Mapping for the Support of First Responders in Critical Domains

Alexander Kleiner and Christian Dornhege

Alexander Kleiner and Christian Dornhege, Mapping for the Support of First Responders in Critical Domains, 2011, Journal of Intelligent and Robotic Systems, (64), 1, 7-31.
http://dx.doi.org/10.1007/s10846-010-9520-x

Noname manuscript No.
(will be inserted by the editor)

Mapping for the Support of First Responders in Critical Domains

Alexander Kleiner · Christian Dornhege

Received: date / Accepted: date

Abstract In critical domains such as urban search and rescue (USAR), and bomb disposal, the deployment of teleoperated robots is essential to reduce the risk of first responder personnel. Teleoperation is a difficult task, particularly when controlling robots from an isolated safety zone. In general, the operator has to solve simultaneously the problems of mission planning, target identification, robot navigation, and robot control. We introduce a system to support teleoperated navigation with real-time mapping consisting of a two-step scan matching method that re-considers data associations during the search over transformations. The algorithm processes data from laser range finder and gyroscope only, thereby it is independent from the robot platform. Furthermore, we introduce a user-guided procedure for improving the global consistency of maps generated by the scan matcher. Globally consistent maps are computed by a graph-based maximum likelihood method that is biased by localizing crucial parts of the scan matcher trajectory on a prior given geo-tiff image. The approach has been implemented as an embedded system and extensively tested on robot platforms designed for teleoperation in critical situations, such as bomb disposal. Furthermore, the system was evaluated in a test maze by first responders during the Disaster City event in Texas, 2008.

A. Kleiner
Robotics Institute
Carnegie Mellon University
5000 Forbes Avenue
Newell-Simon Hall, 1502F
Pittsburgh, PA 15213-3890
Tel.: +1 (412) 268-5921
Fax: +1 (412) 268-5569
E-mail: adkleine@andrew.cmu.edu

C. Dornhege
Computer Science Department
University of Freiburg
Georges-Köhler-Allee 52
79110 Freiburg, Germany
Tel.: +49-761-7379
Fax: +49-761-8222
E-mail: dornhege@informatik.uni-freiburg.de

Keywords SLAM · Mapping · HRI · Teleoperation · Operator Assistance

PACS 42.30.Tz · 42.60.By · 07.79.Fc

1 Introduction

In critical domains such as urban search and rescue (USAR), and bomb disposal, the deployment of teleoperated robots is essential to reduce the risk of first responder personnel. Teleoperation is a difficult task, particularly when robots are controlled from an isolated operator station. This is often the case when the target area is hazardous to human beings, requiring the control of the robots from a safety zone. In general, operators have to solve simultaneously the problems of mission planning, target identification, navigation, and robot control. For untrained operators control and target identification are already challenging on their own.

The goal of the proposed system is to support teleoperated navigation of first responders with real-time mapping, and thereby leaving more freedom to operators for performing other tasks. Under certain constraints, such as low visibility and rough terrain, first responder teleoperation leads to very noisy and unusual data when compared to data sets generated by developers. For example, due to environmental make-up and failures in control, laser scans are frequently taken under a varying roll and pitch angle, making it difficult to reliably find correspondences from successive measurements. Many existing mapping algorithms require as an initial guess reasonable pose estimates, e.g., from wheel odometry. However, pose tracking from wheel odometry has proven to be unreliable when there is a large amount of wheel slip, which is likely the case, for example, when navigating tracked robots on rough terrain. Another difficulty arises from the fact that data sets generated during teleoperation are sparsely containing trajectory loops, which are typically required by mapping approaches for compensating the drift of the pose tracker.

In this paper we propose a solution consisting of a two step scan matching method for the real-time mapping of harsh environments, and an offline tool for post-processing the output of the scan matcher. Existing scan matching methods are following the principle of minimizing the squared sum of error distances between successive scans by searching over scan transformations, i.e., rotations and translations. Scan point correspondences are decided based on the Euclidean distance only once before the search starts [Cox, 1991; Lu and Milios, 1997b]. In contrast to these methods the proposed approach re-considers data associations during the search over transformations, which remarkably increases the robustness of scan matching on rough terrain. The algorithm processes data from laser range finder and gyroscope only, making it independent from odometry failures. Note that the latter has furthermore the advantage that the system is easily applicable on different robot platforms since no data connection with the robot is needed.

Furthermore, we are introducing a user-guided procedure for improving the global consistency of maps generated by the scan matcher. Although the presented scan matcher yields a high degree of local consistency enabling the safe guidance of an operator through the environment, resulting maps are not guaranteed to be globally consistent. Global consistency is important when locations of targets have to be communicated between different platforms, or the target location is given as a waypoint in a large-scale scenario. Globally consistent maps are computed by a graph-based maximum likelihood method that is biased by localizing crucial parts of the scan matcher trajectory on a prior given geo-tiff image. The geo-tiff image can either be generated from an aerial image or even from a geo-tagged image that only contains the outer walls of a building. The procedure can also be considered as a tool for generating close-to-ground-truth maps, which can be used, for example, to evaluate mapping results of different teams during a RoboCup Rescue competition, provided that a rough sketch of the global structure exists [Kleiner *et al.*, 2009].

The mapping system has been implemented as a fan-less embedded system which can easily be attached to different robot types, and is currently integrated into a commercial robot platform designed for bomb disposal missions. The system has been extensively tested on robot platforms designed for teleoperation in critical situations, such as bomb disposal. Furthermore, the system was evaluated in a test maze by first responders during the Disaster City event in Texas 2008. Experiments conducted within different environments show that the system yields comparably accurate maps in real-time when compared to more highly sophisticated offline methods, such as Rao-Blackwellized SLAM.

The remainder of this paper is structured as follows. In Section 2 related work is discussed, in Section 3 the two-step scan matcher, and in Section 4 the offline tool for improving global consistency are introduced. Section 5 discusses the implementation of the mapping system. Results from experiments with different robots and environments are shown in Section 6. In Section 7 the conclusion is presented.

2 Related Work

Simultaneous localization and mapping (SLAM) is a frequently studied problem in the robotics literature. In general, mapping techniques for mobile robots can be classified into three processing stages, which are *pose tracking*, such as dead reckoning and scan matching, *incremental loop closure*, i.e., methods that improve the current pose estimate on the fly when loop closures are detected, and *graph-based* techniques, i.e., methods that optimize the entire pose graph according to constraints added between the poses.

Methods for pose tracking can generally be separated into *scan point-based* and *grid-based* methods. Among the scan point-based methods, Cox and colleges proposed a method particularly suited for polygonal environments [Cox, 1991] for matching range readings with a priori given line mode. Lu and Milios [1997b] presented the IDC algorithm that can be applied in non-polygonal environments. Gutmann showed how to combine these two methods in order to improve their overall performance [Gutmann, 2000]. Grid-based methods have the advantage that they are able to filter-out erroneous scans by averaging them by an occupancy grid [Moravec and Elfes, 1985], whereas their disadvantage is a comparably higher demand on memory and computation time. A robust grid-based method has been presented by Hähnel [2005] that aligns scans on a grid map successively build over time. In this paper we introduce a two stage online pose tracking procedure that combines both scan point-based and grid-based processing. In contrast to existing algorithms depending on an initial guess from wheel odometry, our method computes the pose estimate from laser range and IMU measurements only.

In contrast to pose tracking methods, more sophisticated methods incrementally correcting the pose estimate upon detected loop closures have been introduced. Popular approaches are extended Kalman filters (EKFs) [Leonard and Durrant-Whyte, 1991; Smith *et al.*, 1990], sparse extended information filters [Eustice *et al.*, 2005; Thrun *et al.*, 2004], and particle filters [Montemerlo *et al.*, 2003; Grisetti *et al.*, 2007b]. The effectiveness of the EKF approaches comes from the fact that they estimate a fully correlated posterior about landmark maps and robot poses. Their weakness lies in the strong assumptions that have to be made on both, the robot motion model and the sensor noise. If these assumptions are violated the filter is likely to diverge [Julier *et al.*, 1995; Uhlmann, 1995]. Thrun *et al.* [2004] proposed a method to correct the poses of a robot based on the inverse of the covariance matrix. The advantage of sparse extended information filters (SEIFs) is that they make use of the approximative sparsity of the information matrix. Eustice *et al.* [2005] presented a technique that more accurately computes the error-bounds within the SEIF framework and therefore reduces the risk of becoming overly confident. Dellaert and colleagues proposed a method called square root smoothing and mapping (SAM) [Dellaert, 2005; Kaess *et al.*, 2007; Ranganathan *et al.*, 2007]. It has several advantages compared to EKF-based solutions since it better covers the non-linearities and is faster to compute. In contrast to SEIFs, it furthermore provides an exactly sparse factorization of the information matrix.

Methods based on the so called graph formulation of the SLAM problem [Lu and Milios, 1997a; Olson *et al.*, 2006; Grisetti *et al.*, 2007c] are computing maximum likelihood maps by the application of least square error minimization, i.e., addressing the SLAM problem by optimizing the whole network at once. From a sequence of sensor readings a graph of relations is computed, where each node represents a robot pose and an observation taken at this pose. Edges in the graph represent relative transformations between these nodes computed from overlapping observations. Lu and Milios [1997a] first applied this technique in robotics. Olson *et al.* [2006] presented an optimization approach that applies stochastic gradient descent for resolving relations in a network efficiently. Extensions of this work were presented by Grisetti *et al.* [2007c; 2007a]. Most approaches to graph-based SLAM assume that the relations are given. Gutmann and Konolige [1999] proposed an effective way for constructing the network and for detecting loop closures while running an incremental estimation algorithm. In this paper, we also utilize the graph formulation, however, discuss a method for the reliable augmentation of the network with constraints added by the user.

The SLAM methods discussed so far do not take into account any prior knowledge about the environment. The idea of extracting prior information from aerial images has already been discussed. The pose estimate has been improved, for example, by extracting roads from aerial images [Korah and Rasmussen, 2004], and by finding correspondences between images taken from a ground perspective with a monocular vision system and the aerial image [Leung *et al.*, 2008]. Canny edge detection on aerial images for localization has been applied by [Chen and Wang, 2007; Früh and Zakhor, 2004; Kümmerle *et al.*, 2009]. Chen and Wang [2007] use an energy minimization technique to merge prior information from aerial images and mapping. Using a Canny edge detector, they compute a vector field from the image that models force towards the detected edges. The sum of the forces applied to each point corresponds to the energy measure applied in an error minimization process. Früh and Zakhor [2004] described the generation of edge images from aerial photographs for matching 2D laser scans, and Kümmerle *et al.* [2009] extended this approach to 3D laser scans. In contrast to existing approaches the method described in this paper does not require a dense population of edges in the Canny image. The only requirement is that there are edges present that reflect the global structure of the environment, such as the outer walls of a building

3 DCMapping: A Two-Step Scan Matcher

In this section *DCMapping*, a procedure performing two-step processing of laser and gyro data for incrementally tracking the robot's trajectory is described. Given a sequence of scans $\{S_t, S_{t-1}, S_{t-2}, ...\}$, where each scan is defined by a set of Cartesian points, i.e. end points of the scan, $S : \left\{ s_i = (x_i, y_i)^T \mid i = 0...n-1 \right\}$ relative to the robot center, and a sequence of corresponding gyro readings $\{\psi_t, \psi_{t-1}, \psi_{t-2}, ...\}$, the according relative transformations $\tau_i = (\Delta x_i, \Delta y_i, \Delta \theta_i)^T$ of the robot pose within each time interval Δt_i are computed. When driving on rough terrain it is recommendable to deploy a 3-axes IMU for projecting each scan S_i from the sequence to the ground plane. The concatenation of all computed transformations yields an estimate of the robot trajectory which in turn can be used to build a grid map by integrating scan observations at their pose estimate.

In the first step an initial guess for each τ_i is computed by aligning the two most recent scans S_i and S_{i-1} biased by gyro reading ψ_{t-1}. In the second step this initial guess is utilized for finding the final transformation of each scan with respect to the history of scan observations. Whereas the first step is computed efficiently in real time for each scan (ca. 40-80 Hz), the second step is executed at a much slower frame rate (ca. 1 Hz). Local misalignments that can occur during the first step, e.g. due to a strong drift of the gyro bias, can be corrected by exploiting the history of scan observations during the second step.

3.1 Incremental scan alignment

The incremental scan alignment serves as a first step to compute the pose estimate from laser and gyro data. The first step computation has to be particularly robust towards faulty sensor measurements since the resulting pose estimate biases the search over the scan history performed by the second step. One limiting factor on the robustness of scan matching methods is the accuracy of data associations between scan points from successive scans before computing the transformation. Within most conventional approaches data association is decided once by a nearest neighbor approach before computing the transformation between the scans. However, in situations where faulty laser readings are likely, for example when scans are taken at a varying roll and pitch angle, alignment robustness can be increased by re-considering the data association during the search over transformations, which is an essential part of the algorithm presented in this section.

Scans are firstly preprocessed by a scan reduction filter in order to minimize both sensor noise and computation time of the algorithm. This is carried out by clustering scan points that are within a certain Euclidean distance δ from a common cluster center. In our implementation a cluster radius of $\delta = 5cm$ has been selected. Note that in contrast to simply reducing the resolution of the scanner, cluster-based filtering preserves relevant structure of the environment since all measurements are taken into account for computing the cluster centers.

Scan alignment is taking place as shown by Algorithm 1. The procedure takes as inputs the current scan observation and the change of the gyro angle $\Delta\psi$ observed between the current and previous scan (line 1). Note that *gyro angle* refers to the orientation of the robot computed by constantly integrating the turning rate measured by the gyroscope. The current scan is rotated backwards with respect to the angle change, and then further transformed according to the result of a voting-based search procedure.

Algorithm 1: Voting-based scan matching

```
   Input: Scan $S$, gyro change $\Delta\psi$
   Output: Transformation $\tau = (\Delta x, \Delta y, \Delta\theta)^T$
   Data: Reference Scan $R \leftarrow \emptyset$
   // Rotate $S$ by gyro change $\Delta\psi$:
 1 foreach $s_i \in S$ do
 2     $s_i \leftarrow s_i \begin{pmatrix} \cos -\Delta\psi & -\sin -\Delta\psi \\ \sin -\Delta\psi & \cos -\Delta\psi \end{pmatrix}$;
 3 end
   // Generate set of transformations $T$:
 4 $T \leftarrow genTrans\,(time(S) - time(R))$ ;
   // Find best transformation $\tau_{best}$:
 5 $bestVote \leftarrow 0$ ;
 6 $\tau_{best} \leftarrow \emptyset$ ;
 7 foreach $\tau_j \in T$ do
 8     foreach $s_i \in S$ do
 9         $s_i \leftarrow s_i \begin{pmatrix} \cos \Delta\theta & -\sin \Delta\theta \\ \sin \Delta\theta & \cos \Delta\theta \end{pmatrix} + (\Delta x, \Delta y)^T$;
10         foreach $r_i \in R$ do
11             if $distance\,(s_i, r_i) < matchRadius\,(range\,(s_i))$ then
12                 $vote_j \leftarrow vote_j + 1$ ;
13                 break ;
14             end
15         end
16         if $vote_j + remaining\,(S) < bestVote$ then
17             break ;
18         end
19     end
20     if $vote_j > bestVote$ then
21         $bestVote \leftarrow vote_j$ ;
22         $\tau_{best} \leftarrow \tau_j$;
23     end
24     $R \leftarrow S$ ;
25 end
   // Return best transformation found
26 return $\tau_{best}$;
```

The search procedure considers the set of transformations T generated by the function $genTrans(\Delta t)$ (line 4). This function randomly samples a transformation from a set of discretized transformations. The transformation space and its discretization is selected according to the specific motion model of the robot and the frequency of the laser. For example, the higher the frequency of the laser and as lower the the maximal velocity of the robot the smaller the space of transformations. Furthermore, it is possible to adjust the discretization, and thus the accuracy, to computational resources available.

The selection prefers transformations, i.e. selects them with higher probability, if they are close to those that have been selected in the past. This is carried out by maintaining a short-term history of transformations, which is updated online after each estimation step. The idea is motivated from the fact that within a fixed time frame (depending on the data rate of the laser scanner) only a limited amount of motion change can be achieved due to inertia forces. For example, in the case of a continuous fast forward motion of the robot, backward transformations are unlikely to be selected by this function.

The function *matchRadius* (.) returns the Euclidean distance within scan points are considered as being matched. According to the sensor model of the scanner, the returned distance depends on the spot size of the beam, and by this, on the beam's range. For example, longer point distances are returned for further ranges since they are accompanied with a larger beam diameter when hitting an object. *remaining(S)* counts the number of beams that have not voted yet, thus provides an upper bound of the number of votes that still can be achieved.

Although implementing three nested loops, the procedure turns out to be remarkably efficient in practice. We determined experimentally that due to the break conditions on average less than 20% of all beams are considered by the algorithm. Note that the search over possible data associations increases the robustness of the matching significantly.

3.2 Grid-based scan matching

The scan alignment described above provides a good initial guess of the robot transformation. However, more robustness is achieved by matching scans against a history of observations. This is carried out by grid-based scan matching, as described in Hähnel [2005]. The technique determines the estimated robot pose $\hat{x}_t$ relative to the start location from a history of scan observations $S_t, S_{t-1}, ..., S_{t-n}$ and transformations $\tau_t, \tau_{t-1}, ..., \tau_{t-n}$, computed by the first step.

At each sensor observation S_t, a local grid map $\hat{m}\left(\hat{x}_{t-n:t-1}, S_{t-n:t-1}\right)$ is constructed from the previous n scans and pose estimates. The map is then convoluted with a Gaussian kernel in order to accelerate the scan alignment search procedure.

Given current transformation τ_t and scan observation S_t, the new pose estimate x_t is then computed by maximizing the scan alignment on the grid map:

$$\begin{aligned} \hat{x}_t = \underset{x_t}{argmax} \\ \left\{p\left(S_t | x_t, \hat{m}\left(\hat{x}_{t-n:t-1}, S_{t-n:t-1}\right)\right) \cdot p\left(x_t | \tau_t, x_{t-1}\right)\right\}, \end{aligned} \tag{1}$$

where $p\left(x_t | \tau_t, x_{t-1}\right)$ denotes the probability that the robot is located at x_t given the transformation computed by step 1. Finally, new pose x_t and scan S_t are added to the history buffer.

In a last step the generated pose trajectory is utilized for building a map of the environment. This carried out by integrating each scan in an occupancy grid as originally described by Moravec [1988].

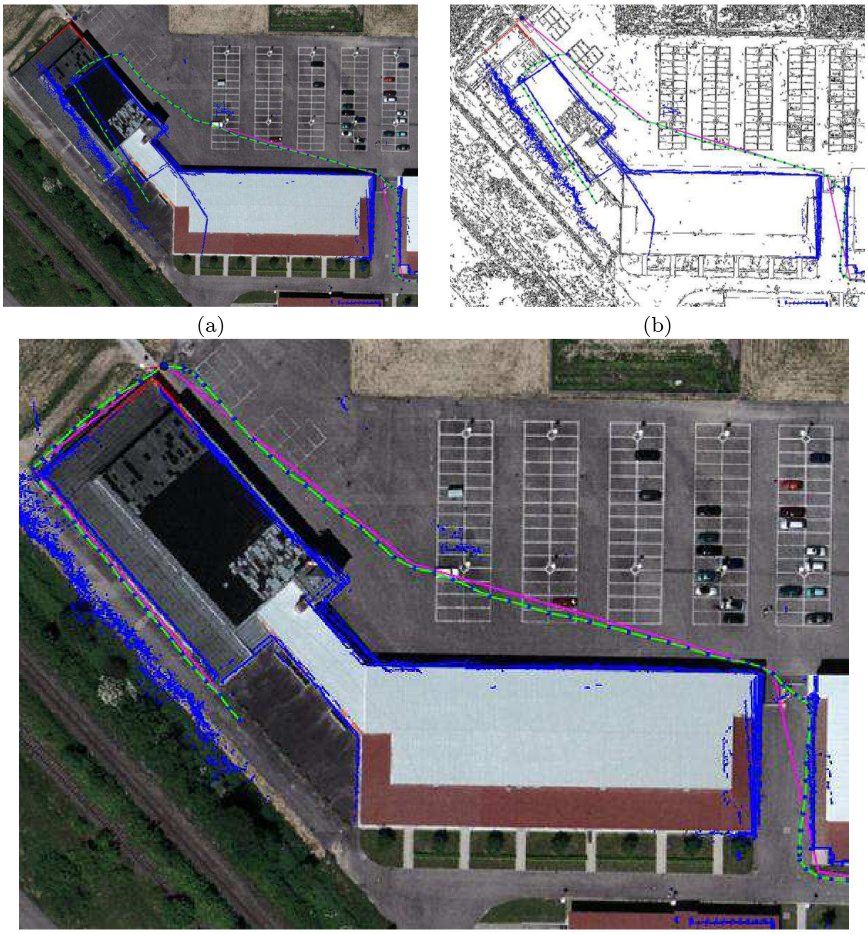
4 Operator Assisted Loop-Closure

In this section we introduce a method for the operator assisted improvement of grid maps that have been generated online by a scan matching method, such as the one described in Section 3. There exist numerous graph-based methods for the offline processing of data sets that improve the trajectory estimate by actively closing loops when they are detected [Olson *et al.*, 2006; Grisetti *et al.*, 2005b]. These approaches typically require the automatic detection of re-visited places from the sensor data, i.e., a robust solution to the data association problem in SLAM. However, data association has to be almost perfectly certain since each association is reflected by a hard constraint in the graph and might cause a strong deformation of the pose network if the underlying association is wrong.

In harsh environments, such as on rough terrain and cluttered outdoor environments, reliable data association from 2D laser scans is a difficult task. Therefore, we propose a user-assisted loop-closure method where the user passively monitors the mapping process from data of a replayed data set until the emerging grid map significantly differs from an overlaid geo-tiff image. By clicking on the image, the user can indicate the approximate location that corresponds to the current readings of the laser range finder. This triggers an automatic process for re-localizing the current pose estimate of the mapping algorithm. More specifically, we apply a variant of monte carlo localization (MCL) [Thrun *et al.*, 2005] on the Canny edge map [Canny, 1986] generated from the geo-tiff image.

The localization procedure can be seen as a tool to fine-positioning the users input, but also as a verification of the users intuition by a scoring function. Constraints that are marginally supported by the data are not added to the graph. By this, data associations are verified twice, on the one hand based on the users intuition and on the other hand based on their support by the data. Figure 1 depicts the dual process of associating the robot trajectory with the geo-tiff image. Note that in this situation the pose tracking process failed to maintain global consistency. This is because at the shown location the environment only contains a single wall on the left, i.e. most of the laser beams are measuring *far ranges*. However, it is comparably easy for a human to identify the approximate correspondence between parts of the local consistent trajectory and the aerial image. As seen by Figure 1 (b) the localizer reliably detects the true correspondence from where on the entire graph is optimized by the graph mapper.

Notice that the presented approach does not require a full representation of the environment by the overlaid map image. For example, the outer walls of a building might be sufficient if they are globally consistent. Here the advantage of the user-guided approach is that relevant structures on the input image can explicitly be selected by the user. Structures on the Canny edge image that are not detected by the laser scanner or structures that do not exist in the real environment can simply be ignored. Given a geo-tiff image of the mapped environment, e.g. an aerial image or a CAD model, and a data set recorded during the pose tracking, the approach can be executed by the following steps:



(a)

(b)

(c)

Fig. 1 This figure demonstrates the operator assisted loop closure process on a data set where the sensorhead was carried by a human. (a) shows a situation, where the current trajectory significantly deviates from the overlaid ground truth. The scan marked in red is the current scan that is localized on the Canny map (b) to the upper right corner of the building. Green edges represent soft constraints in the pose graph and purple edges represent hard constraints. Using the localized scan, a hard constraint is added to the ground truth position. This iterative process of localizing scans on the Canny map and optimizing the pose graph leads to a globally consistent map (c).

1. Computation of the Canny edge image from the geo-tiff image.
2. Construction of the pose graph by adding each pose estimate as a soft constraint.
3. Visualization of the trajectory on the geo-tiff image. Each time the estimate significantly differs from global structures on the map, the current sensor reading is re-localized on the Canny edge map and a hard constraint is added to the graph.
4. Optimization of the constraint graph.

4.1 Map Localization

To localize sensor readings on the Canny edge map a method similar to the one presented in our our previous work [Kümmerle *et al.*, 2009] is used. This approach implies basically the computation of the likelihood $p(\mathbf{z} \mid \mathbf{x}, m)$ of a 2D range scan $\mathbf{z}$ on a map m given the robot is located at pose $\mathbf{x}$. This likelihood is computed by applying the so called endpoint model or likelihood fields approach [Thrun *et al.*, 2005]. Let z_j be the j-th measurement of a scan $\mathbf{z}$ with k measurements. The endpoint model computes the likelihood of z_j based on the distance between the scan point z'_j corresponding to z_j re-projected onto the map m according to the pose estimate $\hat{\mathbf{x}}$ of the robot and the point in the map d'_j which is closest to z'_j as:

$$p(\mathbf{z} \mid \hat{\mathbf{x}}, m) = f(\|z'_1 - d'_1\|, \ldots, \|z'_k - d'_k\|). \tag{2}$$

Under the assumption that the beams are independent and the sensor noise is normally distributed we can rewrite Equation (2) as

$$p(\mathbf{z} \mid \hat{\mathbf{x}}, m) \propto \prod_j e^{-\frac{(z'_j - d'_j)^2}{\sigma^2}}. \tag{3}$$

4.2 Hypothesis Sampling

In contrast to the method presented in [Kümmerle *et al.*, 2009] we do not assume the existence of an accurate motion model of the robot for propagating pose hypotheses on the map. This is because the localization procedure is typically triggered by the user when there is a large deviation of the estimate from the true location on the map. Estimation errors, on the other hand, are most likely due to an overconfident motion model, e.g., they might occur in situations with heavy wheel-slippage or when there is an insufficient amount of features for the laser-based tracking.

Hence, our method searches for the current pose on the map based on the sensor observations only. This is carried out by sampling n poses on the map. The probability density $N(\mathbf{u}, \mathbf{\Sigma_u})$, with $\mathbf{u} = (x_u, y_u, \theta_u)^T$ and 3×3 covariance matrix

$$\mathbf{\Sigma_u} = \begin{pmatrix} \sigma_x^2 & 0 & 0 \\ 0 & \sigma_y^2 & 0 \\ 0 & 0 & \sigma_\theta^2 \end{pmatrix} \tag{4}$$

is provided by the user as an initial guess, e.g., by drawing an arrow according to the approximately true pose on the map. We used $\sigma_x = \sigma_y = 4$ m and $\sigma_\theta = 15°$. The parameters were chosen according to our experience with different users performing manual data association on the map within a reasonably short amount of time. For each sampled pose $\mathbf{s_i} = (x_i, y_i, \theta_i)^T$ we estimate its probability as given in Equation 5. We assume that $\mathbf{z}$ is independent from $\mathbf{u}$ and $\mathbf{s_i}$ is independent from m.

$$p(\mathbf{s_i} \mid \mathbf{z}, \mathbf{u}, m) \propto p(\mathbf{z} \mid \mathbf{s_i}, m) p(\mathbf{s_i} \mid \mathbf{u}) \tag{5}$$

$$p(\mathbf{s_i} \mid \mathbf{z}, \mathbf{u}, m) \propto e^{-\frac{1}{2}(s_i - u)\Sigma_u^{-1}(s_i - u)} \prod_j e^{-\frac{(z'_j - d'_j)^2}{\sigma^2}}. \tag{6}$$

We aim to find the sample $\mathbf{s_i^*}$ that maximizes the likelihood of the scan with respect to the user input. Using the monotonicity of the logarithm we can maximize the log-likelihood as shown in Equation 8.

$$\mathbf{s_i^*} = \underset{s_i}{\operatorname{argmax}}\, p(\mathbf{s_i} \mid \mathbf{z}, \mathbf{u}, m) \tag{7}$$

$$= \underset{s_i}{\operatorname{argmax}} -\frac{1}{2}(\mathbf{s_i} - \mathbf{u})\Sigma_u^{-1}(\mathbf{s_i} - \mathbf{u}) + \sum_j -\frac{(z'_j - d'_j)^2}{\sigma^2} \tag{8}$$

To improve convergence of the sampling procedure after a sample s_i is drawn we perform hill-climbing search around s_i. Therefore we vary x_i, y_i and θ_i individually and gain variations s'_i of the original sample s_i. We compute the weight w'_i for each variation and if any w'_i is greater than w_i we replace s_i by s'_i. The procedure terminates if no variation leads to a higher weight. The hypothesis sampling is outlined by Algorithm 2.

Algorithm 2: Hypothesis sampling with Hill Climbing

```
Input: Scan $z$, Map $m$, User input $u$, $\Sigma_u$
Output: Scan Pose $s_i^*$
$s_i^*$ = None
bestSample ← $-\infty$
// converged will be True, if $N = 10000$ samples have been drawn,
// without finding a better $s_i^*$.
while ¬ converged do
    $s_i = sample(u)$
    improved ← True
    // Perform hill climbing until $s_i$ cannot be improved any more.
    while improved do
        // computeScore calculates the log-likelihood according to Eqn. 8.
        bestNeighbor = computeScore($s_i, z, m, u$)
        // Compute $s_i$'s neighbors by varying the individual components:
        $S \leftarrow neighbors(s_i)$
        improved ← False
        foreach $s'_i \in S$ do
            score = computeScore($s'_i, z, m, u$)
            if score > bestNeighbor then
                bestNeighbor ← score
                $s_i \leftarrow s'_i$
                improved ← True
            end
        end
    end
    if bestNeighbor > bestSample then
        bestSample ← bestNeighbor
        $s_i^* \leftarrow s_i$
    end
end
return $s_i^*$
```

The displacement between the current pose estimate and the pose determined by the location procedure is added as a constraint to the pose graph. The pose graph is then immediately optimized by a graph-based maximum likelihood method as described in [Grisetti *et al.*, 2009]. First, the graph is constructed from soft constraints, where each node i in the graph is created from a robot pose in the data set. An edge between two nodes i and j is represented by the tuple $\langle \delta_{ji}, \Omega_{ji} \rangle$, where δ_{ji} and Ω_{ji} are the mean and the information matrix of the measured relative displacement, respectively. Let $e_{ji}(\mathbf{x}_i, \mathbf{x}_j)$ be the error introduced by the constraint $\langle j, i \rangle$. Assuming the independence of the constraints, the graph can be optimized by

$$(\mathbf{x}_1, \ldots, \mathbf{x}_n)^* = \underset{(\mathbf{x}_1, \ldots, \mathbf{x}_n)}{\operatorname{argmin}} \sum_{\langle j,i \rangle} e_{ji}(\mathbf{x}_i, \mathbf{x}_j)^T \Omega_{ji} e_{ji}(\mathbf{x}_i, \mathbf{x}_j). \tag{9}$$

In order to incorporate hard constraints from the user-based data association we extend Eqn. (9) as follows [Kümmerle *et al.*, 2009]:

$$\begin{aligned}(\mathbf{x}_1, \ldots, \mathbf{x}_n)^* = \underset{(\mathbf{x}_1, \ldots, \mathbf{x}_n)}{\operatorname{argmin}} & \sum_{\langle j,i \rangle} e_{ji}(\mathbf{x}_i, \mathbf{x}_j)^T \Omega_{ji} e_{ji}(\mathbf{x}_i, \mathbf{x}_j) \\ & + \sum_{i \in \mathcal{G}} (\mathbf{s}_i - \hat{\mathbf{x}}_i)^T R_i (\mathbf{s}_i - \hat{\mathbf{x}}_i), \end{aligned} \tag{10}$$

where $\hat{\mathbf{x}}_i$ denotes the pose estimated by the scan matcher and $\mathbf{s}_i$ the corresponding position found by the localization procedure applied on the image, and R_i is the information matrix of this constraint computed from a highly confident covariance matrix. During our experiments we assigned double weights to the hard constraints compared to the soft constraints. Eqn. (10) is optimized by non-linear conjugate gradient resulting in a set of poses that maximize the likelihood of all observations added to the graph. In our implementation we used the freely available TORO library [Grisetti *et al.*, 2009].

5 Implementation

The mapping system has been implemented by two versions. First, a simple setup that can be used by humans to generate maps by carrying a notebook and sensors through the environment. Second, as an embedded system that can be mounted on commercial robot platforms.

5.1 Simple Setup

Due to the fact that the proposed mapping system operates independently from encoder data, such as wheel odometry, environments can be mapped by simply carrying laser and gyroscope through the environment. Figure 2 (a) depicts the setup and Figure 2 (b) shows a map generated with it.

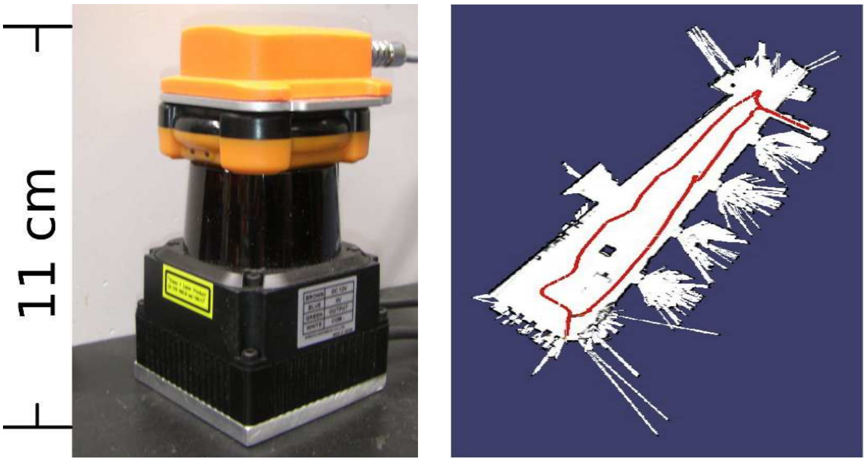


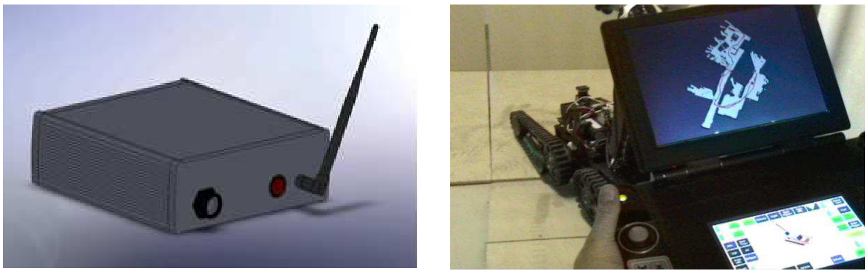
Fig. 2 Simple setup for mapping by a human walking through the environment. (a) The two utilized sensors. (b) Example of a generated map.

5.2 Embedded System

The goal of the proposed mapping system is to enable online mapping on commercial robot platforms during first responder teleoperation. To achieve this goal, the mapping system has to be embeddable on these platforms, wherefore three requirements have to be fulfilled: First, the device has to be small enough to fit onto the platform. Second, the device has to be waterproof, i.e. based on a fan-less CPU. Third, the device has to communicate via the existing communication link of the robot, which is typically an analog video signal linked via a radio transmitter or tethering.

The mapping system has therefore been implemented as an embedded system (see Figure 3 (a)). The black box, which we named *SensorHead*, processes sensor readings from an internal IMU and an externally attached laser range finder (LRF). The unit is directly powered from the power supply of the robot (either 24 V or 48 V). Note that no other connections, e.g. from wheel encoders, are required. The SensorHead contains a 3.5" *Wafer-Atom* board equipped with a 1.6 GHz Intel Atom CPU, and 2 GB memory, a 64 GB solid state disk, a *Xsens MTi* Inertial Measurement Unit (IMU), and a video converter. Online computed maps can additionally be communicated via Wireless-LAN.

Robots designed for teleoperation typically forward video signals from cameras either via a radio link or tethering. Thus, the main output of the box is a video signal that can be directly fed into the on-board video system of the robot and then be transmitted to the operator console. Figure 3 (b) depicts the integration of the SensorHead on a Telemax robot with operator console displaying the generated map and current position. The SensorHead module is currently integrated by a company into a commercial robot platform designed for bomb disposal missions.



(a) (b)

Fig. 3 Implementation of the mapping system as an embedded system: The black-box "SensorHead". (b) The box integrated on the Telemax robot

6 Experiments

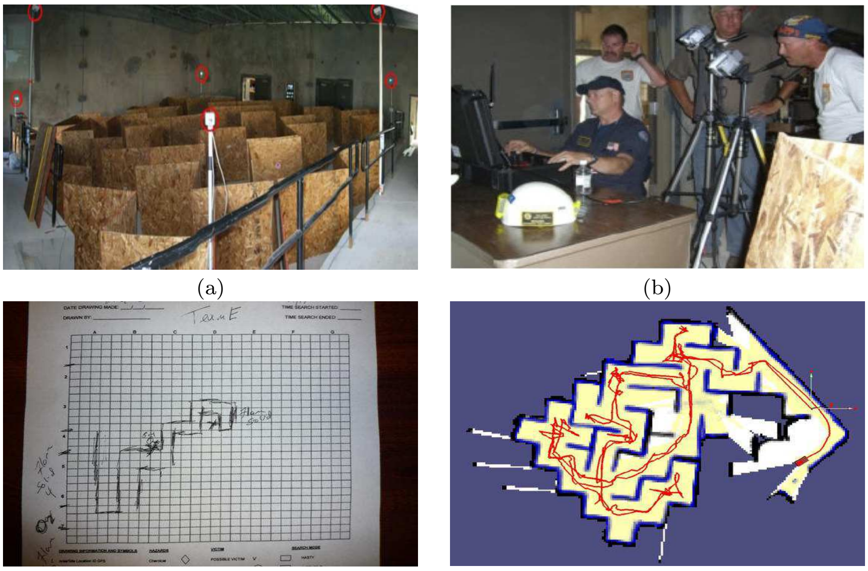
The mapping system was evaluated on several robot platforms, e.g., *Telemax* (Telerob GmbH), *Talon Responder* (Foster-Miller), *Matilda* (Mesa Robotics), *Pioneer AT* (ActiveRobots), a Kyosho Twin Force R/C car, and also by humans carrying the setup. In this section results from experiments conducted on the robot platforms will be presented.

6.1 First Responder Evaluation at DisasterCity

The first responder evaluation during DisasterCity 2008 in Texas was organized by the National Institute of Standards and Technology (NIST). Within time slots of 15 minutes multiple teams consisting of two first responders had to localize and report locations of hazmat symbols that were deployed in a maze-like structure before each run (Figure 4 (a)). On the one hand there were teams exploring the maze by manual mapping (Figure 4 (c)), and on the other hand, teams that utilized the output of our mapping system at the operator console, as shown in Figure 4 (d).

Navigating robots through the maze was a challenging task due to an overall inclination of 15° and additional rolls and ramps (either 10° or 15° inclined) that covered the maze entirely. Furthermore, the maze was obscured. Thus, hazmat symbols and the structure of the maze had to be recognized by the responders via the robot's on-board cameras and lights illuminating the scene. Due to these extraordinarily harsh conditions some responders even failed to simultaneously detect targets and to control the robot. Consequently, some responders had major difficulties to localize within the maze when only supported by the video stream transmitted from the robot's on-board cameras. Hence, there were frequently situations in which regions were explored twice or entirely missed. In contrast, other responders demonstrated efficient maze navigation when supported by the mapping system. They were able to explore the maze systematically and furthermore to exit the maze at any time by following the shortest path from their current location.

In summary, our system was able to repeatedly map the maze (see Figure 4 (d)) even under extremely harsh conditions: Inexperienced responders drove the



(a) (b) (c) (d)

Fig. 4 Test setting and results from the first responder evaluation during DisasterCity 2008: (a) Test maze inclined by 15° and additionally covered with rolls and ramps (obscured during experiments), (b) Responders teleoperating the robot. (c) Map generated manually by responders. (d) Map generated with the presented mapping system by responders.

robot at maximal velocity over rolls and ramps causing jumps and heavy collisions. Although robust robot platforms were used for the evaluation (e.g. Talon from Foster-Miller and Matilda from Mesa Robotics), experiments had to be restarted at least five times because the teleoperated robot had been turned over or major malfunctions occurred.

6.2 Scan Matching Evaluation

In this section results are presented from a quantitative evaluation of our approach (*DCMapping*) compared to rao-blackwellized particle filtering (RBPF) that detects loop-closures for improving the map [Grisetti *et al.*, 2005a]. More specifically, we run the *GMapping* implementation which is freely available on the Internet [Stachniss *et al.*, 2007]. *GMapping* requires data from wheel odometry in order to compute the map. Since our data sets do not contain wheel odometry, *GMapping* was run with the output of the first level of the scan processing described in Section 3. We will first describe our evaluation methodology and then provide results from applying the methods on different maps.

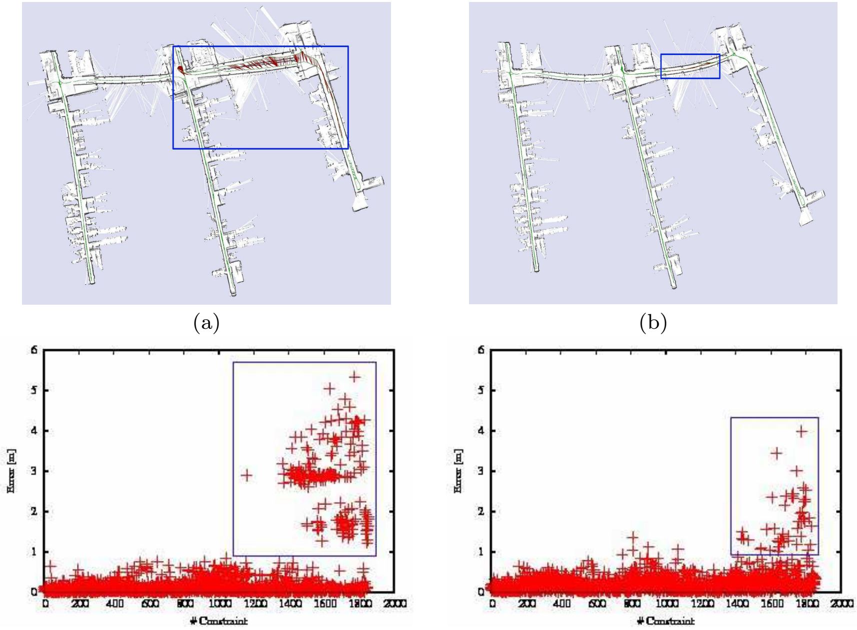
6.2.1 Performance Metric

We based our evaluation on the estimated robot trajectory $x_{1:T}$, where x_i is the robot pose at timestep i from 1 to T. Let $x^*_{1:T}$ be the reference poses of the robot, ideally the true locations. We use a measure based on the *relative* displacement

$\delta_{i,j}$ between poses to perform comparisons. We define $\delta_{i,j} = x_i \ominus x_j$, where $\oplus$ is the standard motion composition operator and $\ominus$ its inverse. Instead of comparing x to x^* (in the global reference frame), we do the operation based on δ and δ^* as

$$\varepsilon(\delta) = \sum_{i,j} (\delta_{i,j} \ominus \delta^*_{i,j})^2. \tag{11}$$

As shown by Equation 11 the computation of the metric requires the true displacements δ^* between the poses of the robot. We use a two step approach to derive those in an assisted manner: First, a SLAM algorithm is executed under manual supervision on the raw data to estimate a trajectory that is globally consistent. From this trajectory initial displacement candidates $\delta^i_{i,j}$ are derived. Next, every candidate $\delta^i_{i,j} = x_i \ominus^i x_j$ is manually verified. In this step a human expert is presented pairs of laserscans at timestep i and j displaced by $\ominus^i$. The expert can *accept* or *reject* the displacement for the final displacements $\delta^*_{i,j}$ and if accepted it is possible to manually adjust $\ominus^i$ to reflect $\ominus^*$. Although work intensive, we believe, that without manual intervention or other external sources it is impossible to generate reliable results. A more detailed description can be found in our previous work [Burgard *et al.*, 2009].



(a)

(b)

(c)

(d)

Fig. 5 Comparison of results on the Imtek dataset. The left side shows the map generated by scan matching (a) and the translational error plotted by relation (c). On the right side the map by RBPF (b) is shown with the corresponding error plot (d). The maps are overlaid with the graph formed by the given relations. Both error plots show a remarkable cluster towards the end. Rectangles mark the corresponding relations in the plot and the map. All marked relations are also drawn in red in the map.

One advantage of this metric is that it allows to compare algorithms in individual situations. An example of a difficult environment with long hallways and glass walls is shown in Figure 5. Figures 5 (c, d) depict the translational error plotted for each relation. Large clusters are visible for both the RBPF and DCMapping. We identified the relations in those clusters and marked the corresponding relations in the resulting maps. The clusters clearly identify the weak point in the map that originates from the robot returning after driving down a long hallway without features. DCMapping is here unable to close the loop fully resulting in the shown shearing effect. The RBPF can close the loop. Therefore, only less relations with a lower magnitude are found in its cluster, which correspond to the slight bend of the corridor.

6.2.2 DCMapping Evaluation

Experiments were carried out on two types of data sets and were executed on an *Intel Core2Duo 2.6GHz*. On the one hand we evaluated data sets that were recorded during teleoperated exploration of building structures. These are the *082er* data set, recorded in a cellar, the *tu-darmstadt* data set, recorded in a long hallway of the Darmstadt university, the *tu-graz* data set, recorded in a cluttered office environment of the Graz university, the *imtek* data set, recorded in three interconnected buildings of the Freiburg university, the *telemax-hard* data set, recorded in a cellar while driving over ramps and rolls, the *hangar* data set, recorded in a hangar like structure, and the *dc-maze* data set, recorded at DisasterCity in Texas. On the other hand we evaluated data sets that are typically used by the SLAM community and available online. These are *aces*, *intel-lab*, and *mit-killian.* Note that the latter data sets contain a comparably high number of situations where the robot re-entered previously explored regions, i.e., enabling loop-closures by the algorithm. Figure 6 depicts the result of DCMapping for some of the maps.

The evaluation presented in Table 1 and Table 2 shows that DCMapping and RBPF yield on average comparably equal good results. This is surprising since RBPF needs much more time to compute the corrected map than DCMapping, which provides results in real-time. For example, we measured a run time of 330 minutes for RBPF on the embedded system described in Section 5 when processing the 82er data set. In contrast, DCMapping took 6 minutes (real-time), which is about 55 times faster. Note, that particularly when closing larger loops, the computation time of RBPF increases drastically. Differences can be found, on the one hand, on maps containing larger loops, such as *aces*, *intel-lab*, and *imtek*, where RBPF shows its strength in closing loops, and on the other hand, *telemax-hard* and *maze* where DCMapping clearly shows robustness against extremely faulty laser range readings from rolls and ramps.

6.3 Assisted Loop-Closure Evaluation

The assisted loop-closure approach has been evaluated on three large scale data sets which will be explained in more detail in the following. The process of user-guided map alignement took in the worst case (the largest data set) no more than 10 minutes. The first data set shown in Figure 7 has been recorded on a Matilda robot from Mesa robotics equipped with a Hokuyo UTM 30 laser range finder and

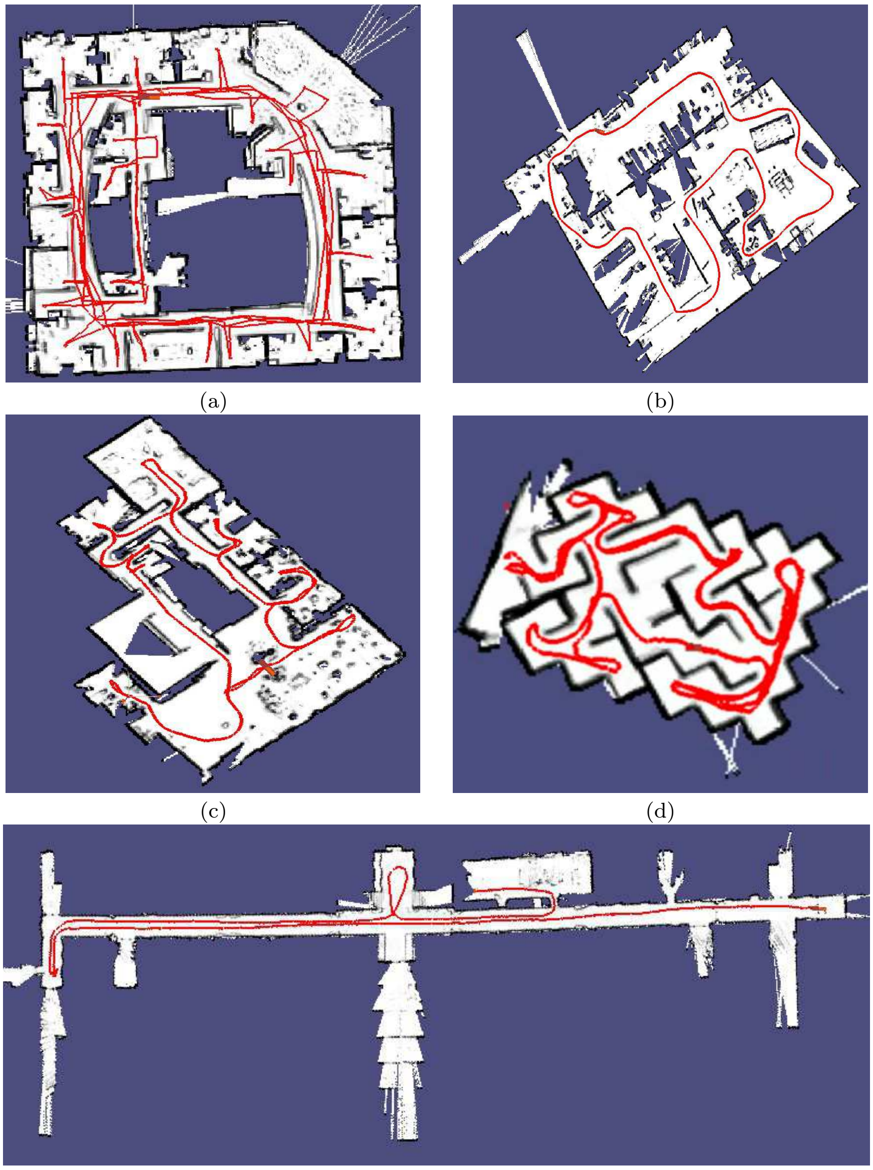

(a)
(b)
(c)
(d)
(e)

Fig. 6 Maps generated with DCMapping: (a) *aces*, (b) *hangar*, (c) *tu-graz*, (d) *maze*, and (e) *tu-darmstadt*. The red line indicates the path taken by the robot.

the MTi IMU from Xsens. The robot has been steered through the hangar-like testing facility from the Southwest Research Institute (SWRI) in San Antonio, Texas. The input to the assisted loop-closure has been computed by DCMapping, the result is shown by Figure 7 (c). As ground truth input a manually sketched drawing containing a single rectangle representing the outer walls of the building has been used (Figure 7 (b)). The exact dimension of the rectangle has been taken

Table 1 Quantitative results of different approaches/datasets on the translational error.

Translational error m (abs) / m^2 (sqr)	DCMapping	RBPF (50 part.)
082er		
abs. errors	**0.072 ± 0.066**	0.115 ± 0.122
squared errors	**0.01 ± 0.024**	0.028 ± 0.074
Maximum abs. error	**0.73**	1.06
aces		
abs. errors	0.121 ± 0.335	**0.068 ± 0.078**
squared errors	0.127 ± 0.719	**0.011 ± 0.035**
Maximum abs. error	2.803	**0.646**
tu-darmstadt		
abs. errors	0.228 ± 0.643	**0.122 ± 0.146**
squared errors	0.465 ± 2.513	**0.036 ± 0.188**
Maximum abs. error	5.942	**1.921**
tu-graz		
abs. errors	**0.054 ± 0.044**	0.112 ± 0.186
squared errors	**0.005 ± 0.009**	0.047 ± 0.312
Maximum abs. error	**0.318**	2.515
imtek		
abs. errors	0.42 ± 0.942	**0.25 ± 0.416**
squared errors	1.063 ± 4.152	**0.235 ± 2.073**
Maximum abs. error	8.463	**6.998**
intel-lab		
abs. errors	0.136 ± 0.132	**0.07 ± 0.082**
squared errors	0.036 ± 0.068	**0.012 ± 0.033**
Maximum abs. error	0.8	**0.687**
mit-killian		
abs. errors	**3.71 ± 12.046**	7.505 ± 26.137
squared errors	**158.832 ± 639.92**	739.314 ± 3112.13
Maximum abs. error	**60.17**	153.087
telemax_hard		
abs. errors	**0.108 ± 0.136**	0.274 ± 0.276
squared errors	**0.03 ± 0.109**	0.152 ± 0.314
Maximum abs. error	**1.334**	1.694
hangar		
abs. errors	**0.207 ± 0.443**	0.291 ± 0.527
squared errors	**0.239 ± 1.215**	0.362 ± 1.648
Maximum abs. error	**3.044**	3.646
dc-maze		
abs. errors	**0.173 ± 0.199**	1.490 ± 2.230
squared errors	**0.070 ± 0.164**	7.179 ± 13.600
Maximum abs. error	**0.967**	7.180

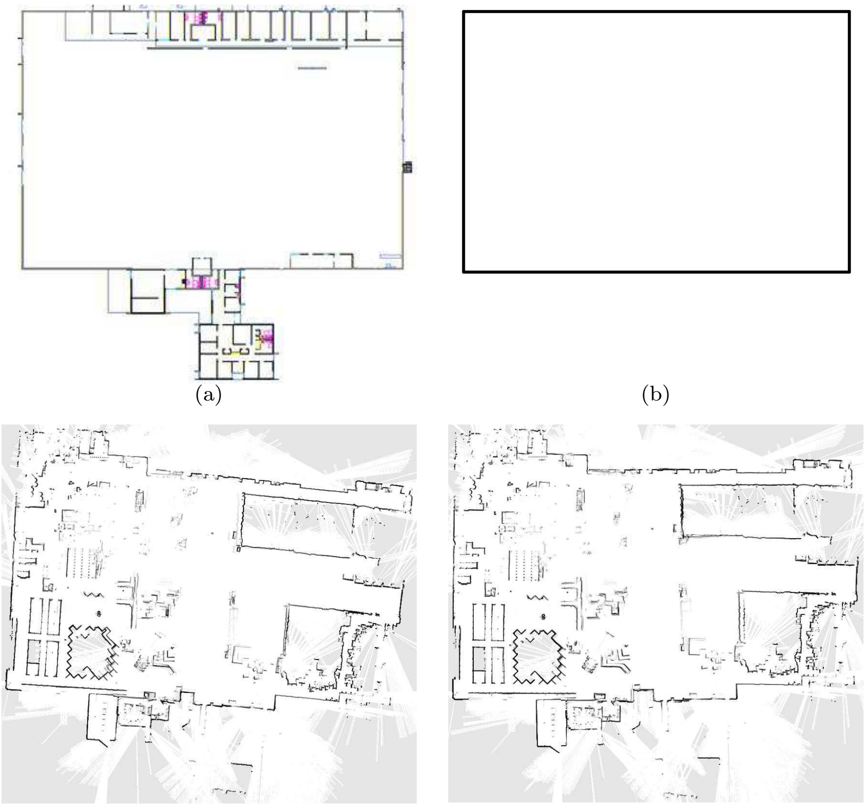
from a CAD drawing provided by SWRI (Figure 7 (a)). Figure 7 (d) depicts the result after the global alignment of the map.

The second data set shown in Figure 8 has been recorded by a person carrying the *simple setup* through the depicted outdoor campus of the University of Freiburg. The simple setup consisted of a Hokuyo UTM30 laser range finder and the MTi IMU from Xsens. Here also the input to the assisted loop-closure has been computed by DCMapping, the result is shown by Figure 8 (c). As can be seen the map has a strong deformation particularly in the upper-right corner nearby the parking lot. This is because the laser range finder was not able to capture any features expect the building wall on the left hand side. As ground truth input the geo-tagged aerial image taken from GoogleEarth (Figure 8 (a)) and the computed Canny edge image (Figure 8 (b)) were used. The result from the alignment procedure is shown in Figure 8 (d).

Table 2 Quantitative results of different approaches/datasets on the rotational error.

Rotational error *deg* (abs) / *deg*^{2} (sqr)	DCMapping	RBPF (50 part.)
082er		
abs. errors	**1.334 ± 1.571**	1.563 ± 1.877
squared errors	**4.245 ± 9.923**	5.964 ± 13.904
Maximum abs. error	**10.558**	12.501
aces		
abs. errors	2.518 ± 3.368	**1.675 ± 2.133**
squared errors	17.678 ± 72.465	**7.351 ± 19.952**
Maximum abs. error	45.015	**14.842**
tu-darmstadt		
abs. errors	0.667 ± 0.886	**0.558 ± 0.674**
squared errors	1.231 ± 3.875	**0.765 ± 2.477**
Maximum abs. error	7.628	**6.794**
tu-graz		
abs. errors	**1.13 ± 1.205**	1.354 ± 1.44
squared errors	**2.727 ± 6.551**	3.906 ± 10.847
Maximum abs. error	**9.09**	15.092
imtek		
abs. errors	**0.899 ± 1.28**	1.041 ± 1.476
squared errors	**2.445 ± 7.737**	3.26 ± 10.363
Maximum abs. error	**9.352**	10.457
intel-lab		
abs. errors	3.661 ± 6.048	**2.494 ± 3.63**
squared errors	49.968 ± 182.194	**19.395 ± 95.383**
Maximum abs. error	**47.267**	50.908
mit-killian		
abs. errors	**2.043 ± 3.781**	4.592 ± 14.08
squared errors	**18.463 ± 79.961**	219.299 ± 875.622
Maximum abs. error	**38.678**	71.114
telemax_hard		
abs. errors	**1.38 ± 1.395**	2.339 ± 2.791
squared errors	**3.849 ± 8.708**	13.251 ± 30.709
Maximum abs. error	**9.129**	14.778
hangar		
abs. errors	3.67 ± 4.096	**2.434 ± 2.696**
squared errors	30.222 ± 93.017	**13.183 ± 42.161**
Maximum abs. error	40.089	**21.584**
dc-maze		
abs. errors	**4.568 ± 3.649**	17.306 ± 20.294
squared errors	**34.148 ± 55.194**	710.284 ± 1051.333
Maximum abs. error	**20.300**	59.889

The last data set shown in Figure 9 has been recorded on a ActiveMedia PowerBot equipped with a Sick LMS200 laser range finder, wheel odometry, and the MTi IMU from Xsens. For this data set the robot has been steered through the outer area of the Freiburg hospital. In contrast to the other two logs the input to the assisted loop-closure has not been computed by DCMapping but by a standard scan matcher biased with wheel odometry and gyro readings. The result shown by Figure 9 (c) shows a strong global deformation of the map. As ground truth input the geo-tagged aerial image taken from GoogleEarth (Figure 9 (a)) and the computed Canny edge image (Figure 9 (b)) were used. The result from the alignment procedure is shown in Figure 9 (d). Notice that the data has been recorded during the winter, whereas the aerial image found on Google Earth has been captured during the summer time. Nevertheless, the data association has been successfully performed as shown by the result.



(a) (b) (c) (d)

Fig. 7 Map computed in a 66 m X 80 m hangar of the Southwest Research Institute (SWRI): (a) ground truth from CAD data, (b) manually sketched canny image from the outer walls according to the ground truth, (c) map generated by DCMapping, (d) output from DCMapping improved by assisted loop-closure. Note that several structures within the hangar are not found on the ground truth map. Gaining global consistency was nevertheless possible by utilizing the outer building walls only.

7 Conclusion

We introduced an online mapping system for assisting navigation tasks of first responders teleoperating a robot in critical domains. The initial goal of the system was the robust computation of locally consistent maps that are then presented in real-time on a operator console. The system has been intensively evaluated against this goal on both diverse environments and robot platforms. As the presented results show, the quality of the produced maps comes close to the one generated by computational costive algorithms. For example, although requiring remarkably less computational resources the map accuracy achieved by DCMapping turned out to be comparable with the one achieved by RBPF.

During the first responder evaluation in Disaster City as well as during several conversations with bomb squads we learned that the system is in deed applicable in realistic scenarios. It finally has been marked as advantageous by first responders

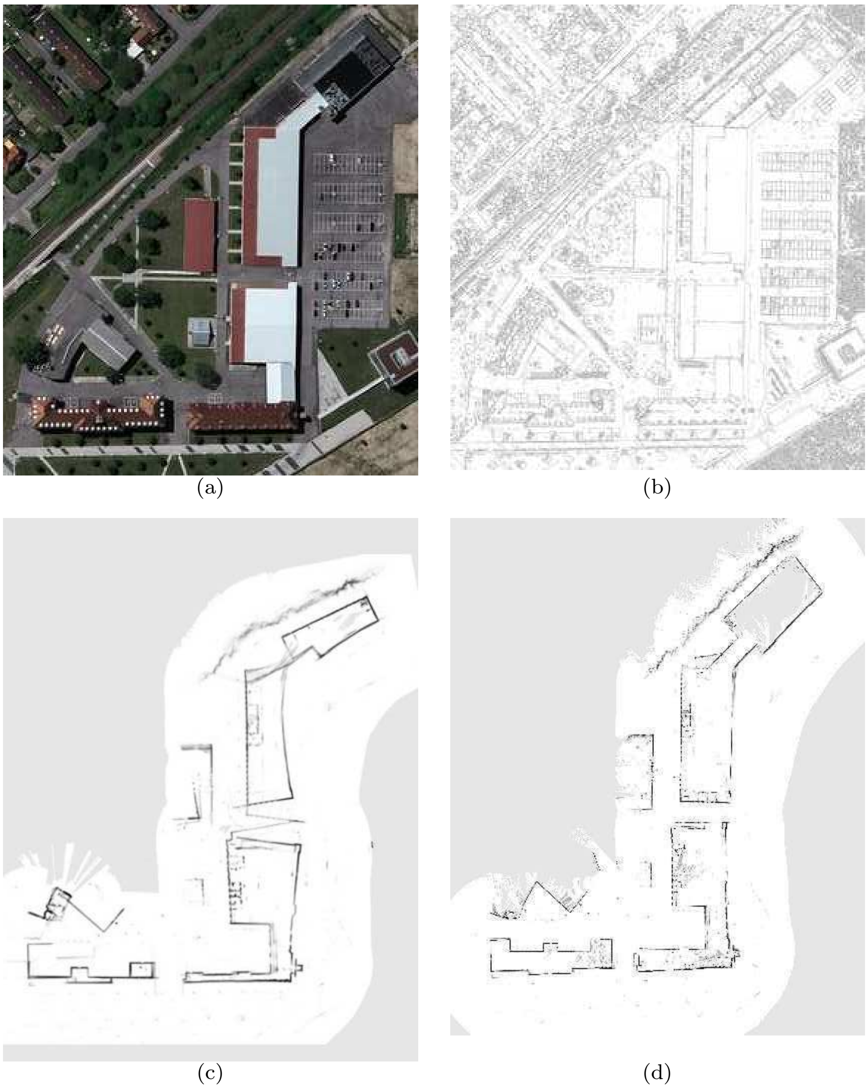
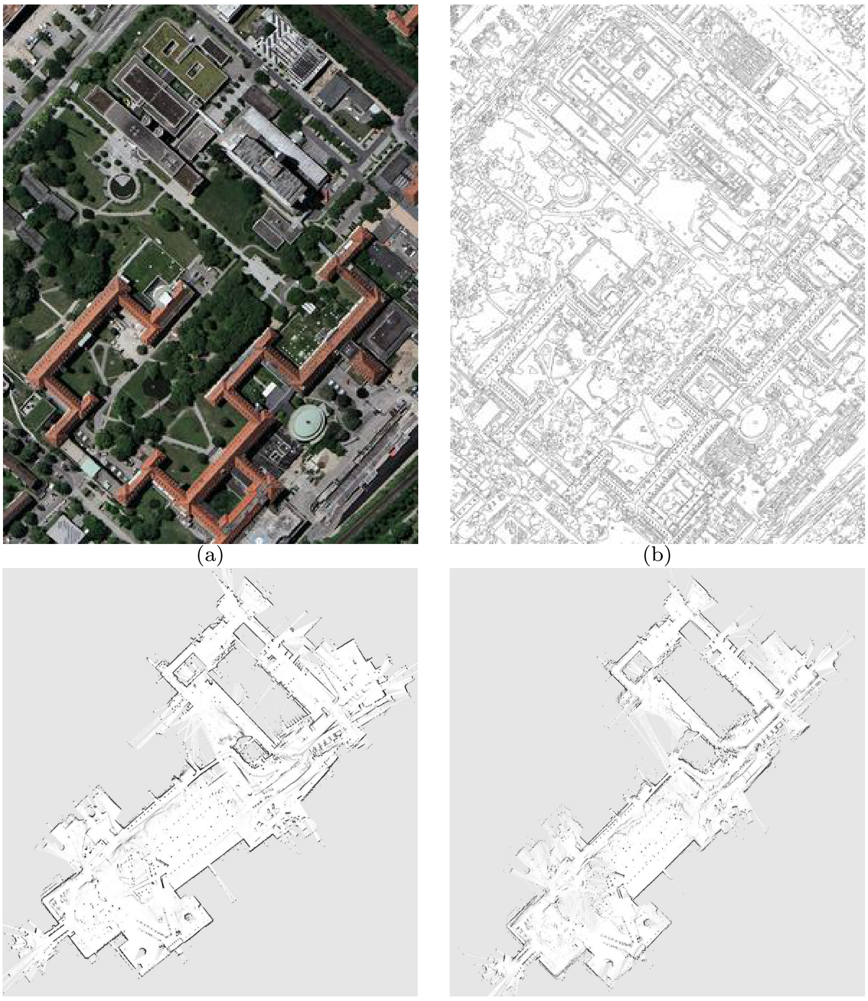


Fig. 8 Map generated on a 266 m X 180 m campus area of the Freiburg University: (a) ground truth aerial image, (b) extracted canny image, (c) map generated by DCMapping, (d) output from DCMapping improved by assisted loop-closure.

for teleoperation in their specific scenarios. The fact that the system is now being implemented on a commercial robot platform was mainly motivated from their demand.

We furthermore presented a method for improving locally consistent maps towards global consistence. The method combines the strength of human contextual understanding when working with geo-tagged aerial images with the accuracy of a MCL variant applied on their Canny edge counterparts. We demonstrated on three large-scale data sets the advantage of this method.



(a) (b)

(c) (d)

Fig. 9 Map generated on a 300 m X 350 m area of the Freiburg Hospital: (a) ground truth aerial image, (b) extracted canny image, (c) map generated by scan matching, (d) output from scan matching improved by assisted loop-closure.

Finally, we described the complete setup and algorithms that can be adopted by the reader to implement a fully functional mapping system on any specific hardware platform.

References

[Burgard *et al.*, 2009] W. Burgard, C. Stachniss, G. Grisetti, B. Steder, R. Kümmerle, C. Dornhege, M. Ruhnke, A. Kleiner, and Juan D. Tardós. A comparison of slam algorithms based on a graph of relations. In *Proc. of the IEEE/RSJ Int. Conf. on Intelligent Robots & Systems (IROS)*, St. Louis, MO, USA, October 2009.

[Canny, 1986] J. Canny. A computational approach to edge detection. *IEEE Trans. Pattern Anal. Mach. Intell.*, 8(6):679–698, 1986.

[Chen and Wang, 2007] C. Chen and H. Wang. Large-scale loop-closing by fusing range data and aerial image. 22(2):160–169, 2007.

[Cox, 1991] I. J. Cox. Blanche: Position estimation for an autonomous robot vehicle. In S. S. Iyengar and A. Elfes, editors, *Autonomous Mobile Robots: Control, Planning, and Architecture (Vol. 2)*, pages 285–292. IEEE Computer Society Press, Los Alamitos, CA, 1991.

[Dellaert, 2005] F. Dellaert. Square Root SAM. In *Proc. of Robotics: Science and Systems (RSS)*, pages 177–184, Cambridge, MA, USA, 2005.

[Eustice *et al.*, 2005] R. Eustice, H. Singh, and J.J. Leonard. Exactly sparse delayed-state filters. In *Proc. of the IEEE Int. Conf. on Robotics & Automation (ICRA)*, pages 2428–2435, Barcelona, Spain, 2005.

[Früh and Zakhor, 2004] C. Früh and A. Zakhor. An automated method for large-scale, ground-based city model acquisition. *International Journal of Computer Vision*, 60:5–24, 2004.

[Grisetti *et al.*, 2005a] G. Grisetti, Stachniss C., and Burgard W. Improving grid-based SLAM with rao-blackwellized particle filters by adaptive proposals and selective resampling. In *Proc. of the IEEE Int. Conf. on Robotics & Automation (ICRA)*, pages 667–672, Barcelona, Spain, 2005.

[Grisetti *et al.*, 2005b] G. Grisetti, C. Stachniss, and W. Burgard. Improving grid-based SLAM with Rao-Blackwellized particle filters by adaptive proposals and selective resampling. In *Proc. of the IEEE Int. Conf. on Robotics & Automation (ICRA)*, pages 2443–2448, Barcelona, Spain, 2005.

[Grisetti *et al.*, 2007a] G. Grisetti, S. Grzonka, C. Stachniss, P. Pfaff, and W. Burgard. Efficient estimation of accurate maximum likelihood maps in 3D. In *Proc. of the Int. Conf. on Intelligent Robots and Systems (IROS)*, San Diego, CA, USA, 2007.

[Grisetti *et al.*, 2007b] G. Grisetti, C. Stachniss, and W. Burgard. Improved techniques for grid mapping with rao-blackwellized particle filters. *IEEE Transactions on Robotics*, 23:34–46, 2007.

[Grisetti *et al.*, 2007c] G. Grisetti, C. Stachniss, S. Grzonka, and W. Burgard. A tree parameterization for efficiently computing maximum likelihood maps using gradient descent. In *Proc. of Robotics: Science and Systems (RSS)*, Atlanta, GA, USA, 2007.

[Grisetti *et al.*, 2009] G. Grisetti, C. Stachniss, and W. Burgard. Non-linear constraint network optimization for efficient map learning. 2009. In press.

[Gutmann and Konolige, 1999] J.-S. Gutmann and K. Konolige. Incremental mapping of large cyclic environments. In *Proc. of the IEEE Int. Symposium on Computational Intelligence in Robotics and Automation (CIRA)*, 1999.

[Gutmann, 2000] J.-S. Gutmann. *Robuste Navigation autonomer mobiler Systeme.* PhD thesis, Albert-Ludwigs-Universit"at Freiburg, 2000. ISBN 3-89838-241-9.

[Hähnel, 2005] D. Hähnel. *Mapping with Mobile Robots.* PhD thesis, Universität Freiburg, Freiburg, Deutschland, 2005.

[Julier *et al.*, 1995] S. Julier, J. Uhlmann, and H. Durrant-Whyte. A new approach for filtering nonlinear systems. In *Proc. of the American Control Conference*, pages 1628–1632, Seattle, WA, USA, 1995.

[Kaess *et al.*, 2007] M. Kaess, A. Ranganathan, and F. Dellaert. iSAM: Fast incremental smoothing and mapping with efficient data association. In *Proc. of the IEEE Int. Conf. on Robotics & Automation (ICRA)*, Rome, Italy, 2007.

[Kleiner *et al.*, 2009] A. Kleiner, C. Scrapper, and A. Jacoff. Robocuprescue interleague challenge 2009: Bridging the gap between simulation and reality. In *Proc. of the Int. Workshop on Performance Metrics for Intelligent Systems (PerMIS)*, Washington D.C., USA, 2009. To appear.

[Korah and Rasmussen, 2004] T. Korah and C. Rasmussen. Probabilistic contour extraction with model-switching for vehicle localization. *Intelligent Vehicles Symposium, 2004 IEEE*, pages 710–715, June 2004.

[Kümmerle *et al.*, 2009] R. Kümmerle, B. Steder, C. Dornhege, A. Kleiner, G. Grisetti, and W. Burgard. Large scale graph-based SLAM using aerial images as prior information. In *Proceedings of Robotics: Science and Systems (RSS)*, Seattle, WA, USA, June 2009.

[Leonard and Durrant-Whyte, 1991] J.J. Leonard and H.F. Durrant-Whyte. Mobile robot localization by tracking geometric beacons. *IEEE Transactions on Robotics and Automation*, 7(4):376–382, 1991.

[Leung *et al.*, 2008] Keith Yu Kit Leung, Christopher M. Clark, and Jan P. Huissoon. Localization in urban environments by matching ground level video images with an aerial image. In *Proc. of the IEEE Int. Conf. on Robotics & Automation (ICRA)*, Pasadena, CA, USA, 2008.

[Lu and Milios, 1997a] F. Lu and E. Milios. Globally consistent range scan alignment for environment mapping. *Journal of Autonomous Robots*, 4:333–349, 1997.

[Lu and Milios, 1997b] Feng Lu and Evangelos Milios. Robot pose estimation in unknown environments by matching 2d range scans. *J. Intell. Robotics Syst.*, 18(3):249–275, 1997.

[Montemerlo *et al.*, 2003] M. Montemerlo, S. Thrun, D. Koller, and B. Wegbreit. FastSLAM 2.0: An improved particle filtering algorithm for simultaneous localization and mapping that provably converges. In *Proc. of the Int. Conf. on Artificial Intelligence (IJCAI)*, pages 1151–1156, Acapulco, Mexico, 2003.

[Moravec and Elfes, 1985] H.P. Moravec and A.E. Elfes. High resolution maps from wide angle sonar. In *Proc. of the IEEE Int. Conf. on Robotics & Automation (ICRA)*, pages 116–121, 1985.

[Moravec, 1988] Hans P. Moravec. Sensor fusion in certainty grids for mobile robots. *AI Magazine*, pages 61–74, 1988.

[Olson *et al.*, 2006] E. Olson, J. Leonard, and S. Teller. Fast iterative optimization of pose graphs with poor initial estimates. In *Proc. of the IEEE Int. Conf. on Robotics & Automation (ICRA)*, pages 2262–2269, 2006.

[Ranganathan *et al.*, 2007] A. Ranganathan, M. Kaess, and F. Dellaert. Loopy sam. In *Proc. of the Int. Conf. on Artificial Intelligence (IJCAI)*, 2007.

[Smith *et al.*, 1990] R. Smith, M. Self, and P. Cheeseman. Estimating uncertain spatial realtionships in robotics. In I. Cox and G. Wilfong, editors, *Autonomous Robot Vehicles*, pages 167–193. Springer Verlag, 1990.

[Stachniss *et al.*, 2007] C. Stachniss, U. Frese, and G. Grisetti. OpenSLAM.org – give your algorithm to the community. http://www.openslam.org, 2007.

[Thrun *et al.*, 2004] S. Thrun, Y. Liu, D. Koller, A.Y. Ng, Z. Ghahramani, and H. Durrant-Whyte. Simultaneous localization and mapping with sparse extended information filters. *Int. Journal of Robotics Research*, 23(7/8):693–716, 2004.

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

[Uhlmann, 1995] J. Uhlmann. *Dynamic Map Building and Localization: New Theoretical Foundations*. PhD thesis, University of Oxford, 1995.