Hector Open Source Modules for Autonomous Mapping and Navigation with Rescue Robots

Stefan Kohlbrecher[1], Johannes Meyer[2], Thorsten Graber [2], Karen Petersen[1], Uwe Klingauf[2], and Oskar von Stryk[1]

[1]Department of Computer Science, TU Darmstadt, Germany
[2]Department of Mechanical Engineering, TU Darmstadt, Germany
http://www.gkmm.tu-darmstadt.de/rescue

Abstract. Key abilities for robots deployed in urban search and rescue tasks include autonomous exploration of disaster sites and recognition of victims and other objects of interest. In this paper, we present related open source software modules for the development of such complex capabilities which include *hector_slam* for self-localization and mapping in a degraded urban environment. All modules have been successfully applied and tested originally in the RoboCup Rescue competition. Up to now they have already been re-used and adopted by numerous international research groups for a wide variety of tasks. Recently, they have also become part of the basis of a broader initiative for key open source software modules for urban search and rescue robots.

1 Introduction

While robots used for Urban Search and Rescue (USAR) tasks will remain mainly tele-operated for the immediate future when used in real disaster sites, increasing the autonomy level is an important area of research that has the potential to vastly improve the capabilites of robots used for disaster response in the future.

The RoboCup Rescue project aims at advancing research towards more capable rescue robots [1]. Rescue robotics incorporates a vast range of capabilities needed to address the challenges involved, e.g. resulting from a degraded environment. The availability of re-useable and adaptable open source software can significantly reduce development time and increase robot capabilities while simultaneously freeing resources and, thus, accelerating progress in the field.

In this paper, we present open source modules that provide the building blocks for a system capable of autonomous exploration in USAR environments. Different modules have been applied with great success in RoboCup Rescue and other applications, both by Team Hector (Heterogeneous Cooperating Team of Robots) of TU Darmstadt and numerous other international research groups.

Robot Operating System (ROS) [2] is used as the robot middleware for the software modules. It has been widely adopted in robotics research and can be considered a de-facto standard. The provided modules have also become part of a recently established, broader initiative of the RoboCup Rescue community for providing standard software modules useful for USAR tasks [3].

At the RoboCup competition, we mainly use the Ackermann-steered Hector UGV vehicle (Figure 1)[4]. While this method is in many ways more challenging than differential steering, we do not focus on these challenges in this paper, instead providing a simulated skid-steered vehicle based on the Hector Lightweight UGV (Figure 1) that bears more similarity to differential drive vehicles commonly used for USAR tasks.

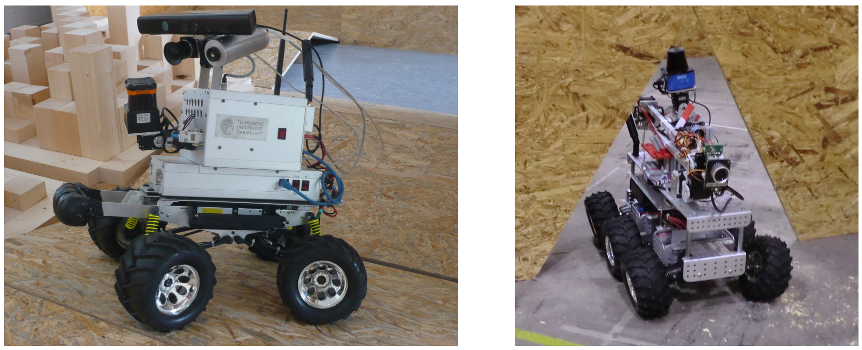

Fig. 1. Robots used by Team Hector. Left: Hector UGV based on Kyosho Twin Force chassis. Right: Hector Lightweight UGV based on "Wild Thumper" robot kit.

1.1 Related Work

Research in Simultaneous Localization and Mapping (SLAM) and exploration of unknown environments received a lot of attention in recent years, with impressive results being demonstrated. Many of these results often cannot be reproduced due several reasons, like a lack of standardized interfaces, closed source software and limited robustness to different (e.g. environmental) conditions.

Evaluation of state-of-the-art visual SLAM approaches [5], [6] in the standardized RoboCup Rescue setting showed promising results, but consistent localization/mapping as with the system described in this paper could not be achieved so far, as ramps and other obstacles lead to jerky vehicle motion and pose significant challenges to any SLAM system.

The RoboCup Rescue Robot League competition provides especially challenging scenarios, as the competition setting enforces strict constraints on the time and environment for robot operation.

2 System Overview

This paper covers many of the higher-level nodes originally developed and tested for the Hector UGV system, which can be used and adapted for other platforms without or with only slight modifications (Fig. 2). Hardware dependent modules

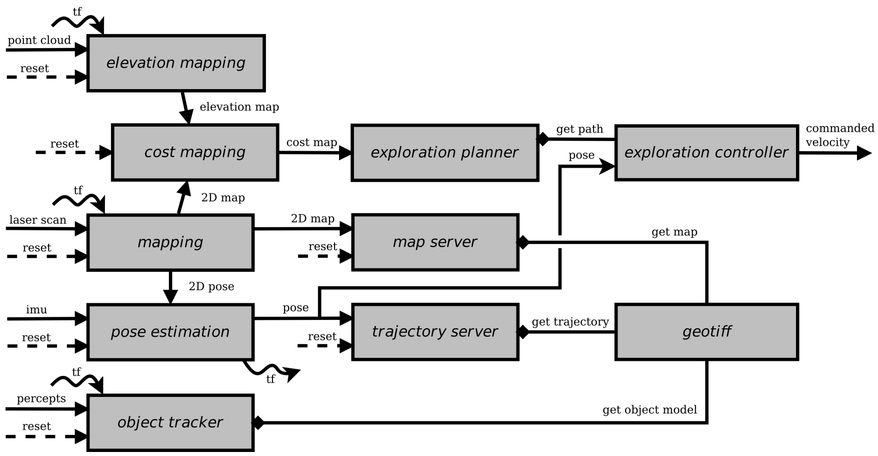

Fig. 2. System overview schematic. ROS nodes are represented by rectangles, topics by arrow-headed and services by diamond-headed lines. Services are originated at the service caller.

like camera and motor drivers or low-level controllers are not within the scope of this work. It is assumed that robots intended to use the described modules provide the necessary sensor data according to existing ROS standards and are steerable by publishing velocity commands. All nodes holding some sort of state information are subscribing the command topic which is primarily used to reset the system whenever necessary.

The following sections describe the ROS nodes provided[1]. Section 3 presents the open source software for 2D and 3D mapping, perception of objects of interests and the generation of GeoTIFF maps to visualize the relevant information according to the RoboCup Rescue rules. The subsequent Section 4 introduces the modules required for planning and autonomous exploration. While not directly related to autonomous robots being able to test individual modules and the robots overall behavior in simulation in a close-to-reality scenario is crucial in order to detect bugs or possible failure cases earlier and allows shorter development cycles. We present our simulation environment in Section 5.

3 Localization and Mapping

Creating maps of the environment is important for two reasons: Allowing first responders to both perform situation assessment and localize themselves inside buildings and for path planning and high level autonomous behaviors of robot systems.

While purely geometric maps such as occupancy grid maps are useful for navigation and obstacle avoidance, additional semantic information like the lo-

[1] for details see `http://www.ros.org/wiki/tu-darmstadt-ros-pkg`

cation of objects of interest is very important for first responders and required for intelligent high level autonomous behavior control.

3.1 Simultaneous Localization and Mapping (SLAM)

As disasters can significantly alter the environment compared to a pre-disaster state, USAR robots have to be considered as operating in unknown environments as to be most robust against changes. This means the SLAM problem has to be solved to generate sufficiently accurate metric maps useful for navigation of first responders or a robot system.

For this task we provide *hector_slam*, consisting of *hector_mapping*, *hector_map_server*, *hector_geotiff* and *hector_trajectory_server* modules. As odometry is notoriously unreliable in USAR scenarios, the system is designed to not require odometry data, instead purely relying on fast LIDAR data scan-matching at full LIDAR update rate. Combined with an attitude estimation system and an optional pitch/roll unit to stabilize the laser scanner, the system can provide environment maps even if the ground is non-flat as encountered in the RoboCup Rescue arena. A comprehensive discussion of *hector_slam* is available in [7].

3.2 Pose Estimation

The estimation of the full 6 degrees of freedom robot pose and twist is realized in the *hector_pose_estimation* node that implements an Extended Kalman Filter (EKF) and fuses measurements from an inertial measurement unit (IMU), the 2D pose error from the laser scan matcher and optionally from additional localization sensors like satellite navigation receivers, magnetometers and barometric pressure sensors if available. The filter is based on a generic motion model for ground vehicles and is primarily driven by the IMU, without using the control inputs or wheel odometry as they typically are unreliable due to wheelspin or side drift on uneven or slippery ground.

3.3 Elevation and Cost Mapping

In addition to a two-dimensional world representation obtained by the *hector_slam* package, USAR robots have to take the traversability of the environment into account. To this end we developed *hector_elevation_mapping*. This package fuses point cloud measurements obtained by a RGBD-camera such as the Microsoft Kinect into an elevation map. The elevation map is represented by a 2D grid map storing a height value with a corresponding variance for each cell. The cell measurement update is based on a local Kalman Filter and adapted from the approach described in [8].

Finally, *hector_costmap* fuses the 2.5D elevation map with the 2D occupancy grid map provided by *hector_mapping* and computes a two-dimensional cost map for the exploration task.

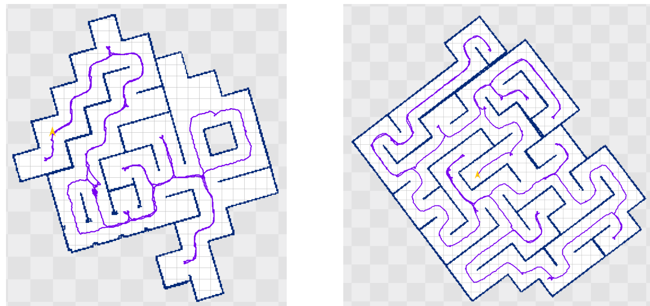

Fig. 3. Examples for autonomous exploration. Left: Simulated Thailand Rescue Robot Championship 2012 arena. Right: Simulated random maze.

3.4 Objects of Interest

Plain occupancy grid maps provide information about the environment geometry, but do not contain semantic information. We track information about objects of interest in a separate module, using a Gaussian representation for their position. The *hector_object_tracker* package is based on an approach described comprehensively in [9]. It subscribes to percept messages from victim, QR code or other object detectors, projects them to the map frame based on the robot's pose, camera view angle and calibration information and solves the association and tracking problem for subsequent detections.

3.5 GeoTIFF Maps

To achieve comparability between environment maps generated by different approaches, the GeoTIFF format is used as standard map format in the RoboCup Rescue League competition. Using geo-reference and scale information, maps can be overlaid over each other using existing tools and accuracy can be compared. The *hector_geotiff* package allows generating RoboCup Rescue rules compliant GeoTIFF maps which can be annotated through a plugin interface. Plugins for adding the path travelled by the robot, victim and QR code locations are provided. The node can run onboard a robot system and save maps to permanent storage based on a timer, reducing the likelihood of map loss in case of connectivity problems. All map shown in figures in this paper have been generated using the *hector_geotiff* node.

4 Planning and Exploration

While a plethora of research results are available for exploration using autonomous robots, there are very few methods readily available for re-use as open source software. We provide the *hector_exploration_planner* that is based on the exploration transform approach presented in [10]. In our exploration planner, frontiers towards the front of the robot are weighted favorably, to prevent frequent costly turning of the robot. Inspired by wall following techniques used by firefighters [11], a "follow wall" trajectory can also be generated using the exploration planner. The planned trajectory is generated based on map data and thus does not exhibit weaknesses associated with reactive approaches that only consider raw sensor data [12]. High level behaviors can thus switch between using the exploration transform and wall follow approach at any time. In case the environment has been completely explored, the planner has been extended to start an "inner exploration" mode. Here, the traversed path of the robot containing a discrete set of past robot poses is retrieved from the *hector_trajectory_server* node. These positions are sampled based on distance from each other and added to a list. This list is passed to the exploration transform algorithm as a list of goal points. An exhaustive search for the exploration transform cell with the highest value then yields a point that is farthest away from the previous path and safe to reach for the robot.

5 Simulation

Experiments using real robots are time-consuming and costly as availability of appropriate scenarios and wear and tear of robot systems have to be considered. This holds especially for USAR environments (like the RoboCup Rescue arena) as those also put high strain on robot hardware and lab space is often limited.

5.1 Environments

To conveniently be able to create simulated environments for experiments, the *hector_nist_arenas_gazebo* stack provides the necessary tools that allow the creation of scenarios by composition of provided NIST standard test arena elements. Users can also easily add further elements. The *hector_nist_arena_worlds* package provides example arenas, including both models for the RoboCup German Open 2011 and the Thailand Robot Championship 2012 Rescue arenas. Gazebo does not support multispectral sensor simulation originally. To enable simulation of thermal images often used for the detection of victims that emit body heat, the *hector_gazebo_thermal_camera* package provides a gazebo camera plugin that can be used for this task.

5.2 Ground Vehicles

The *hlugv_gazebo* package provides a model of the Hector Lightweight UGV system (Fig. 1 left). The robot uses differential drive for its six wheels and thus behaves similar to tracked robot systems commonly applied in USAR scenarios.

6 Application and Impact

6.1 RoboCup

Within less than two years *hector_slam* has become the de-facto standard SLAM system used by many teams with great success in RoboCup competitions. With

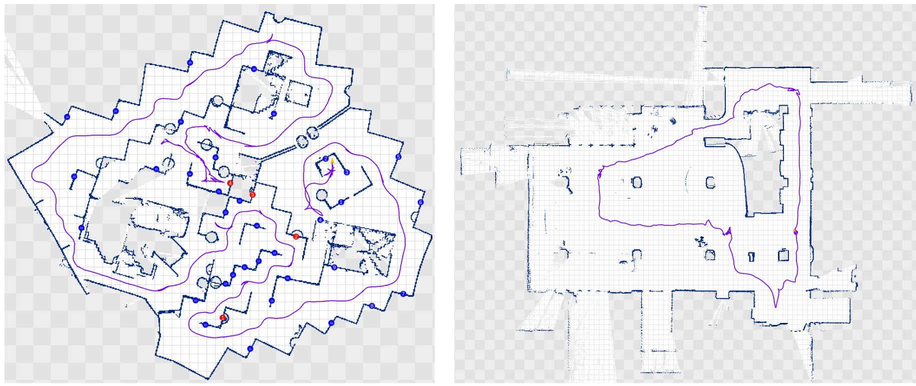

Fig. 4. Maps learned using the provided Hector modules. Left: Map learned using *hector_slam* and *hector_exploration_planner* at the RoboCup 2012 final mission with the Hector UGV robot. The robot started at the right middle position and autonomously explored the majority of the arena, finding 3 victims (red markers). The fourth victim was found using tele-operation for the last 4 meters of the travelled path. Blue markers indicate the positions of 35 QR codes that were detected autonomously by the robot. Right: Application of *hector_slam* to the ccny quadrotor lobby dataset [13].

Team Hector winning the Best in Class Autonomy award both at the RoboCup German Open 2012 and RoboCup Mexico 2012 and Team BARTlab winning the award at the Thai Rescue Robot Championship 2012, the applicability and adaptability of the system to challenging environments and different robot platforms has clearly been demonstrated. Fig. 4 left shows a real-world map learned using the presented modules with the Hector UGV system.

6.2 Other Applications

Hector open source modules have been re-used for both research and commercial purposes[2]. *hector_mapping* was succesfully deployed in different applications such as mapping of littoral areas using a unmanned surface vehicle, mapping different environments using a handheld mapping system and building radio maps for wireless sensor networks [14]. Fig. 4 right shows results when applied to the quadrotor datasets provided in [13]. The resulting map is consistent and comparable to the results in the original paper, showing the flexibility of the system.

7 Conclusion

A collection of open source modules has been presented for providing urban search and rescue robots with abilities like mapping and exploration of disaster sites and tracking of objects of interest. Many of the presented modules have already been adopted by other research groups for RoboCup Rescue and beyond.

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

Acknowledgments. This work was supported by the DFG Research Training Group 1362. We thank contributing past and present team members, notably Florian Berz, Florian Kunz, Mark Sollweck, Johannes Simon, Georg Stoll and Laura Strickland.

References

1. Jacoff, A., Sheh, R., Virts, A.M., Kimura, T., Pellenz, J., Schwertfeger, S., Suthakorn, J.: Using competitions to advance the development of standard test methods for response robots. In: Proc. Workshop on Performance Metrics for Intelligent Systems. PerMIS '12, New York, NY, USA, ACM (2012) 182–189
2. Quigley, M., Gerkey, B., Conley, K., Faust, J., Foote, T., Leibs, J., Berger, E., Wheeler, R., Ng, A.: ROS: an open-source Robot Operating System. In: ICRA workshop on open source software. Volume 3. (2009)
3. Kohlbrecher, S., Petersen, K., Steinbauer, G., Maurer, J., Lepej, P., Uran, S., Ventura, R., Dornhege, C., Hertle, A., Sheh, R., , Pellenz, J.: Community-Driven Development of Standard Software Modules for Search and Rescue Robots. In: IEEE Intern. Symposium on Safety, Security and Rescue Robotics (SSRR). (2012)
4. Graber, T., Kohlbrecher, S., Meyer, J., Petersen, K., von Stryk, O., Klingauf, U.: RoboCupRescue 2013 - Robot League Team Hector Darmstadt (Germany). Technical report, Technische Universität Darmstadt (2013)
5. Geiger, A., Ziegler, J., Stiller, C.: StereoScan: Dense 3d Reconstruction in Real-time. In: IEEE Intelligent Vehicles Symposium, Baden-Baden, Germany (2011)
6. Huang, A.S., Bachrach, A., Henry, P., Krainin, M., Maturana, D., Fox, D., Roy, N.: Visual odometry and mapping for autonomous flight using an RGB-D camera. In: International Symposium on Robotics Research (ISRR). (2011)
7. Kohlbrecher, S., Meyer, J., von Stryk, O., Klingauf, U.: A Flexible and Scalable SLAM System with Full 3D Motion Estimation. In: IEEE International Symposium on Safety, Security and Rescue Robotics. (2011)
8. Kleiner, A., Dornhege, C.: Real-time Localization and Elevation Mapping within Urban Search and Rescue Scenarios. Journal of Field Robotics (8-9) (2007) 723–745
9. Meyer, J., Schnitzspan, P., Kohlbrecher, S., Petersen, K., Schwahn, O., Andriluka, M., Klingauf, U., Roth, S., Schiele, B., von Stryk, O.: A Semantic World Model for Urban Search and Rescue Based on Heterogeneous Sensors. In: RoboCup 2010: Robot Soccer World Cup XIV. Lecture Notes in Computer Science (2011) 180–193
10. Wirth, S., Pellenz, J.: Exploration transform: A stable exploring algorithm for robots in rescue environments. In: IEEE International Workshop on Safety, Security and Rescue Robotics (SSRR). (2007) 1–5
11. International Association of Fire Chiefs and National Fire Protection Association: Fundamentals of fire fighter skills. Jones & Bartlett Learning (2008)
12. Van Turennout, P., Honderd, G., Van Schelven, L.: Wall-following control of a mobile robot. In: IEEE International Conference on Robotics and Automation (ICRA). (1992) 280–285
13. Dryanovski, I., Morris, W., Xiao, J.: An open-source pose estimation system for micro-air vehicles. In: IEEE International Conference on Robotics and Automation (ICRA). (2011) 4449–4454
14. Scholl, P.M., Kohlbrecher, S., Sachidananda, V., van Laerhoven, K.: Fast Indoor Radio-Map Building for RSSI-based Localization Systems. In: Demo Paper, International Conference on Networked Sensing Systems. (2012)