# A Flexible and Scalable SLAM System with Full 3D Motion Estimation

Stefan Kohlbrecher and Oskar von Stryk
Technische Universität Darmstadt
Hochschulstraße 10
Darmstadt, Germany
kohlbrecher,stryk@sim.tu-darmstadt.de

Johannes Meyer and Uwe Klingauf
Technische Universität Darmstadt
Petersenstraße 30
Darmstadt, Germany
meyer,klingauf@fsr.tu-darmstadt.de

***Abstract*—For many applications in Urban Search and Rescue (USAR) scenarios robots need to learn a map of unknown environments. We present a system for fast online learning of occupancy grid maps requiring low computational resources. It combines a robust scan matching approach using a LIDAR system with a 3D attitude estimation system based on inertial sensing. By using a fast approximation of map gradients and a multi-resolution grid, reliable localization and mapping capabilities in a variety of challenging environments are realized. Multiple datasets showing the applicability in an embedded hand-held mapping system are provided. We show that the system is sufficiently accurate as to not require explicit loop closing techniques in the considered scenarios. The software is available as an open source package for ROS.**

***Keywords***: *Simultaneous Localization and Mapping, Inertial Navigation, Robust and Fast Localization*

## I. Introduction

The ability to learn a model of the environment and to localize itself is one of the most important abilities of truly autonomous robots able to operate within real world environments. In this paper, we present a flexible and scalable system for solving the SLAM (Simultaneous Localization and Mapping) problem that has successfully been used on unmanned ground vehicles (UGV), unmanned surface vehicles (USV) and a small indoor navigation system. The approach consumes low computational resources and thus can be used on low-weight, low-power and low-cost processors such as those commonly used on small-scale autonomous systems. Our approach uses the ROS meta operating system [1] as middleware and is available as open source software. It honors the API of the the ROS navigation stack and thus can easily be interchanged with other SLAM approaches available in the ROS ecosystem.

The system introduced in this paper aims at enabling sufficiently accurate environment perception and self-localization while keeping computational requirements low. It can be used for SLAM in small scale scenarios where large loops do not have to be closed and where leveraging the high update rate of modern LIDAR systems is beneficial. Such scenarios include the RoboCup Rescue competition, where robots have to find victims in simulated earthquake scenarios which feature rough terrain and thus require full 6DOF motion estimation of vehicles, or the indoor navigation of agile aerial vehicles that move fast compared to ground robots. Previous results where the system has been used in the context of building semantic world models in USAR environments are available in [2].

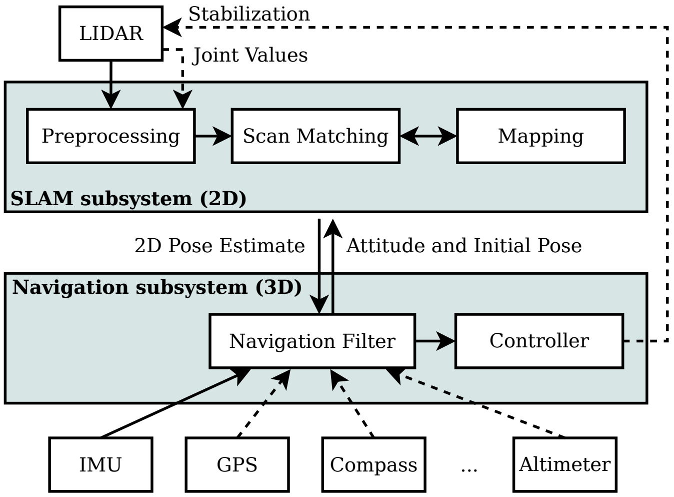


Fig. 1. Overview of the mapping and navigation system (dashed lines depict optional information)

Our approach combines a 2D SLAM system based on the integration of laser scans (LIDAR) in a planar map and an integrated 3D navigation system based on an inertial measurement unit (IMU), which incorporates the 2D information from the SLAM subsystem as one possible source of aiding information (Fig. I). While SLAM usually runs in soft real-time triggered by the updates of the laser scanner device, the full 3D navigation solution is calculated in hard real-time and usually forms a part of the vehicle control system.

## II. Related Work

There has been a wealth of research into the SLAM problem in recent years, with reliably working solutions for typical office-like indoor scenarios using Rao-Blackwellized particle filters like Gmapping [3] being available as open source software. However, these solutions work best in planar environments, rely on available, sufficiently accurate odometry and do not leverage the high update rate provided by modern LIDAR systems. For unstructured environments, that lead to significant roll and pitch motion of the carrier, or implementation on aerial platforms such systems are not applicable or have to be modified significantly.

A distinction between SLAM frontend and backend system has to be made. While SLAM frontends are used to estimate robot movement online in real-time, the backend is used to perform optimization of the pose graph given constraints between poses that have been generated before using the frontend. The approach presented in this paper serves as a SLAM frontend and does not provide pose graph optimization like the solutions presented in [4] and [5]. However, we show that in many scenarios such optimizations are not needed under real world conditions as the approach is sufficiently accurate for robots to perform their mission.

Multiple indoor navigation systems based on laser scan matching have been presented for the use on Quadrotor UAVs [6], [7], [8]. Here, a two-stage approach is employed, with a front end fast scan alignment step for pose estimation and a slower backend mapping step running in the background or on a remote computer. The pose estimates from scan alignment are not directly incorporated in the vehicle's control loop and thus they move at a low speed only.

In [9] and [10] other front end systems used on mobile robots are described. In contrast to system presented in this paper they do not provide a full 6DOF pose estimate and are not available as open source software.

Work on localization using scan matching started with the Iterative Closest Point (ICP) [11] which originated as a general approach for registering 3D point clouds. The main drawback of many ICP-based methods is the expensive search for point correspondences, which has to be done in every iteration. Polar Scan Matching (PSM) [12] avoids the correspondence search by taking advantage of the natural polar coordinate system of the laser scans to estimate a match between them. Scans have to be preprocessed to be used in the polar scan matcher. The real-time correlative scan matching approach [13] uses an exhaustive sampling based approach for scan matching. Using several optimizations this approach is capable of real-time application. Normal Distribution Transform (NDT) [14] based scan matching aligns scans to a mixture of normal distributions representing preceding scans.

For littoral water scenarios there has been research into using expensive multi-sensor scanners [15], but to the authors' knowledge there is no single emitter LIDAR-based SLAM approach available that was tested under real world conditions.

## III. System Overview

As the presented system has to be usable on platforms that exhibit full 6DOF motion as opposed to the 3DOF motion assumed in many other 2D grid-based SLAM algorithms, our system has to estimate the full 6DOF state consisting of translation and rotation of the platform. To achieve this, the system consists of two major components. A navigation filter fuses information from the inertial measurement unit and other available sensors to form a consistent 3D solution, while a 2D SLAM system is used to provide position and heading information within the ground plane. Both estimates are updated individually and only loosely coupled so that they remain synchronized over time.

We define the navigation coordinate system as a right-handed system having the origin at the starting point of the platform with the z axis pointing upwards and the x axis pointing into the yaw direction of the platform at startup. The full 3D state is represented by $\mathbf{x} = \begin{pmatrix}\mathbf{\Omega}^{\mathrm{T}} & \mathbf{p}^{\mathrm{T}} & \mathbf{v}^{\mathrm{T}}\end{pmatrix}^{\mathrm{T}}$ , where $\mathbf{\Omega} = (\phi, \theta, \psi)^{\mathrm{T}}$ are the roll, pitch and yaw Euler angles, and $\mathbf{p} = (p_x, p_y, p_z)^{\mathrm{T}}$ and $\mathbf{v} = (v_x, v_y, v_z)^{\mathrm{T}}$ are the position and velocity of the platform expressed in the navigation frame.

The inertial measurements constitute the input vector $\mathbf{u} = \begin{pmatrix}\boldsymbol{\omega}^{\mathrm{T}} & \mathbf{a}^{\mathrm{T}}\end{pmatrix}^{\mathrm{T}}$ with the body-fixed angular rates $\boldsymbol{\omega} = (\omega_x, \omega_y, \omega_z)^{\mathrm{T}}$ and accelerations $\mathbf{a} = (a_x, a_y, a_z)^{\mathrm{T}}$. The motion of an arbitrary rigid body is described by the nonlinear differential equation system

$$\dot{\mathbf{\Omega}} = \mathbf{E}_{\Omega} \cdot \boldsymbol{\omega} \quad (1)$$

$$\dot{\mathbf{p}} = \mathbf{v} \quad (2)$$

$$\dot{\mathbf{v}} = \mathbf{R}_{\Omega} \cdot \mathbf{a} + \mathbf{g} \quad (3)$$

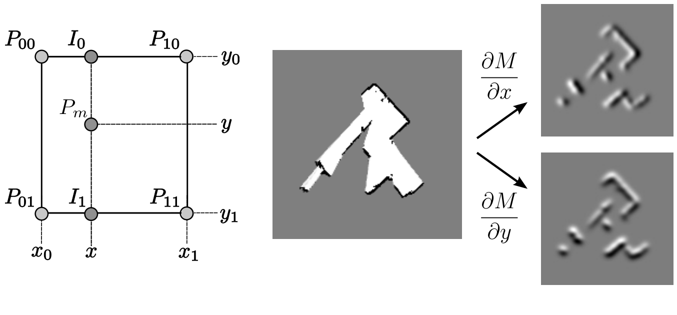
where $\mathbf{R}_{\Omega}$ is the direction cosine matrix that maps a vector in the body frame to the navigation frame. $\mathbf{E}_{\Omega}$ maps the body-fixed angular rates to the derivatives of the Euler angles and $\mathbf{g}$ is the constant gravity vector [16]. The effects of pseudo forces due to the earth's rotation can usually be neglected when low-cost sensors are used.

Due to sensor noise the integrated velocity and position exhibit significant drift. Therefore further sensor information has to be integrated. In this paper, this information is provided by the scan matcher, which is well suited for use in indoor scenarios. Other possible sources are magnetic field sensors for heading information or barometric pressure sensor for altitude estimation. If available, wheel odometry can be used to provide measurements of velocity. In outdoor scenarios, satellite navigation systems are commonly used as an aiding system to prevent the inertial navigation solution from drifting [17].

Depending on the target platform, further constraints can be introduced in the system equation, resulting in a reduction of the state space.

## IV. 2D SLAM

To be able to represent arbitrary environments an occupancy grid map is used, which is a proven approach for mobile robot localization using LIDARs in real-world environments [18]. As the LIDAR platform might exhibit 6DOF motion, the scan has to be transformed into a local stabilized coordinate frame using the estimated attitude of the LIDAR system. Using the estimated platform orientation and joint values, the scan is converted into a point cloud of scan endpoints. Depending



(a) (b)

Fig. 2. (a) Bilinear filtering of the occupancy grid map. Point $P_m$ is the point whose value shall be interpolated. (b) Occupancy grid map and spatial derivatives.

on the scenario, this point cloud can be preprocessed, for example by downsampling the number of points or removal of outliers. For the presented approach, only filtering based on the endpoint $z$ coordinate is used, so that only endpoints within a threshold of the intended scan plane are used in the scan matching process.

### A. Map Access

The discrete nature of occupancy grid maps limits the precision that can be achieved and also does not allow the direct computation of interpolated values or derivatives. For this reason an interpolation scheme allowing sub-grid cell accuracy through bilinear filtering is employed for both estimating occupancy probabilities and derivatives. Intuitively, the grid map cell values can be viewed as samples of an underlying continuous probability distribution.

Given a continuous map coordinate $P_m$, the occupancy value $M(P_m)$ as well as the gradient $\nabla M(P_m) = \left(\frac{\partial M}{\partial x}(P_m), \frac{\partial M}{\partial y}(P_m)\right)$ can be approximated by using the four closest integer coordinates $P_{00..11}$ as depicted in Fig. 2(a). Linear interpolation along the x- and y-axis then yields

$$\begin{aligned} M(P_m) \approx \frac{y-y_0}{y_1-y_0}\left(\frac{x-x_0}{x_1-x_0}M(P_{11}) + \frac{x_1-x}{x_1-x_0}M(P_{01})\right) \\ + \frac{y_1-y}{y_1-y_0}\left(\frac{x-x_0}{x_1-x_0}M(P_{10}) + \frac{x_1-x}{x_1-x_0}M(P_{00})\right) \end{aligned} \tag{4}$$

The derivatives can be approximated by:

$$\begin{aligned} \frac{\partial M}{\partial x}(P_m) \approx & \frac{y-y_0}{y_1-y_0}(M(P_{11}) - M(P_{01})) \\ & + \frac{y_1-y}{y_1-y_0}(M(P_{10}) - M(P_{00})) \end{aligned} \tag{5}$$

$$\begin{aligned} \frac{\partial M}{\partial y}(P_m) \approx & \frac{x-x_0}{x_1-x_0}(M(P_{11}) - M(P_{10})) \\ & + \frac{x_1-x}{x_1-x_0}(M(P_{01}) - M(P_{00})) \end{aligned} \tag{6}$$

It should be noted that the sample points/grid cells of the map are situated on a regular grid with distance 1 (in map coordinates) from each other, which simplifies the presented equations for the gradient approximation.

### B. Scan Matching

Scan matching is the process of aligning laser scans with each other or with an existing map. Modern laser scanners have low distance measurement noise and high scan rates. A method for registering scans might yield very accurate results for this reason. For many robot systems the accuracy and precision of the laser scanner is much higher than that of odometry data, if available at all. The evaluation scenarios presented in section VI show examples where odometry data is not available.

Our approach is based on optimization of the alignment of beam endpoints with the map learnt so far. The basic idea using a Gauss-Newton approach is inspired by work in computer vision [19]. Using this approach, there is no need for a data association search between beam endpoints or an exhaustive pose search. As scans get aligned with the existing map, the matching is implicitly performed with all preceding scans.

We seek to find the rigid transformation $\boldsymbol{\xi} = (p_x, p_y, \psi)^{\mathrm{T}}$ that minimizes

$$\boldsymbol{\xi}^* = \underset{\boldsymbol{\xi}}{\operatorname{argmin}} \sum_{i=1}^{n} \left[1 - M(\mathbf{S}_i(\boldsymbol{\xi}))\right]^2 \tag{7}$$

that is, we want to find the transformation that gives the best alignment of the laser scan with the map. Here, $\mathbf{S}_i(\boldsymbol{\xi})$ are the world coordinates of scan endpoint $\mathbf{s}_i = (s_{i,x}, s_{i,y})^{\mathrm{T}}$. They are a function of $\boldsymbol{\xi}$, the pose of the robot in world coordinates:

$$\mathbf{S}_i(\boldsymbol{\xi}) = \begin{pmatrix} \cos(\psi) & -\sin(\psi) \\ \sin(\psi) & cos(\psi) \end{pmatrix} \begin{pmatrix} s_{i,x} \\ s_{i,y} \end{pmatrix} + \begin{pmatrix} p_x \\ p_y \end{pmatrix} \tag{8}$$

The function $M(\mathbf{S}_i(\boldsymbol{\xi}))$ returns the map value at the coordinates given by $\mathbf{S}_i(\boldsymbol{\xi})$. Given some starting estimate of $\boldsymbol{\xi}$, we want to estimate $\Delta\boldsymbol{\xi}$ which optimizes the error measure according to

$$\sum_{i=1}^{n} \left[1 - M(\mathbf{S}_i(\boldsymbol{\xi} + \Delta\boldsymbol{\xi}))\right]^2 \to 0\,. \tag{9}$$

By first order Taylor expansion of $M(\mathbf{S}_i(\boldsymbol{\xi} + \Delta\boldsymbol{\xi}))$ we get:

$$\sum_{i=1}^{n} \left[1 - M(\mathbf{S}_i(\boldsymbol{\xi})) - \nabla M(\mathbf{S}_i(\boldsymbol{\xi}))\frac{\partial \mathbf{S}_i(\boldsymbol{\xi})}{\partial \boldsymbol{\xi}}\Delta\boldsymbol{\xi})\right]^2 \to 0\,. \tag{10}$$

This equation is minimized by setting the partial derivative with respect to $\Delta\boldsymbol{\xi}$ to zero:

$$\begin{aligned} 2\sum_{i=1}^{n} \left[\nabla M(\mathbf{S}_i(\boldsymbol{\xi}))\frac{\partial \mathbf{S}_i(\boldsymbol{\xi})}{\partial \boldsymbol{\xi}}\right]^T \\ \left[1 - M(\mathbf{S}_i(\boldsymbol{\xi})) - \nabla M(\mathbf{S}_i(\boldsymbol{\xi}))\frac{\partial \mathbf{S}_i(\boldsymbol{\xi})}{\partial \boldsymbol{\xi}}\Delta\boldsymbol{\xi})\right] = 0 \end{aligned} \tag{11}$$

Solving for $\Delta\boldsymbol{\xi}$ yields the Gauss-Newton equation for the minimization problem:

$$\Delta\boldsymbol{\xi} = \mathbf{H}^{-1}\sum_{i=1}^{n} \left[\nabla M(\mathbf{S}_i(\boldsymbol{\xi}))\frac{\partial \mathbf{S}_i(\boldsymbol{\xi})}{\partial \boldsymbol{\xi}}\right]^T \left[1 - M(\mathbf{S}_i(\boldsymbol{\xi}))\right] \tag{12}$$

with

$$\mathbf{H} = \left[\nabla M(\mathbf{S}_i(\boldsymbol{\xi}))\frac{\partial \mathbf{S}_i(\boldsymbol{\xi})}{\partial \boldsymbol{\xi}}\right]^T \left[\nabla M(\mathbf{S}_i(\boldsymbol{\xi}))\frac{\partial \mathbf{S}_i(\boldsymbol{\xi})}{\partial \boldsymbol{\xi}}\right] \quad (13)$$

An approximation for the map gradient $\nabla M(\mathbf{S}_i(\boldsymbol{\xi}))$ is provided in section IV-A. With equation (8) we get

$$\frac{\partial S_i(\boldsymbol{\xi})}{\partial \boldsymbol{\xi}} = \begin{pmatrix} 1 & 0 & -\sin(\psi)s_{i,x} - \cos(\psi)s_{i,y} \\ 0 & 1 & \cos(\psi)s_{i,x} - \sin(\psi)s_{i,y} \end{pmatrix} \quad (14)$$

Using $\nabla M(\mathbf{S}_i(\boldsymbol{\xi}))$ and $\frac{\partial S_i(\boldsymbol{\xi})}{\partial \boldsymbol{\xi}}$, the Gauss-Newton equation (12) can now be evaluated, yielding a step $\Delta\boldsymbol{\xi}$ towards the minimum. It is important to note that the algorithm works on non-smooth linear approximations of the map gradient $\nabla M(\mathbf{S}_i(\boldsymbol{\xi}))$, meaning that local quadratic convergence towards a minimum cannot be guaranteed. Nevertheless, the algorithm works with sufficient accuracy in practice.

For many applications a Gaussian approximation of the match uncertainty is desirable. Examples are updates of parametric filters as well as pose constraints for use with graph optimization SLAM backends. One approach is to use a sampling based covariance estimate, sampling different pose estimates close to the scan matching pose and constructing a covariance from those. This is similar to the idea of the Unscented Kalman Filter [20]. A second method is the use of the approximate Hessian matrix to arrive at a covariance estimate. Here, the covariance matrix is approximated by

$$\mathbf{R} = \mathrm{Var}\{\boldsymbol{\xi}\} = \sigma^2 \cdot \mathbf{H}^{-1} \quad (15)$$

where $\sigma$ is a scaling factor dependent on the properties of the laser scanner device. A comprehensive derivation is available in [21].

### C. Multi-Resolution Map Representation

Any hill climbing/gradient based approach has the inherent risk of getting stuck in local minima. As the presented approach is based on gradient ascent, it also is potentially prone to get stuck in local minima. The problem is mitigated by using a multi-resolution map representation similar to image pyramid approaches used in computer vision. In our approach, we optionally use multiple occupancy grid maps with each coarser map having half the resolution of the preceding one. However, the multiple map levels are not generated from a single high resolution map by applying Gaussian filtering and downsampling as is commonly done in image processing. Instead, different maps are kept in memory and simultaneously updated using the pose estimates generated by the alignment process. This generative approach ensures that maps are consistent across scales while at the same time avoiding costly downsampling operations. The scan alignment process is started at the coarsest map level, with the resulting estimated pose getting used as the start estimate for the next level, similar to the approach presented in [22]. A positive side-effect is the immediate availability of coarse grained maps which can for example be used for path planning.

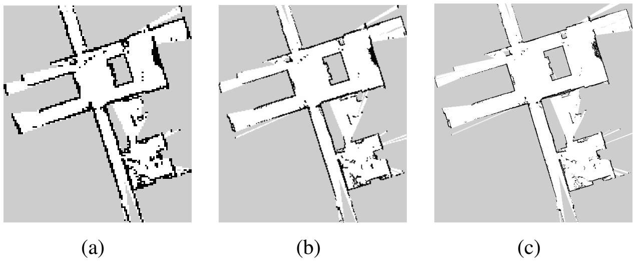

(a) (b) (c)

Fig. 3. Multiresolution representation of the map: (a): 20cm grid cell length (b) 10 cm grid cell length (c) 5cm grid cell length

## V. 3D State Estimation

This section covers the estimation of the full 3D state vector according to section III and the integration of the 2D SLAM solution. The navigation filter runs in real-time at a constant rate of 100 Hz and is asynchronously updated with the scan matching pose and other sensor information as soon as they arrive. The fusion of other sensors than the scan matching pose is not within the scope of this paper. Implementation details can be found in [23].

### A. Navigation Filter

For estimating the 6D pose of the platform we use an Extended Kalman Filter (EKF) with the general platform model defined by equations (1)-(3). In addition, the state vector is augmented with the biases of the gyroscopes and the accelerometers as they vary over time and influence the result significantly. Note that the system equation is non-linear due to the Euler angle terms in the matrices $\mathbf{E}_\Omega$ and $\mathbf{R}_\Omega$ and therefore a non-linear filter has to be used. The inertial measurements are considered as known system inputs.

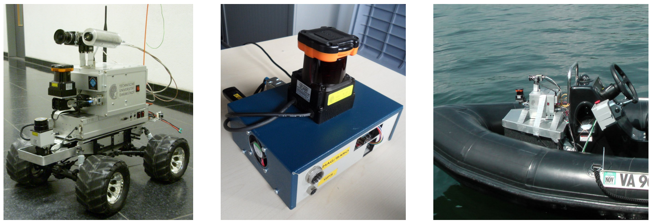
The velocity and position update is a pure integration of the measured accelerations and the system would be unstable without additional feedback through measurement updates. An usual countermeasure to prevent the state estimate to grow indefinitely when no measurements are available are pseudo zero-velocity updates as soon as the variance reaches a certain threshold and stability cannot be assured otherwise. In this paper, the 2D position and orientation in the plane is updated by the scan matcher, while for full 3D estimation an additional height sensor such as a barometer or range sensor is needed.

### B. SLAM Integration

For best performance, information between the 2D SLAM solution and the 3D EKF estimate has to be exchanged in both directions. The systems are not synchronized and the EKF usually runs at a higher update rate.

To improve performance of the scan matching process, the pose estimate of the EKF is projected on the xy-plane and is used as start estimate for the optimization process of the scan matcher. Alternatively, the estimated velocities and angular rates can be integrated to provide a scan matching start estimate.

In the opposite direction, covariance intersection (CI) is used to fuse the SLAM pose with the full belief state [24]. A



(a) (b) (c)

Fig. 4. Platforms: (a) Hector UGV system used in the simulated USAR scenario of the RoboCup Rescue League. (b) Self-contained embedded mapping system for handheld mapping. (c) UGV mapping system mounted on a USV platform.

simple Kalman measurement update would lead to overconfident estimates because it assumes statistically independent measurement errors.

We denote the Kalman estimate at the time of the scan as $\hat{\mathbf{x}}$ with covariance $\mathbf{P}$ and the SLAM pose $(\boldsymbol{\xi}^*, \mathbf{R})$ as defined in (7) and (15). The fusion result is then given by

$$(\mathbf{P}^+)^{-1} = (1-\omega)\cdot\mathbf{P}^{-1} + \omega\cdot\mathbf{C}^{\mathrm{T}}\mathbf{R}^{-1}\mathbf{C} \qquad \text{and} \tag{16}$$

$$\hat{\mathbf{x}}^+ = \mathbf{P}^+\left((1-\omega)\cdot\mathbf{P}^{-1}\hat{\mathbf{x}} + \omega\cdot\mathbf{C}^{\mathrm{T}}\mathbf{R}^{-1}\boldsymbol{\xi}^*\right)^{-1} \tag{17}$$

with the observer matrix $\mathbf{C}$ that projects the full state space into the 3-dimensional subspace of the SLAM system. The parameter $\omega \in (0,1)$ is chosen to tune the effect of the SLAM update.

In analogy to the dualism between the Kalman Filter and the Information Filter the covariance intersection can also be written in its covariance form

$$\mathbf{P}^+ = \mathbf{P} - (1-\omega)^{-1}\cdot\mathbf{KCP} \qquad \text{and} \tag{18}$$

$$\hat{\mathbf{x}}^+ = \hat{\mathbf{x}} + \mathbf{K}(\boldsymbol{\xi}^* - \mathbf{C}\hat{\mathbf{x}}) \tag{19}$$

with

$$\mathbf{K} = \mathbf{PC}^{\mathrm{T}}\left(\frac{1-\omega}{\omega}\cdot\mathbf{R} + \mathbf{C}^{\mathrm{T}}\mathbf{PC}\right)^{-1} . \tag{20}$$

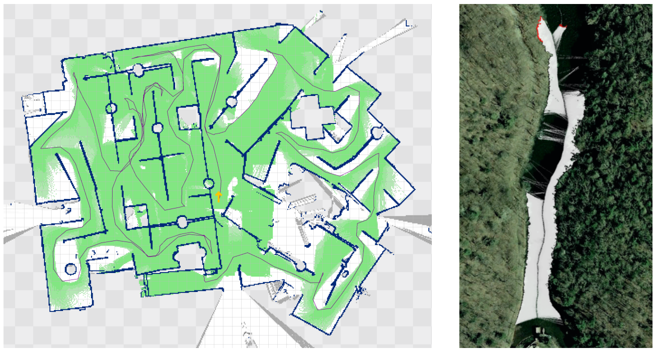
As inversion of the full state covariance according to (16) and (17) is computationally expensive, this is the preferred approach here.

## VI. Results

We show the examples of the versatility and robustness of the presented system for three different use cases. It should be noted that standard datasets used for benchmarking SLAM approaches lack the 6DOF motion considered for our approach as well as the high LIDAR update rate that it leverages. For this reason, we provide datasets in the ROS bag format for benchmarking.

### A. USAR Scenario

The presented approach is used on UGVs performing autonomous exploration and victim detection in Urban Search and Rescue (USAR) scenarios like the RoboCup Rescue Robot League competition. The Hector UGV system is depicted in Fig. 4(a). The LIDAR is stabilized around the roll and pitch axes to keep the laser aligned with the ground plane and maximize information gain from the planar LIDAR scans. As can be seen in Fig. 5(a) and 6(a), our approach is capable of learning highly accurate and consistent maps despite significant changes in the 6DOF state of the UGV system. Video showing online mapping in the RoboCup Rescue arena at RoboCup 2010 as well as the other experiments in this section is available online[1].



(a) (b)

Fig. 5. Vehicle experiments: (a): Map learned at the RoboCup 2010 competition. (b) Learned grid map of 0.25m resolution overlaid over satellite imagery (© Google, Commonwealth of Virginia)

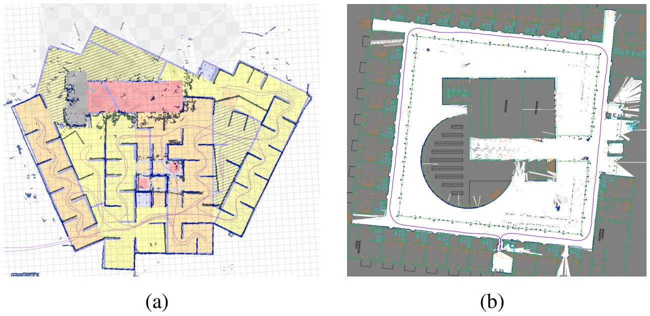
### B. Littoral Waters Scenario

For testing in littoral waters with GPS outages due to dense vegetation, the mapping system of the Hector UGV was mounted on a USV system (Fig. 4(c)). No modification to the USV system was necessary and the complete SLAM system was mounted and operational in less than 30 minutes. Logfiles of different narrow parts of Claytor Lake in Virginia, USA, were recorded and used to learn maps of the enviroment without GPS data as well as with no specific motion model of the vehicle, as this would have required tighter integration and a longer preparation time. As can be seen in Fig. 5, our approach produces maps sufficiently accurate for navigation in case of GPS outages. As it does not use extracted features like polylines, it works well in the highly unstructured environment provided by coastal vegetation. It should be noted that the used Hokuyo UTM-30LX LIDAR does not return valid distance measurements when the beams hit water, which is advantageous for preventing faulty measurements from influencing the SLAM system. Once GPS is reacquired, a Graph-based SLAM approach can be used to optimize the pose graph provided by scan matching by adding GPS poses acquired before and after GPS outage as additional absolute position constraints. This is subject of future work.

### C. Embedded mapping system

An embedded mapping system is shown in Fig. 4(b). It consists of a Hokuyo UTM-30LX LIDAR system, an Intel Atom Z530 based CPU board as well as a small low-cost MEMS IMU. The system thus provides all necessary sensors

[1] http://www.youtube.com/playlist?list=PL0E462904E5D35E29



(a)

(b)

Fig. 6. Maps learned using the handheld mapping system overlaid with ground truth data: (a): RoboCup 2011 Rescue Arena with multiple small loops.(b) New building at Schloss Dagstuhl exhibiting a large loop.

for the application of the presented SLAM approach. It can easily be mounted on unmanned vehicles or carried by hand to learn maps of the environment. Fig. 6(a) shows a map learned by walking through the RoboCup 2011 Rescue Arena with the system in hand overlaid over a ground truth map and Fig. 6(b) shows the system used for mapping the new building at Dagstuhl castle, Germany. As it is visible from paths and maps, the system is sufficiently accurate to close loops typically encountered in small scale scenarios without use of an approach for explicit loop closure, keeping computational requirements low and preventing changes to the estimated map during runtime. Videos and ROS bagfiles of the experiments are available online[2]. The logged sensor data can be played back to the SLAM system at 3x real-time speed on the Atom Z530 CPU without loss of map quality. The SLAM system thus consumes less than half of overall computational resources of the embedded mapping system with the used settings.

## VII. Conclusions

A flexible and scalable SLAM approach usable for a multitude of scenarios is presented in this paper. We demonstrate the applicability for diverse scenarios such as simulated urban search and rescue, littoral mapping on a USV as well as in a hand-held embedded mapping system. We show that the system correctly estimates and incorporates the 3D motion of the used LIDAR system while consuming low computational resources. The system is available as open source software[3].

To the authors' knowledge the combination of high update rate simultaneous onboard 2D mapping and 6DOF pose estimation for low power platforms is otherwise not available as open source software.

## Acknowledgments

This work has been funded by the Research Training Group 1362 "Cooperative, Adaptive and Responsive Monitoring in Mixed-Mode Environments" of the German Research Foundation (DFG). We thank Dan Stilwell and Aditya Gadre, Bradley Department of Electrical and Computer Engineering at Virginia Tech, for giving us the opportunity to perform the USV experiment on Claytor Lake as well as Dr. Christian Lindig for providing the Dagstuhl floor plan.

## References

[1] M. Quigley, K. Conley, B. P. Gerkey, J. Faust, T. Foote, J. Leibs, R. Wheeler, and A. Y. Ng, "ROS: an open-source Robot Operating System," in *ICRA Workshop on Open Source Software*, 2009.

[2] J. Meyer, P. Schnitzspan, S. Kohlbrecher, K. Petersen, O. Schwahn, M. Andriluka, U. Klingauf, S. Roth, B. Schiele, and O. von Stryk, "A Semantic World Model for Urban Search and Rescue based on Heterogeneous Sensors," in *RoboCup 2010: Robot Soccer World Cup XIV*, 2011, pp. 180–193.

[3] G. Grisetti, C. Stachniss, and W. Burgard, "Improved techniques for grid mapping with rao-blackwellized particle filters," *IEEE Transactions on Robotics*, vol. 23, no. 1, pp. 34–46, 2007.

[4] M. Kaess, A. Ranganathan, and F. Dellaert, "iSAM: Incremental Smoothing and Mapping," *IEEE Transactions on Robotics*, vol. 24, no. 6, pp. 1365–1378, 2008.

[5] K. Konolige, G. Grisetti, R. Kummerle, W. Burgard, B. Limketkai, and R. Vincent, "Sparse Pose Adjustment for 2D Mapping," in *IEEE/RSJ Intern. Conf. on Intelligent robots and systems (IROS)*, Oct 2010.

[6] A. Bachrach, R. He, and N. Roy, "Autonomous flight in unknown indoor environments," *International Journal of Micro Air Vehicles*, vol. 1, no. 4, pp. 217–228, 2009.

[7] S. Grzonka, G. Grisetti, and W. Burgard, "Towards a Navigation System for Autonomous Indoor Flying," in *IEEE Intern. Conf. on Robotics and Automation (ICRA)*, 2009.

[8] I. Dryanovski, W. Morris, and J. Xiao, "An open-source pose estimation system for micro-air vehicles," in *IEEE Intern. Conf. on Robotics and Automation (ICRA)*, 2011, pp. 4449–4454.

[9] A. Kleiner and C. Dornhege, "Operator-Assistive Mapping in Harsh Environments," in *IEEE Int. Workshop on Safety, Security and Rescue Robotics (SSRR)*, 2009.

[10] D. Holz and S. Behnke, "Sancta simplicitas - on the efficiency and achievable results of SLAM using ICP-based incremental registration," in *IEEE Intern. Conf. on Robotics and Automation (ICRA)*, 2010, pp. 1380–1387.

[11] Z. Zhang, "Iterative point matching for registration of free-form curves and surfaces," *Int. J. Comput. Vision*, vol. 13, pp. 119–152, Oct 1994.

[12] A. Diosi and L. Kleeman, "Fast Laser Scan Matching using Polar Coordinates," *The International Journal of Robotics Research*, vol. 26, no. 10, p. 1125, 2007.

[13] E. Olson, "Real-Time Correlative Scan Matching," in *IEEE Intern. Conf. on Robotics and Automation (ICRA)*, Jun 2009, pp. 4387–4393.

[14] P. Biber and W. Straßer, "The Normal Distributions Transform: A New Approach to Laser Scan Matching," in *IEEE/RJS Intern. Conf. on Intelligent Robots and Systems*, 2003.

[15] R. Halterman and M. Bruch, "Velodyne HDL-64E lidar for unmanned surface vehicle obstacle detection," in *SPIE Conference*, Apr 2010.

[16] J. Kuipers, *Quaternions and rotation sequences*. Princeton University Press, 1999.

[17] D. Titterton and J. Weston, *Strapdown inertial navigation technology*. Peter Peregrinus Ltd, 2004.

[18] S. Thrun, W. Burgard, and D. Fox, *Probabilistic robotics*. MIT Press, 2008.

[19] B. D. Lucas and T. Kanade, "An iterative image registration technique with an application to stereo vision (darpa)," in *DARPA Image Understanding Workshop*, Apr 1981, pp. 121–130.

[20] E. A. Wan and R. V. D. Merwe, "The Unscented Kalman Filter for Nonlinear Estimation," in *Adaptive Systems for Signal Processing, Communications, and Control Symposium*. IEEE, 2000, pp. 153–158.

[21] A. Burguera, Y. González, and G. Oliver, "On the use of likelihood fields to perform sonar scan matching localization," *Autonomous Robots*, vol. 26, pp. 203–222, May 2009.

[22] M. Habbecke and L. Kobbelt, "Iterative Multi-View Plane Fitting," in *Vision, modeling, and visualization*, 2006, pp. 73–80.

[23] J. Meyer and A. Strobel, "A Flexible Real-Time Control System for Autonomous Vehicles," in *41st International Symposium on Robotics (ISR) and the 6th German Conference on Robotics (ROBOTIK)*, Jun 2010, pp. 860–865.

[24] S. Julier and J. Uhlmann, "Using covariance intersection for SLAM," *Robotics and Autonomous Systems*, vol. 55, no. 1, pp. 3–20, 2007.

[2] http://www.gkmm.tu-darmstadt.de/rescue

[3] http://code.google.com/p/tu-darmstadt-ros-pkg/