation for a cell's height. We assume that the positioning error grows linearly with the accumulated distance and angle traveled. Hence, observations taken in the past loose significance with the distance the robot traveled after they were made.

$$\hat{h}(t) = \hat{h}(k) \quad (23)$$

$$\sigma_{\hat{h}(t)}^2 = \sigma_{\hat{h}(k)}^2 + \sigma_d^2 d(t-k) + \sigma_\alpha^2 \alpha(t-k), \quad (24)$$

where t denotes the current time, k denotes the time of the last height measurement at the same location, $d(t-k)$ and $\alpha(t-k)$ denotes the traveled distance and angle within the time

¹Note that global localization errors can be reduced by loop-closure, i.e. by re-computing the elevation map based on the corrected trajectory, which, however, can usually not be applied in real-time.

interval $t - k$, and $\sigma_d^2, \sigma_\alpha^2$ are variances that have to be determined experimentally according to the utilized pose tracker. Since it would be computationally expensive to update the variances of all grid cells each time the robot moves, updates according to Equation 24 are only carried out on variances before they are utilized for a Kalman update with a new observation. The traveled distances can efficiently be generated by maintaining the integral functions $I_d(t)$ and $I_\alpha(t)$ that provided the accumulated distance and angle for each discrete time step t , respectively. Then, for example, $d(k - t)$ can be calculated by $I_d(k) - I_d(t)$. The integrals are represented by a table, indexed by time t with a fixed discretization, e.g. $\Delta t = 1s$.

6.2 Map filtering with a convolution kernel

The limited resolution of the LRF occasionally leads to missing data in the elevation map, e.g. conspicuous by surface holes. Furthermore, the effect of “mixed pixels”, which frequently happens if the laser beam hits edges of objects, whereas the returned distance measure is a mixture of the distance to the object and the distance to the background, might lead to phantom peaks within the elevation map (Ye and Borenstein, 2003). Therefore, the successively integrated elevation map has to be filtered.

In computer vision, filtering with a convolution kernel is implemented by the convolution of an input image with a convolution kernel in the spatial domain, i.e. each pixel in the filtered image is replaced by the weighted sum of the pixels in the filter window. The effect is that noise is suppressed and the edges in the image are blurred at the same time. We apply the same technique on the elevation map in order to reduce the errors described above. Hence, we define a convolution kernel of the size of 3×3 cells, whereas each value is weighted by its certainty and distance to the center of the kernel. Let $h(x + i, y + j)$ denote a height value relative to the kernel center at map location (x, y) , with $i, j \in \{-1, 0, 1\}$. Then, the weight for each value is calculated as follows:

$$w_{i,j} = \begin{cases} \frac{1}{\sigma_{h(x+i,y+j)}^2} & \text{if } |i| + |j| = 0 \\ \frac{1}{2\sigma_{h(x+i,y+j)}^2} & \text{if } |i| + |j| = 1 \\ \frac{1}{4\sigma_{h(x+i,y+j)}^2} & \text{if } |i| + |j| = 2 \end{cases} \quad (25)$$

Consequently, the filtered elevation map h_f can be calculated by:

$$h_f(x, y) = \frac{1}{C} \sum_{i,j} h(x + i, y + j) w_{i,j}, \quad (26)$$

whereas $C = \sum w_{i,j}$.

6.3 3D Pose estimation

So far we have shown an incremental procedure for updating elevation map cells relative to the coordinate frame of the robot. In order to update map cells globally, the full 3D pose of the robot has to be considered, which is described by the vector $l = (x, y, h, \theta, \phi, \psi)^T$, where θ denotes the yaw angle, ϕ denotes the pitch angle, and ψ denotes the roll angle, respectively.

We assume that IMU measurements of the three orientation angles are given with known variance. The position (x, y, h) is estimated by dead reckoning, which is based on the pitch angle and traveled distance measured by visual odometry and scan matching. The scan matching algorithm, which originates from former work (Kleiner et al., 2005), can reliably be applied on robots operating in the plane. However, when traversing three-dimensional terrain, it is very likely that the two-dimensional reference frame changes and thus scan matching leads to inaccurate distance estimates. To obtain correct poses on rough terrain, we employ visual odometry for estimating the distances traversed by the robot. The method detects robot motion based on a voting procedure applied on tracked features generated from camera images and estimates the traveled distance δ from the detected motion and the robot's velocity. A detailed description of the visual odometry is available in (Dornhege and Kleiner, 2006).

Since scan matching and visual odometry are estimating the relative displacement δ with respect to the 3D surface, δ has to be projected onto the plane, as depicted by Figure 4. Given the input $u = (\theta, \phi, \delta)^T$, represented by the Gaussian distribution $N(\mu_u, \sigma_u)$, the

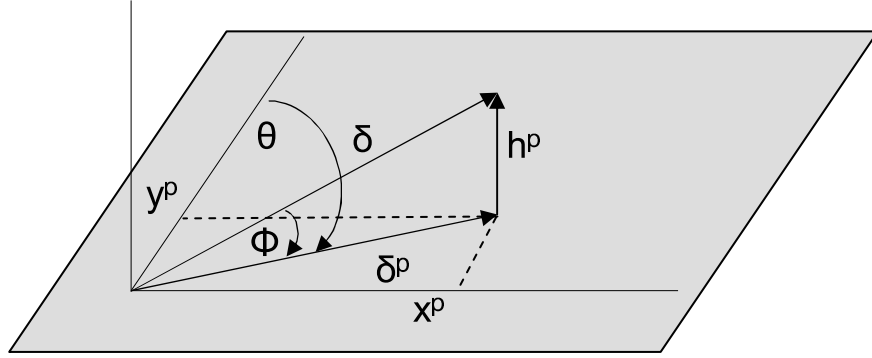


Figure 4. Dead reckoning of the projected Cartesian position (x_p, y_p, h_p) from yaw angle θ , pitch angle ϕ , and traveled distance δ .

projected position $l = (x^p, y^p, h^p)^T$, represented by the Gaussian distribution $N(\mu_l, \Sigma_l)$, can be calculated as follows:

$$\begin{pmatrix} x_t^p \\ y_t^p \\ h_t^p \end{pmatrix} = F_{lu} \begin{pmatrix} x_{t-1}^p \\ y_{t-1}^p \\ h_{t-1}^p \\ \phi \\ \theta \\ \delta \end{pmatrix} = \begin{pmatrix} x_{t-1}^p + \delta \cos \theta \cos \phi \\ y_{t-1}^p + \delta \sin \theta \cos \phi \\ h_{t-1}^p + \delta \sin \phi \end{pmatrix} \quad (27)$$

$$\Sigma_{lu} = \nabla F_{lu} \Sigma_{lu} \nabla F_{lu}^T \quad (28)$$

$$\Sigma_{lu} = \nabla F_l \Sigma_l \nabla F_l^T + \nabla F_u \Sigma_u \nabla F_u^T, \quad (29)$$

where

$$\nabla F_l = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}, \quad (30)$$

$$\nabla F_u = \begin{pmatrix} -\delta \cos \theta \sin \phi & -\delta \sin \theta \cos \phi & \cos \theta \cos \phi \\ -\delta \sin \theta \sin \phi & \delta \cos \theta \cos \phi & \sin \theta \cos \phi \\ \delta \cos \phi & 0 & \sin \phi \end{pmatrix}, \quad (31)$$

$$\Sigma_u = \begin{pmatrix} \sigma_\phi^2 & 0 & 0 \\ 0 & \sigma_\theta^2 & 0 \\ 0 & 0 & \sigma_\delta^2 \end{pmatrix}, \quad (32)$$

$$\Sigma_l = \begin{pmatrix} \sigma_{x^p}^2 & \sigma_{x^p y^p}^2 & \sigma_{x^p h^p}^2 \\ \sigma_{x^p y^p}^2 & \sigma_{y^p}^2 & \sigma_{y^p h^p}^2 \\ \sigma_{x^p h^p}^2 & \sigma_{y^p h^p}^2 & \sigma_{h^p}^2 \end{pmatrix} \quad (33)$$

Equation 27 allows to predict the current height of the robot. However, due to the accumulation of errors, the accuracy of the height estimate will decrease continuously. Therefore, it is necessary to update this estimate from direct observation. For this purpose, we utilize the height estimate $(\hat{h}(t), \sigma_{\hat{h}(t)}^2)$ at the robot's position from Equation 20 and 21, respectively.

Then, the new estimate can be calculated by Kalman-fusing $(\hat{h}(t), \sigma_{\hat{h}(t)}^2)$ with the predicted height estimate $(h^p, \sigma_{h^p}^2)$ analogous to Equation 18.

The global location (x^g, y^g) of a measurement, i.e. the elevation map cell for which the height estimate $\hat{h}(t)$ will be updated, can be calculated straightforward by:

$$x^g = x^r + x^p \quad (34)$$

$$y^g = y^r + y^p \quad (35)$$

7 Experimental Results

In this section we provide results from both simulated and real-robot experiments. All real-robot experiments have been carried out on the robot platforms described in Section 3 within outdoor scenarios, and testing arenas that are equal or similar to those proposed by NIST. In Sections 7.1 results from wheel odometry-based pose tracking are presented. In Section 7.2 we provide results from indoor and outdoor RFID-SLAM experiments, and in Section 7.3 results from elevation mapping during the Rescue Robotics Camp 2006 in Rome, are presented.

7.1 Results from wheel odometry-based pose tracking

The slippage detection method has been extensively evaluated on the *Zerg* robot. During this experiment, the robot performed different maneuvers, such as moving straight, turning, and accelerating while driving first on normal and then on slippery ground. Afterwards, each situation has been manually labeled with one of the six classes *slip-straight*, *slip-turn*,

slip-accelerate, *noslip-straight*, *noslip-turn*, and *noslip-accelerate*. Table 1 summarizes the results of the classification, whereas bold numbers indicate the correct classification, i.e. *true-positives*. As can be seen, the method is able to reliably detect slippage even while the robot is accelerating or performing turns.

Classification		Slip	No Slip
True situation			
Straight	No Slip	10 (0.5%)	2051 (99.5%)
	Slip	2363 (90.1%)	236 (8.9%)
Turn	No Slip	28 (0.9%)	3226 (99.1%)
	Slip	2684 (96.4%)	102 (3.6%)
(De-)Acceleration	No Slip	75 (14.9%)	426 (85.1%)
	Slip	126 (98.5%)	2 (1.5%)

Table 1. Classification error of the slippage detection under different maneuvers of the robot. Bold numbers indicate the correct classifications, i.e. *true-positives*.

In order to evaluate the slippage detection-based odometry improvement, we conducted experiments for the comparison of both improved and conventional odometry and their covariance bounds. Figure 5 shows the performance of slippage sensitive odometry compared to conventional odometry. As can be seen from Figure 5 (a), the error of the conventional odometry increases drastically during slippage (taking place between 10 and 20 meters). Moreover, the covariance bound significantly underestimates the error. However, within the same situation, slippage sensitive odometry is capable of reducing the error Figure 5 (b), while providing valid covariance bounds.

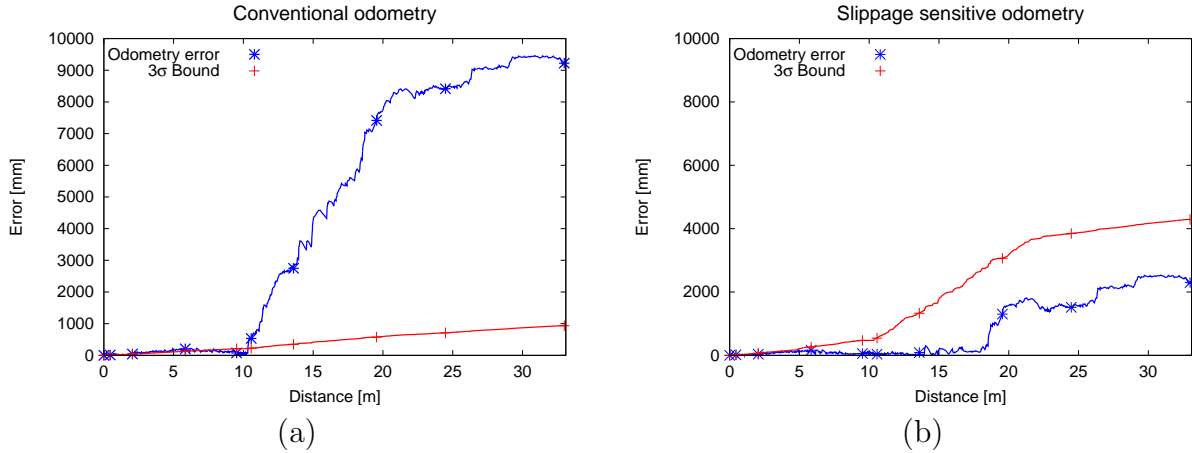


Figure 5. Conventional odometry (a) compared to slippage sensitive odometry (b) during the event of slippage (between 10 and 20 meters): In contrast to conventional odometry, improved odometry reduces the position error (asterisks) and provides valid covariance bounds (crosses) during slippage.

The approach of slippage detection has been utilized during the RoboCup Rescue competition. Figure 6 depicts the *Zerg* robot during the final of the “Best in Class Autonomy” competition, held in the NIST arena for Urban Search and Rescue (USAR) (Jacoff et al., 2001) during RoboCup 2005. In this scenario robots had to explore an unknown area within

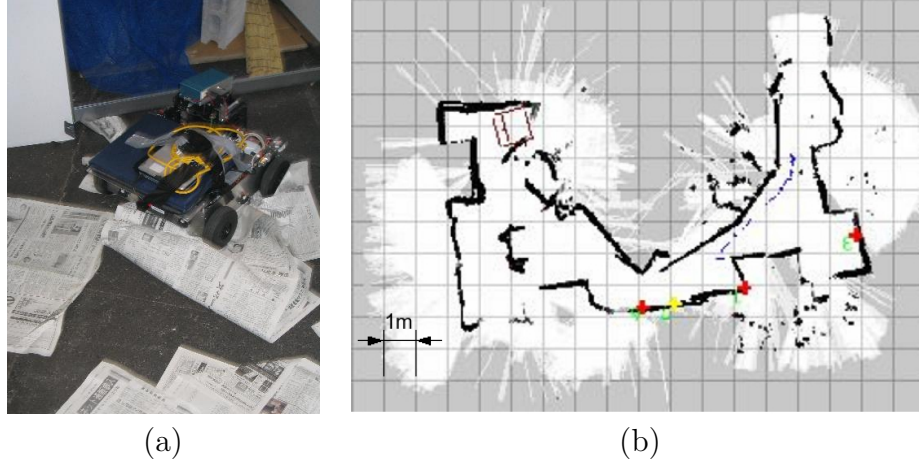


Figure 6. *Zerg* robot during the final of the *Best in Class autonomy* competition at RoboCupRescue 2005 in Osaka: (a) slipping on newspapers and (b) the autonomously generated map. Crosses mark locations of victims which have been found by the robot.

20 minutes autonomously, to detect all victims, and finally to deliver a map sufficient for human teams to locate and rescue the victims. Conditions for exploration and SLAM were intentionally made difficult. For example, the occurrence of wheel slip was likely due to newspapers and cardboards covering the ground, which was partially made of steel and concrete. Stone bricks outside the robot's FOV caused the robot to get stuck, and walls made of glass caused the laser range finder to frequently return far readings. We applied computer vision techniques on images generated by a thermo (IR) camera in order to estimate the relative locations of victims, if they were detected within the camera's FOV. As shown in Figure 6, the system was able to cope with these difficulties and also to build a map reliably, augmented with victim locations detected by the robot. Finally, the system won the autonomy competition in 2005.

7.2 Results from RFID technology-based SLAM

The proposed method for RFID technology-based SLAM has been tested extensively with data generated by a simulator (Kleiner and Buchheim, 2003) as well as on the *Zerg* robot platform. The simulated robot explored three different building maps, a *small* map, *normal* map, and *large* map of the sizes $263m^2$, $589m^2$, $1240m^2$, while automatically distributing RFID tags in the environment. Figures 7 (a-c) show the averaged results from 100 executions of RFID-SLAM on the three maps at five different levels of odometry noise. We measured a computation time of 0.42 seconds on the small map, 2.19 seconds for the normal map, and 13.87 seconds for the large map, with a *Pentium4 2.4GHz*. The small map after and before the correction is shown in Figure 8 (b,d). For this result, the robot distributed approximately 50 RFID tags.

In order to evaluate the performance of RFID-SLAM in a real environment, we collected data from a robot autonomously exploring a cellar for 20 minutes while detecting RFID tags on the ground. The robot continuously tracked its pose as described in Section 4. As depicted by Figure 8 (a,c), the non-linear method successfully corrected the angular error

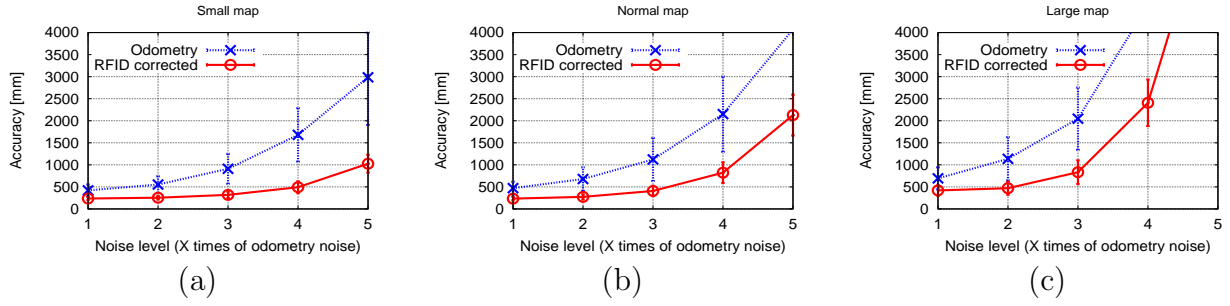


Figure 7. (a) - (c) Result from applying RFID-SLAM at different levels of odometry noise within a simulated office environment: (a) The small map ($263m^2$), (b) the normal map ($589m^2$), and (c) the large map ($1240m^2$).

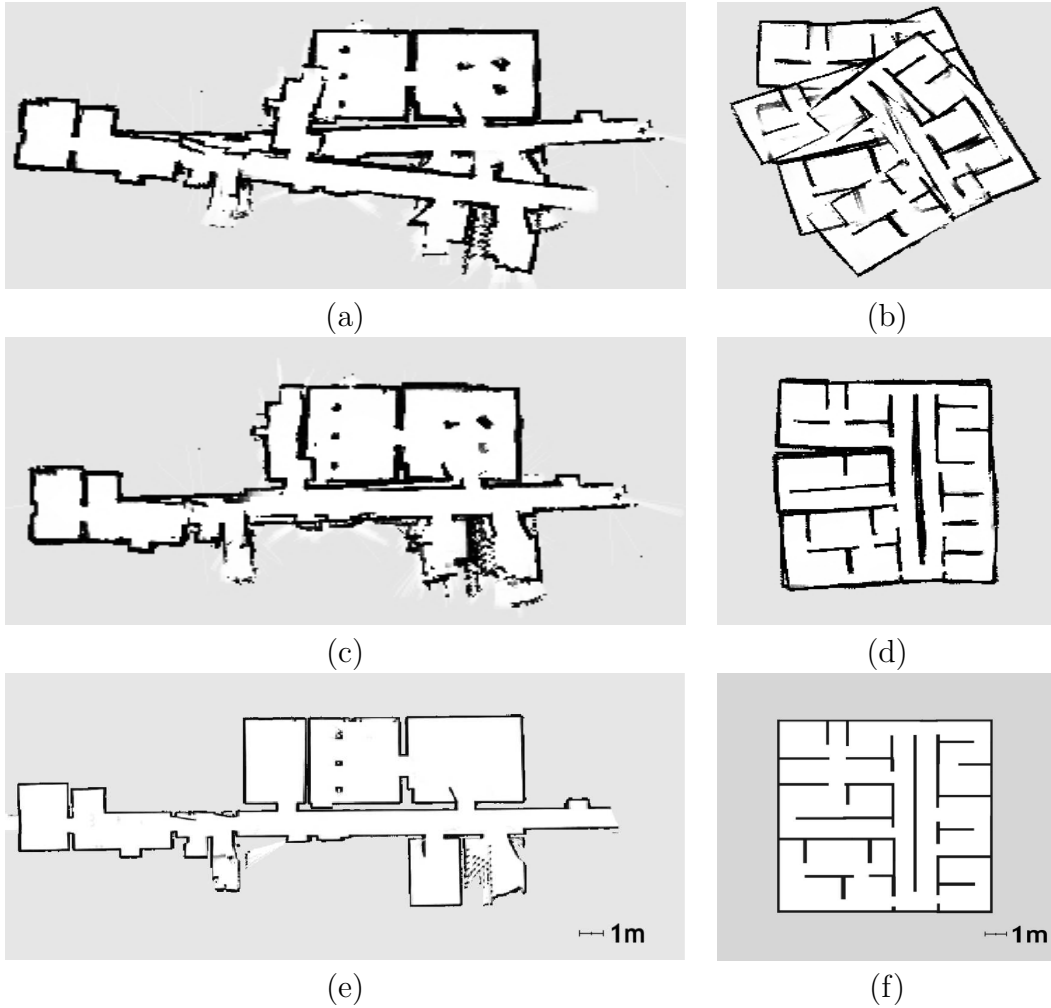


Figure 8. (a,c,e) Result from RFID-SLAM on a robot driving in a cellar: (a) the noisy map, (c) the corrected map, and (e) the ground truth created by iterative scan matching. (b,d,f) Result from applying the RFID-SLAM to data generated in the simulation. (b) The small map with odometry noise, (d) the corrected map, and (f) the ground truth taken directly from the simulator's map editor. Note that the cellar's ground truth map displays unoccupied rooms. The rectangular structures that can be seen in the upper left room in the constructed maps (a) and (c) origin from crates stored in those rooms.

based on RFID data association. The correction was based on approximately 20 RFID tags.

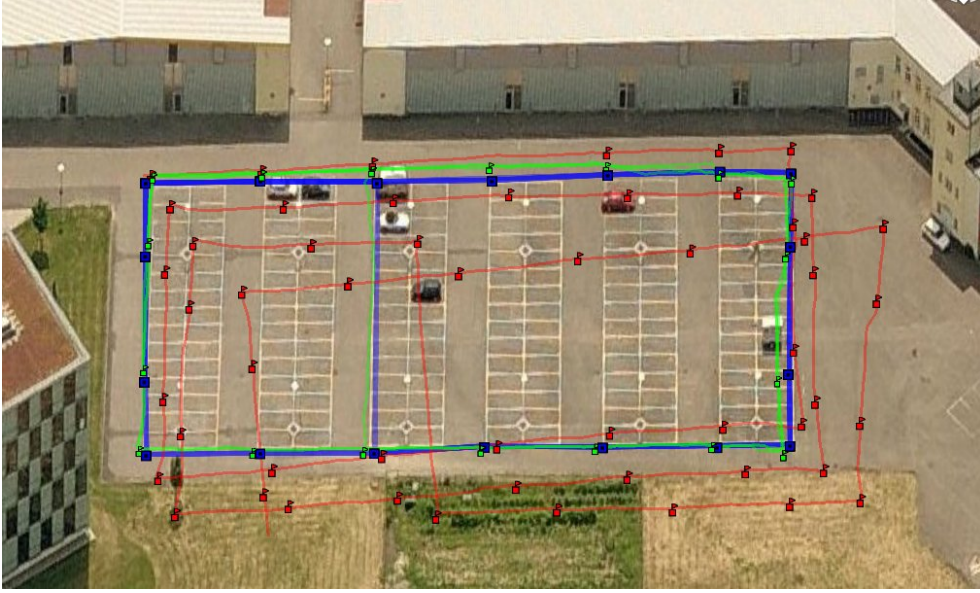


Figure 9. Result from applying RFID-SLAM outdoors while driving with 1m/s on a parking lot. Trajectories are visualized with GoogleEarth, showing RFID locations estimated by the odometry (red), the ground truth (blue), and corrected by RFID-SLAM (green). Whereas the odometry diverges from the driven rectangle, the corrected trajectory is close to the ground truth.

Additionally, we conducted an outdoor experiment with a *Zerg* robot driving with an average speed of 1 m/s on a parking lot. The odometry has been generated from the wheel encoders (translation) and IMU (rotation). Furthermore, the robot detected RFIDs with the antenna described in Section 5. We obtained position ground truth from both Differential GPS (DGPS) and manual measurements, whereas faulty GPS positions, e.g. due to multi-path propagations close to buildings, have been corrected from the manual measurements. Figure 9 shows the RFID locations estimated by the odometry (red), the ground truth (blue), and corrected by RFID-SLAM (green). The corrected trajectory has a mean Cartesian error of $1.8 \pm 3.1\text{m}$, compared to the uncorrected trajectory, which has a mean Cartesian error of $8.3 \pm 8.5\text{m}$.

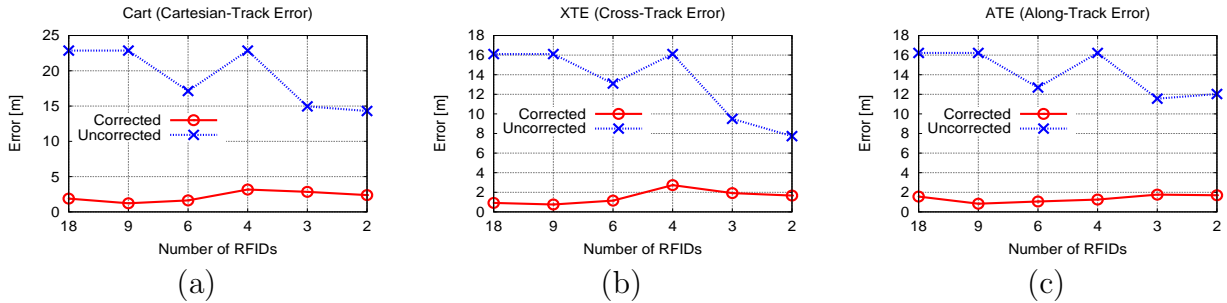


Figure 10. Result from applying RFID-SLAM outdoors while driving with 1m/s for approximately 1km on a parking lot with a *Zerg* robot. (a) the Cartesian error, (b) the cross-track error (XTE), and (c) the along-track error (ATE) with respect to the number of utilized RFIDs.

Furthermore, we evaluated the influence of the number of detected RFIDs with respect to the stability of the SLAM approach. Figures 10 (a), (b), and (c) show the Cartesian error, the cross-track error (XTE), and the along-track error (ATE) at decreasing number of RFIDs, respectively. As can be seen, the route graph optimization consistently improves the accuracy of the trajectory, even with a comparably little number of RFIDs, e.g. one RFID each 500m. The correction of 18 RFIDs took 2.1 seconds on a PentiumM 1.7MHz, and the interpolation of the odometry trajectory took 0.2 seconds. Figure 10 (a) also indicates that the accuracy only slightly improves with increasing number of RFIDs, e.g. the average Cartesian positioning error with 18 RFIDs is 1.2 meters, whereas the error with 2 RFIDs is 2.4 meters. This is due to the fact that during this experiment the rectangular loop has been successfully closed with any number of RFIDs, leading to a near-optimal improvement of the track with respect to the sensor model of the utilized RFID antenna. Note that within arbitrary environments, e.g. non-rectangular shaped, there can be indeed a larger variance of the accuracy given a different amount of RFIDs.

In order to evaluate the scalability of the approach within large-scale environments, we conducted a second outdoor experiment. During this experiment, the robot was driving a total distance of more than 2.5 km with an average speed of 1.58 m/s. Note that this velocity requires human beings to walk comparably fast in order to follow the robot. Furthermore, the robot was heavily shaking from fast navigation over uneven ground, such as road holes, small debris, and grass. Also during this experiment, pose tracking has been performed from data of the wheel encoders (translation) and IMU (rotation), and position ground truth has been obtained from DGPS. The optimization yielded an average Cartesian error of $9.3 \pm 4.9\text{m}$, compared to the uncorrected trajectory, which has an average Cartesian error of $147.1 \pm 18.42\text{m}$. The correction of 10 RFIDs took 0.3 seconds on a PentiumM 1.7MHz, and the interpolation of the odometry trajectory took 2.4 seconds. Figure 11 shows the covariance bounds during EKF-based dead reckoning of the improved odometry (a), and after the global optimization (b), and Figure 12 shows the trajectory of the odometry (red), the ground truth (blue), and the corrected RFID graph (green). As can be seen, the optimization reduces successfully the uncertainties of the poses. Note that for the sake of readability, Figure 11 only shows the first loop of the performed trajectory.

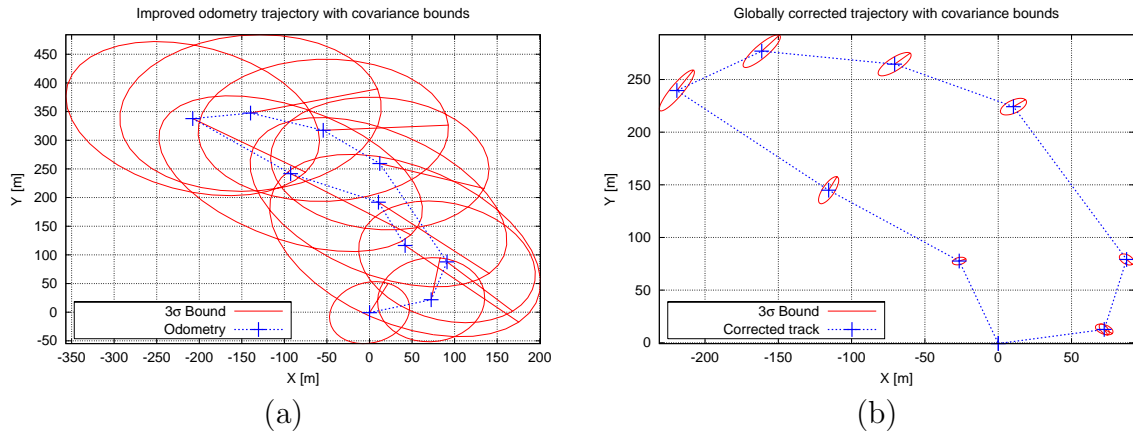


Figure 11. Result from applying RFID-SLAM outdoors while driving with 1.58m/s for more than 2.5km with a Zerg robot. Covariance bounds before (a) and after the correction (b) of the outdoor experiment.



Figure 12. Result from applying RFID-SLAM outdoors while driving with $1.58m/s$ for more than $2.5km$ with a *Zerg* robot. Trajectories are visualized by GoogleEarth, showing the odometry trajectory (red), the ground truth trajectory (blue), and the corrected RFID trajectory (green). Whereas the odometry strongly diverges from the driven path, the corrected trajectory is close to the ground truth.

7.3 Results from elevation mapping

Elevation mapping has been evaluated on a *Lurker* robot, which is capable to overcome autonomously rough terrain containing ramps and rolls. The system has been successfully demonstrated during the RoboCup Rescue autonomy competition in 2006, where the robot won the first prize. The testing arena, which was utilized for the experiments presented in this section, has been installed by NIST during the Rescue Robotics Camp 2006 with the same degree of difficulty as presented at RoboCup'06, i.e. also containing rolls and ramps. During all experiments, the robot was equipped with an IMU sensor, a side camera for visual odometry, and two LRFs, one for scan matching and one for elevation mapping. The latter sensor has been tilted downwards by 35° .

Figure 13 depicts the Kalman filter-based pose estimation of the robot as described in Section 6.3. For this experiment, conditions have been made intentionally harder. Map smoothing has been turned off, which had the effect that missing data, due to a limited resolution of 2D scans, lead to significant holes on the surface of the map. Furthermore, we added a constant error of -2° to pitch angle measurements of the IMU. As shown in Figure 13, the Kalman filter was able to cope with these errors, and finally produced a trajectory close to ground truth (indicated by the gray surface).

In order to quantitatively evaluate elevation mapping performed with visual odometry, we recorded the height estimates of the robot while autonomously exploring an area and finally

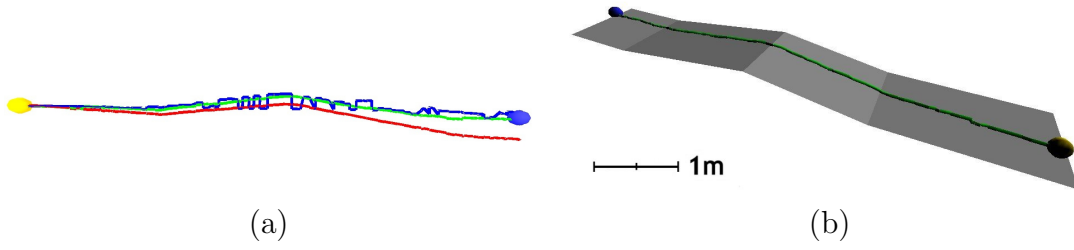


Figure 13. Evaluation of the efficiency of the Kalman filter for estimating the robot's height. (a) Height values predicted from the IMU (lower line in red) are merged with height values taken from the generated map (non-continuous line in blue). Errors from inaccuracies in the map, as well as a simulated continuous drift error of the IMU sensor are successfully reduced (middle line in green). (b) Merged trajectory compared to ground truth (gray ramp).

climbing-up an open ramp. Figure 14 shows the results of the Kalman filter-based height estimation (blue line with crosses) in comparison to the manually measured ground truth (black line with triangles). As shown by Figure 14, the Kalman filter computes continuously an estimate close to ground truth, which stays consistently within the expected covariance bounds.

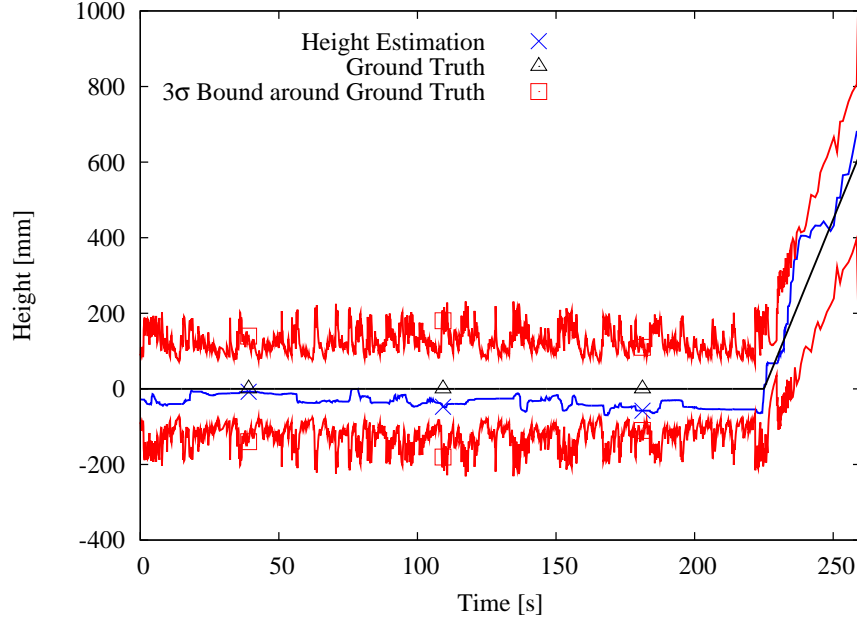


Figure 14. Quantitative evaluation of the Kalman-filtered height estimation. During this experiment the robot started exploration on the floor and then climbed a ramp at around 220s. The graph shows the ground truth (black line with triangles) in comparison to the height estimation by the Kalman filter (blue line with crosses).

Another experiment has been performed for evaluating the influence of visual odometry on elevation mapping. Figure 15 depicts two elevation maps generated on the same ramp, one with support of visual odometry, and the other without. The corresponding error graphs show, that scan matching cannot correctly reflect the robot's motion, when the robot drives on the ramp between 2.75m and 4.75m. Usually estimating the robot's pose based on two dimensional scan alignment can be done reliably, but this is no longer possible when the 2D

reference system changes. The result is a rapid error growth (Figure 15(c)) that leads to distortions in the map due to the incorrect pose assumption. Fusing distance estimates from the visual odometry, that are not influenced by this effect, into the scan matching's pose clearly reduces the error. Mapping based on scan matching only yields a compressed map since in this environment 2D laser scans do not provide sufficient information on the motion of the robot, whereas generating a map based on visual odometry reveals the true size of the ramp, which has been verified by measuring the ramp's dimensions manually.

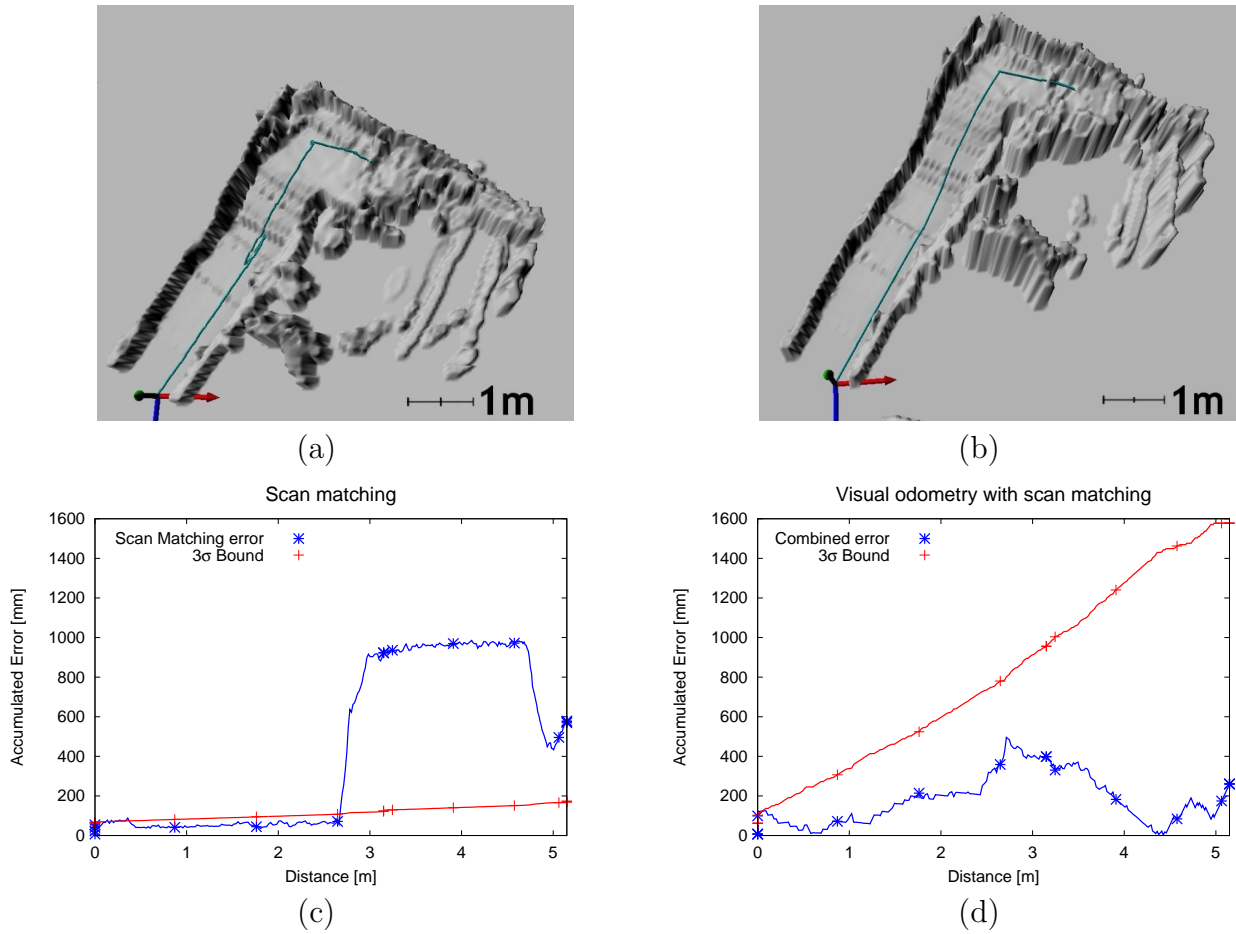


Figure 15. Comparing elevation mapping based on scan matching only (a) and scan matching combined with visual odometry (b). The scan matching's small error (c) grows rapidly out of its usual error bound, when the robot drives on the ramp, while the visual odometry (d) is not influenced by this effect. Scan matching without visual odometry support does not correctly reflect the true length of the ramp, because insufficient motion evidence causes the map to be partially compressed.

Figure 16 and 17 show the final result from applying the proposed elevation mapper during the Rescue Robotics camp. Figure 16 (a) depicts an overview on the arena, and Figure 16 (b) shows the calculated height values, whereas the height of each cell is indicated by a gray value, the darker the cell, the bigger the elevation. Figure 16 (c) depicts the variance of each height cell, going from pink (high variance) to yellow (low variance), whereas the current position of the robot is indicated by a blue circle in the lower left corner. The further away cell updates on the robot's trajectory, the lower their variance (see Section 6.1). Figure 17 shows a 3D visualization of the generated elevation map. Structures, such as the long ramp

at the end of the robot's trajectory, and the stairs, can clearly be identified. We measured on an *AMD64X2 3800+* a total integration time (without map smoothing) of $1.88 \pm 0.47ms$ for a scan measurement with 683 beams, including $0.09 \pm 0.01ms$ for the 3D pose estimation. Map smoothing has generally the time complexity of $O(N^2M)$, where N is the number of rows and columns of the map and M the size of the kernel. We measured on the same architecture $34.79 \pm 14.84ms$ for smoothing a map with $N = 300$ and $M = 3$. However, this can be improved significantly during runtime by only smoothing recently modified map cells and their immediate neighbors within distance M .

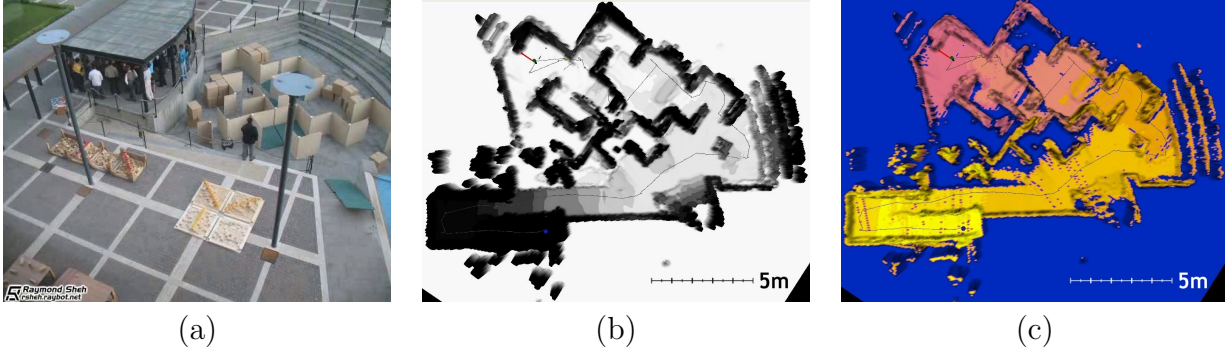


Figure 16. Elevation mapping during the Rescue Robotics Camp 2006 in Rome: (a) The arena build-up by NIST, (b) The corresponding digital elevation model (DEM), build by the *lurker* robot, going from white (low altitude) to black (high altitude). (c) The variances of each height value, going from pink (high variance) to yellow (low variance).

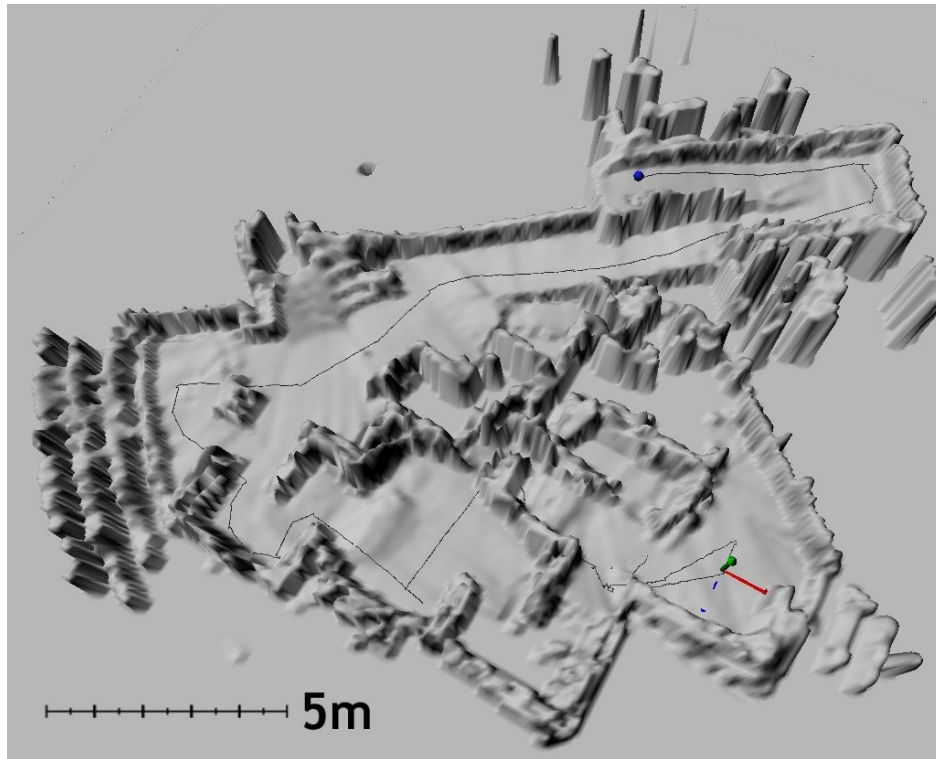


Figure 17. Elevation mapping during the Rescue Robotics Camp 2006 in Rome: 3D perspective.

8 Conclusion

We proposed solutions to the problems of slippage sensitive pose tracking on wheeled platforms, the building of globally consistent maps based on a network of RFID tags, and the building of elevation maps from readings of a tilted LRF. While these methods have been particularly designed for two specific application scenarios, e.g. the rapid mapping of a large-scale environment by wheeled robots, and the mapping of rough terrain by tracked robots, they basically serve as building blocks for tailoring systems according to specific needs.

The quantitative evaluation of indoor and outdoor experiments, partially conducted within testing arenas proposed by NIST for Urban Search and Rescue (USAR), showed that the proposed methods are deployable in real-time, while leading to a robust mapping of the environment. We believe that elevation maps provide the right trade-off between computational complexity and expressiveness. We demonstrated that they can be build reliably in real-time while the robot is in continuous motion, even on rough terrain under consideration of the full 3D pose of the robot. This has partially been achieved by the applying the visual odometry method, which significantly improved the accuracy of scan matching. As we showed within another work, resulting elevation maps can be utilized for structure classification, and the planning of skill execution (Dornhege and Kleiner, 2007).

We showed that RFID-based SLAM allows the efficient generation of globally consistent maps, even if the density of landmarks is comparably low. For example, the method corrected an outdoor large-scale map within a few seconds from odometry data and RFID perceptions only. This has been partially achieved due to reliable pose tracking based on slippage sensitive odometry, but also due to the data association solved by RFIDs. Solving data association by RFIDs allows to speed-up the route graph corrections by decomposing the problem into optimization and interpolation. Besides, RFID-SLAM offers many advantages, particularly within the disaster response scenario. One practical advantage is that humans can be integrated easily into the search, whereas the exchange of maps can be carried out via the memory of RFIDs, hence without need for direct communication (Kleiner and Sun, 2007). Furthermore, they can communicate with RFIDs by a PDA and leave behind information related to the search or to victims. The idea of labeling locations with information that is important to the rescue task, has already been applied in practice. During the disaster relief in New Orleans in 2005, rescue task forces marked buildings with information concerning, for example, hazardous materials or victims inside the buildings (FEMA, 2003). The RFID-based marking of locations is a straight forward extension of this concept.

Within former work, we have already shown the applicability of our method for localizing pedestrians equipped with Personal Dead Reckoning Modules (PDRMs) (Kleiner and Sun, 2007). In future work, we will consider the jointly mapping of places by humans and robots, exchanging map data via the memory of RFIDs. Furthermore, we will evaluate RFID technology operating in the UHF frequency domain, allowing reading and writing within distances of meters, and to extend our approach accordingly. As we have already demonstrated in former work (Ziparo et al., 2007), the combination of a RFID route graph representation with local mapping opens the door to efficient large scale exploration and mapping. In future work, we will deal with the problem of building globally consistent elevation maps by

utilizing RFID technology-based route graph optimization for loop-closure.

Acknowledgments

The authors thank the many enthusiastic students that contributed to the development of the system architecture. Rainer Kümmerle, Daniel Meyer-Delius, Johann Prediger, Michael Ruhnke, and Bastian Steder of the University of Freiburg. Furthermore, we would like to thank the reviewers for their help on significantly improving the quality of the paper.

References

- AlienTechnology (2003). Alien technology. <http://www.alientechnology.com/>.
- Bohn, J. and Mattern, F. (2004). Super-distributed rfid tag infrastructures. In *Proceedings of the 2nd European Symposium on Ambient Intelligence (EUSAI 2004)*, number 3295 in Lecture Notes in Computer Science (LNCS), pages 1–12, Eindhoven, The Netherlands. Springer-Verlag.
- Borenstein, J. and L., F. (1996). Measurement and correction of systematic odometry errors in mobile robots. *IEEE Journal of Robotics and Automation*, 12(6):869–880.
- Dissanayake, M., Newman, P., Clark, S., Durrant-Whyte, H., and Csorba, M. (2001). A solution to the simultaneous localization and map building (slam) problem. *IEEE Transactions on Robotics and Automation*, 17(3):229–241.
- Djugash, J., Singh, S., and Corke, P. (2005). Further results with localization and mapping using range from radio. In *Proc. of the Fifth Int. Conf. on Field and Service Robotics*, Pt. Douglas, Australia.
- Dornhege, C. and Kleiner, A. (2006). Visual odometry for tracked vehicles. In *Proc. of the IEEE Int. Workshop on Safty, Security and Rescue Robotics (SSRR)*, Gaithersburg, Maryland, USA.
- Dornhege, C. and Kleiner, A. (2007). Behavior maps for online planning of obstacle negotiation and climbing on rough terrain. In *Proc. of the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, San Diego, California. to appear.
- FEMA (2003). National urban search and rescue (usr) response system - field operations guide. Technical Report US&R-2-FG, U.S. Department of Homeland Security. <http://www.fema.gov/emergency/usr/resources.shtm>.
- Frese, U. (2006). Treemap: An $o(\log n)$ algorithm for indoor simultaneous localization and mapping. *Autonomous Robots*, 21(2):103–122.
- Gassmann, B., Frommberger, L., Dillmann, R., and Berns, K. (2003). Real-time 3d map building for local navigation of a walking robot in unstructured terrain. In *Proc. of the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, volume 3, pages 2185–2190.
- Grisetti, G., C., S., and W., B. (2005). Improving grid-based slam with rao-blackwellized particle filters by adaptive proposals and selective resampling. In *Proc. of the IEEE International Conference on Robotics and Automation (ICRA)*, pages 667–672, Barcelona, Spain.

- Grisetti, G., Iocchi, L., and Nardi, D. (2002). Global hough localization for mobile robots in polygonal environments. In *Proc. of the IEEE Int. Conf. on Robotics and Automation (ICRA)*, volume 1, pages 353–358.
- Gutmann, J. and Schlegel, C. (1996). Amos: Comparison of scan matching approaches for self-localization in indoor environments. In *Proceedings of the 1st Euromicro Workshop on Advanced Mobile Robots*, pages 61–67. IEEE Computer Society Press.
- Haehnel, D., Burgard, W., Fox, D., and Thrun, S. (2003). A highly efficient fastslam algorithm for generating cyclic maps of large-scale environments from raw laser range measurements. In *Proc. of the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, pages 206–211, Las Vegas, USA.
- Hähnel, D., Burgard, W., Fox, D., Fishkin, K., and Philipose, M. (2004). Mapping and localization with rfid technology. In *In Proc. of the IEEE International Conference on Robotics and Automation (ICRA)*, volume 1, pages 1015–1020.
- Hitachi (2003). *Mu Chip Data Sheet*. http://www.hitachi-eu.com/mu/products/mu_chip_data_sheet.pdf.
- Jacoff, A., Messina, E., and Evans, J. (2001). Experiences in deploying test arenas for autonomous mobile robots. In *Proceedings of the 2001 Performance Metrics for Intelligent Systems (PerMIS) Workshop*.
- Kantor, G. and Singh, S. (2002). Preliminary results in range-only localization and mapping. In *Proc. of the IEEE International Conference on Robotics and Automation (ICRA)*, volume 2, pages 1818–1823, Washington D.C., USA.
- Kantor, G., Singh, S., Peterson, R., Rus, D., Das, A., Kumar, V., Pereira, G., and Spletzer, J. (2003). Distributed search and rescue with robot and sensor team. In *Proceedings of the Fourth International Conference on Field and Service Robotics*, pages 327–332. Sage Publications.
- Kehagias, A., Djugash, J., and Singh, S. (2006). Range-only slam with interpolated range data. Technical Report CMU-RI-TR-06-26, Robotics Institute, Carnegie Mellon University.
- Kleiner, A. and Buchheim, T. (2003). A plugin-based architecture for simulation in the F2000 league. In *RoboCup 2003 : robot soccer world cup VII*, pages 434–445. Springer, Berlin, Padova, Italy.
- Kleiner, A., Prediger, J., and Nebel, B. (2006). Rfid technology-based exploration and slam for search and rescue. In *Proc. of the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, pages 4054–4059.
- Kleiner, A., Steder, B., Dornhege, C., Hoefler, D., Meyer-Delius, D., Prediger, J., Stueckler, J., Glogowski, K., Thurner, M., Lubert, M., Schnell, M., Kuemmerle, R., Burk, T., Braeuer, T., and Nebel, B. (2005). Robocuprescue - robot league team rescuerobots freiburg (germany). In *RoboCup 2005 (CDROM Proceedings), Team Description Paper, Rescue Robot League*, Osaka, Japan.
- Kleiner, A. and Sun, D. (2007). Decentralized slam for pedestrians without direct communication. In *Proc. of the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, San Diego, California. to appear.

- Krotkov, E. and Hoffman, R. (1994). Terrain mapping for a walking planetary rover. *IEEE Transaction on Robotics and Automation*, 10:728–739.
- Kurth, D., Kantor, G., and Singh, S. (2003). Experimental results in range-only localization with radio. In *Proc. of the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, volume 1, pages 974–979, Las Vegas, USA.
- Logitech (2006). *Logitech QuickCam Pro 4000*. <http://www.logitech.com/index.cfm/products/details/US/EN,CRID=2204,CONTENTID=5042>.
- Lu, F. and Milios, E. (1997). Globally consistent range scan alignment for environment mapping. *Autonomous Robots*, 4:333–349.
- Maybeck, P. S. (1979). *Stochastic models, estimation, and control*, volume 141 of *Mathematics in Science and Engineering*.
- Miller, L., Wilson, P. F., Bryner, N. P., Francis, Guerrieri, J. R., Stroup, D. W., and Klein-Berndt, L. (2006). Rfid-assisted indoor localization and communication for first responders. In *Proc. of the Int. Symposium on Advanced Radio Technologies*.
- Montemerlo, M., Thrun, S., Koller, D., and Wegbreit, B. (2002). Fastslam: a factored solution to the simultaneous localization and mapping problem. In *Eighteenth national conference on Artificial intelligence*, pages 593–598, Menlo Park, CA, USA. American Association for Artificial Intelligence.
- Ojeda, L. and Borenstein, J. (2004). Methods for the reduction of odometry errors in over-constrained mobile robots. *Autonomous Robots*, 16:273–286.
- Pfaff, P., Triebel, R., and Burgard, W. (2007). An efficient extension to elevation maps for outdoor terrain mapping and loop closing. *Int. Journal of Robotics Research (IJRR) - Special Issue of Int. Conf. on Field and Service Robotics (FSR)*, 26(2):217–230.
- Quinlan, J. R. (2003). Induction of decision trees. *Machine Learning*, 1(1):81–106.
- Smith, R., Self, M., and Cheeseman, P. (1988). Estimating uncertain spatial relationships in robotics. *Autonomous Robot Vehicles*, pages 167–193.
- Thrun, S., Liu, Y., Koller, D., Ng, A., Ghahramani, Z., and Durrant-Whyte, H. (2004). Simultaneous mapping and localization with sparse extended information filters. *International Journal of Robotics Research*, 23(7–8):693–716.
- Thrun, S., Montemerlo, M., Dahlkamp, H., Stavens, D., Aron, A., Diebel, J., Fong, P., Gale, J., Halpenny, M., Hoffmann, G., Lau, K., Oakley, C., Palatucci, M., Pratt, V., Stang, P., Strohband, S., Dupont, C., Jendrossek, L.-E., Koelen, C., Markey, C., Rummel, C., van Niekirk, J., Jensen, E., Alessandrini, P., Bradski, G., Davies, B., Ettinger, S., Kaehler, A., Nefian, A., and Mahoney, P. (2006). Winning the darpa grand challenge. *JFR*. accepted for publication.
- Wolf, D., Sukhatme, G., Fox, D., and Burgard, W. (2005). Autonomous terrain mapping and classification using hidden markov models. In *Proc. of the IEEE International Conference on Robotics and Automation (ICRA)*, pages 2026–2031.
- Ye, C. and Borenstein, J. (2003). A new terrain mapping method for mobile robots obstacle negotiation. In Gerhart, G. R., Shoemaker, C. M., and Gage, D. W., editors, *Unmanned Ground Vehicle Technology V.*, volume 5083, pages 52–62.

Ziparo, V., Kleiner, A., Nebel, B., and Nardi, D. (2007). Rfid-based exploration for large robot teams. In *Proc. of the IEEE International Conference on Robotics and Automation (ICRA)*, pages 4606–4613.