

Towards Highly Reliable Autonomy for Urban Search and Rescue Robots

Stefan Kohlbrecher^(✉), Florian Kunz, Dorothea Koert, Christian Rose,
Paul Manns, Kevin Daun, Johannes Schubert, Alexander Stumpf,
and Oskar von Stryk

Department of Computer Science, Technische Universität Darmstadt,
Karolinenplatz 5, 64289 Darmstadt, Germany
rescue@sim.tu-darmstadt.de
<http://www.gkmm.tu-darmstadt.de/rescue>

Abstract. Participating in the RoboCup Rescue Real Robot League competition for approximately 5 years, the members of Team Hector Darmstadt have always focused on robot autonomy for Urban Search and Rescue (USAR). In 2014, the team won the RoboCup RRL competition. This marked the first time a team with a strong focus on autonomy won the championship. This paper describes both the underlying research and open source developments that made this success possible as well as ongoing work focussed on increasing rescue robot performance.

The exploration of disaster environments poses a great challenge to both human rescuers and rescue robot systems (as well as canines). The RoboCup Rescue Robot League (RRL) competition benchmarks robot system performance in scenarios that are designed to represent many of the same challenges as real Urban Search and Rescue (USAR) situations in a systematic manner to enable reproducible benchmarking of robot capabilities.

In this work we present recent advances in software for USAR robots, provide an overview of performance at the RoboCup 2014 competition and present ongoing work towards further improved robot capabilities.

We use the open source Robot Operating System (ROS) as a middleware for our software. In the following sections, ROS package or stack names written in *italics* like *hector_slam* are available as open source software and can be found on the ROS wiki, e.g., www.ros.org/wiki/hector_slam.

The remainder of the paper is structured as follows. The next section presents related work on USAR robotics. It outlines the modularity concept of the hardware and stresses the importance of simulation capabilities for robots in the USAR context. The two subsequent sections show recent developments in navigation and perception software including victim search, navigation towards goal poses, path smoothing and victim localization. Afterwards, we assess the performance of the robots with respect to the benchmark that RoboCup Rescue League provides. Then, we give a short overview on ongoing work towards 3D mapping, planning and active gaze control. We close with a conclusion.

1 Related Work

While capabilities for navigation or victim search using unmanned ground vehicles have been research topics for a long time, only few publications consider fully integrated autonomous UGV systems such as the one described in this work. For this reason, we focus on multiple aspects and modules that, in combination, enable autonomous operation in challenging environments. In previous work we provide a comprehensive description of components for world modeling (*hector_worldmodel*) [1] and SLAM (*hector_slam*) [2]. Our approach and open source software for navigation is described in [3]. Further detailed information about used hardware is available in the team description papers (TDPs) published yearly [4]. Multiple members of the team are also of member of the DARPA Robotics Challenge Team ViGIR. We are currently adapting their research in manipulation and 3D motion planning [5] for use with autonomous wheeled and tracked ground robots.

1.1 Hardware

We use two different types of mobility platforms for research in autonomous systems for USAR tasks. The Hector UGV (Fig. 1a) is a lightweight agile ground vehicle. The platform is based on a modified Kyosho Twin Force R/C model. The 4-wheel-drive with a differential gear for each axis allows the vehicle to move on slippery surfaces. The front and rear wheels can be steered independently, providing higher mobility than common 2-wheel-steering.

A major drawback of the Hector UGV platform is its limited mobility in rough terrain. To address this shortcoming, we introduced the tracked “Obelix” UGV platform, shown in Fig. 1b. This tracked robot with front and back flippers provides state of the art mobility, similar to other (response) robots used in challenging terrain.

A key feature of both platforms is their modularity. The main components are the mobility platform and the “autonomy box” featuring various sensors for navigation and victim detection as well as a state of the art (Intel Core i7) computer system. The “autonomy box” can be mounted on any mobile platform with minor modifications to high-level software. Only low-level interfaces (e.g. motion controllers) of the mobility platform have to be adapted to communicate with the ROS-based high-level software.

1.2 Simulation

To achieve high reliability of complex autonomous systems, the capability to perform comprehensive testing of the complete system is highly important. Setting up a real robot for such testing is both time and effort consuming, especially when (for instance large) scenarios are to be considered that cannot be provided physically. We simulate the complete system, including actuation, sensors and scenarios using *Gazebo*¹. This enables testing of autonomous behaviors with

¹ <http://gazebosim.org/>.

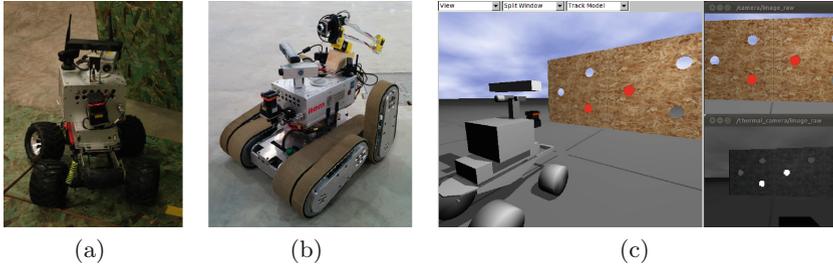


Fig. 1. Team Hector mobile platforms: (a) Hector UGV (b) Tracked vehicle Obelix (c) Hector UGV in Gazebo with simulation of thermal sensor data visualized.

negligible setup effort in multiple scenarios without the need for real hardware. As an example the simulation of a thermal camera can be seen in Fig. 1c. Simulated RoboCup Rescue arena scenarios serve to evaluate the robot in the USAR context. The *hector_nist_arena_designer* ROS package permits the fast and intuitive creation of user defined test scenarios.

2 Navigation

This chapter describes on software modules developed for autonomous navigation. We focus on recent research and refer the reader to prior publications for information about our approaches for SLAM [2] and navigation [3].

2.1 Exploration Planner

Based on the exploration transform approach [6] commonly used for frontier-based, risk-aware exploration, we developed multiple modifications in the *hector_exploration_planner* for improved performance in search and rescue missions.

Improved Victim Search. To detect objects of interest in disaster scenarios reliably, a thorough search has to be performed by (autonomous) robot systems tasked with the exploration of the environment. Exploration of unknown environments has been intensively studied in the past, but only few approaches consider different sensing modalities. Often, the environment that has been covered by a LIDAR sensor is considered explored. This approach is sufficient for tasks like mapping an unknown environment, but it does not take the limited range of sensors into account when searching for objects of interest like victims in a disaster environment. The reliable detection of victims trapped under rubble using thermal imaging requires the USAR robot system to come as close as possible to enable detection even if only very small portions of the victim are visible to sensors. To provide this capability, we modified the original exploration transform approach. In our modified version, we do not plan paths to map frontiers (grid cells that have a transition from known free space to unknown space),

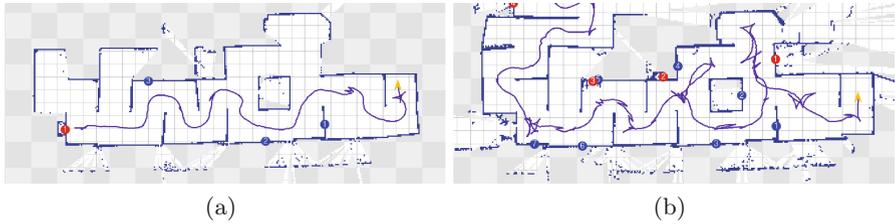


Fig. 2. Victim search at RoboCup 2014: (a) Use of standard exploration transform. (b) Exploration by approaching to target points close to the path.

but limit the generation of target poses to those that have a specified distance from the path that the robot has traveled so far. Using this approach, the system also explores tight spaces that would otherwise be ignored by LIDAR map based exploration. Figure 2 shows examples of standard exploration and our new approach.

Robust Navigation Towards Goal Poses. Both in autonomous and semi-autonomous operation, target poses for the robot system might be invalid due to obstacles preventing the robot from reaching it. Without handling of this, the planner will simply fail planning to such poses. In many practical cases, however, a pose close-by may be reachable, allowing continuation of the robot mission without interruption.

A common example we observed in the USAR scenario is approaching victims for inspection. To reach victims that are situated inside or close-by to obstacles, just forwarding their pose P_V to the planner as goal poses would naturally fail. Instead, a *observation pose* P_o has to be found that is sufficiently close to the victim to provide sensors with a view that allows to inspect and possibly confirm the victim hypothesis.

To achieve this task, the planner is augmented with an approach for generating valid observation poses. Based on an obstacle transform map, an area around the target pose is searched for collision free poses as shown in Fig. 3. For all candidates, the vector from the original target to the candidate is checked against the original target orientation. The candidate is only used if the orientation difference lies within a predefined threshold. This is motivated by the fact that the observation of the target pose is only possible if it lies in front of the robot.

2.2 Path Smoothing

Based on a discrete map representation, the exploration planner generates continuous piecewise linear paths. Two successive segments can be right- or even acute-angled with respect to each other. Such non-smooth paths cannot be reliably followed by motion controllers, so a smoothing approach is employed.

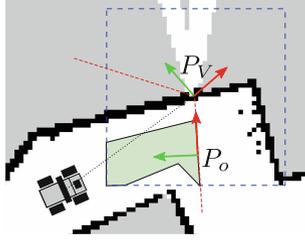


Fig. 3. Generating observation poses: P_V is the estimated victim pose. Constraints based on victim orientation (dashed red lines), distance from original pose (dashed blue lines) and distance from walls result in the mint green polygon being searched and P_o getting selected (Colour figure online).

The piecewise linear path is passed as the set of corner points. We introduce a variable t , which represents the integrated length along the path, and obtain a functional representation γ of the path as

$$\gamma(t) = \sum_{i=0}^{N-1} \chi_{[t_i, t_{i+1})}(t) \left(\mathbf{x}_i + \frac{t - t_i}{t_{i+1} - t_i} (\mathbf{x}_{i+1} - \mathbf{x}_i) \right) \quad (1)$$

where $\mathbf{x}_i \in \mathbb{R}^3$ denotes the i -th one of the $N + 1$ corner points, t_i the integrated Euclidean length until \mathbf{x}_i and χ_A the characteristic function for $A \subseteq \mathbb{R}$. We consider $\gamma(t)$ and \mathbf{x}_i in \mathbb{R}^3 , but the approach works in \mathbb{R}^2 as well.

This approach allows us to treat the path smoothing problem as a function smoothing problem. It is tackled by sampling γ with an intermediate discretization $D_M = \{\tilde{t}_0, \dots, \tilde{t}_M\}$ and computing a Gaussian-weighted convex combination from the points $\gamma(\tilde{t}_i)$. The resulting smooth function $\tilde{\gamma}$ is given by

$$\tilde{\gamma}(t) = \sum_{m=0}^M \gamma(\tilde{t}_m) \frac{\exp\left(-\frac{(t - \tilde{t}_m)^2}{2\eta^2}\right)}{\sum_{m=0}^M \exp\left(-\frac{(t - \tilde{t}_m)^2}{2\eta^2}\right)} \quad (2)$$

The steepness of the Gaussian weight is controlled by the parameter η . It trades off accuracy versus smoothness of $\tilde{\gamma}$. After computing $\tilde{\gamma}$, it is sampled again, yielding another piecewise linear path $\hat{\gamma}$ which is fed into the base controller. Figure 4 shows an input path and its smoothed versions for different choices of η . One observes that a small value yields a path being very close to the original path while a high value implies big deviations and a centralization effect on the end points of $\hat{\gamma}$.

Compared to smooth function approximation approaches like Gaussian processes as described in [7] or curve smoothing approaches like B-splines as described in [8], our approach works more heuristically. However, it overcomes the need for solving optimization problems or linear equation systems for new input paths with complexities of $\mathcal{O}(n^2)$ or worse.

Complexity lies in $\mathcal{O}(n)$ which is beneficial given limited onboard computational resources and frequent path updates. Furthermore, the approach only

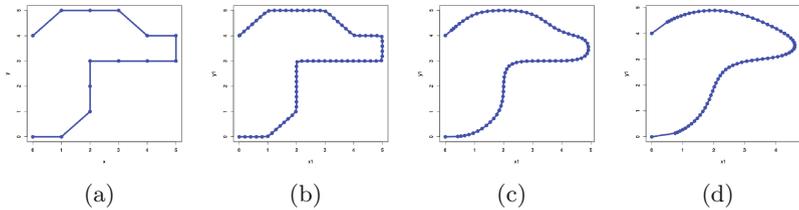


Fig. 4. Input path (a) and output path with different values for the temperature parameter η . (b) $\eta = 0.1$, (c) $\eta = 0.5$, (d) $\eta = 1.0$

requires to choose a single parameter, which is easy to handle compared to other techniques that have more than three degrees of freedom and suffer from numerical instabilities if the prediction is computed without $\mathcal{O}(n^3)$ techniques [9, 10]. B-splines suffer from a very complex definition and a difficulty to control the trade-off between smoothness and overfitting [11]. To choose η , we propose to reduce big directional differences along the path while staying in an ε -tube around the original path.

3 Perception

The semantic world model backend of our perception system is described in [1] and details are omitted here for brevity.

Victim Localization. Reliable localization of human victims in unstructured post-disaster environments is a key issue for USAR robots. We tackle this problem by using complementary information provided by imagery cues such as RGB, thermal and depth images. Possible victims are identified by analyzing thermal images with a blob detection algorithm with configurable thresholds. While providing important cues for victim locations, thermal images can yield a high number of false-positives. They can be caused by other heat emitting spots, such as radiators or fire. We minimize the number of false positives by using the complementary information provided by visual perception and depth information provided by an RGB-D camera. Using this 3D information, false-positive victims can be identified by the shape of the environment as well as by missing depth information at the victim's location.

To identify the correct position of a victim after its detection by the robot's sensors and insert it into the map, we cast a 3D ray along the robot's gaze direction into the onboard octomap [12]. This map is created from the depth data of the RGB-D camera. Limited octomap resolution and state estimation noise might result in erroneous ray casts. Without consideration of this issue, a estimated victim location might be mapped onto a corner instead of the correct spot behind it. This case is illustrated in Fig. 5 where the victim would be mapped on the corner at \mathbf{p}_0 instead of \mathbf{p}_V behind it. To overcome this issue, we perform a spread fan ray cast of n rays which hit the octomap on the right and

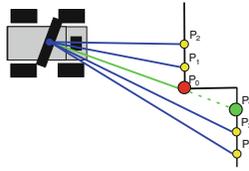


Fig. 5. Example of a victim at P_V mapped to a corner P_0 due to sensor noise. The additional rays (blue) allow detection of the false mapping (Colour figure online).

the left of the original gaze direction ray. The fan is scaled such that the points are equally spaced on a horizontal line around \mathbf{p}_0 . The ray casts return the wall points \mathbf{p}_i . For all of them, we compute the Euclidean distances d_i to the camera origin \mathbf{c} . If they differ significantly, we decide on a corner case and dismiss this object localization attempt. This provides the chance of a correct mapping when new sensor data arrive and the robot does not waste time during exploration of false-positives.

4 Results

The RoboCup Rescue competition represent a systematic and reproducible benchmark for testing and evaluation of teleoperated and autonomous USAR robots in a complex environment in direct competition with the approaches of other teams. We present results from both the RoboCup German Open 2014 and RoboCup 2014. In both competitions we won both the “Best in Class Autonomy” award as well as the overall competition.

Figure 7 shows the performance increase during the RoboCup 2014 competition with respect to the performance metrics *number of found victims* and *number of found QR-codes*. The number of found victims evaluates performance for the USAR task. The total number of victims was 12 in all missions. The number of found QR-codes correlates with the fraction of the arena that was explored by the robot during the mission. The performance increase is also visible when comparing maps generated in the preliminary round (Fig. 2) with those of the final missions at RoboCup 2014 (Fig. 6).

Intense training of the operator is required for the case that human control becomes desirable in difficult situations. As our robot allows flexible switching between full and semi-autonomous behavior during the mission, we leveraged this capabilities in multiple competition runs. In both RoboCup 2014 final missions, the robot successfully found 4 victims autonomously, after which the operator took over control to explore the rest of the arena. It should be noted that a operator mistake lead to the robot rolling over in the second final mission, ending the mission prematurely.

We provide combined external and operator control station video from multiple missions at the RoboCup German Open 2014 competition online².

² <http://www.youtube.com/playlist?list=PLqdOEBv9QGGrEiqlUklq0BIIQPU55IhTO2>.

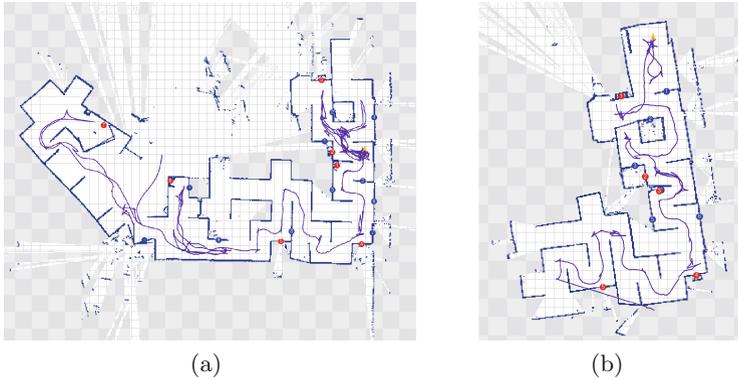


Fig. 6. RoboCup 2014 example maps: (a) Final mission 1 (b) Final mission 2. The starting pose of the robot is marked by yellow arrows. In both missions, the robot discovered 4 victims autonomously (right part of the arena) (Colour figure online).

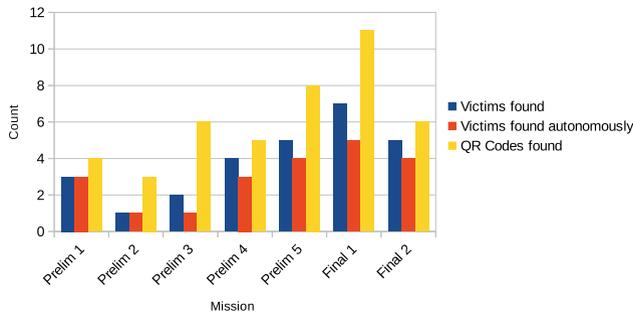


Fig. 7. Improvement of Team Hector's performance during the RoboCup 2014 competition.

5 Ongoing Work

Despite the good performance demonstrated in the previous section, there remain open issues that have to be tackled to increase robustness of the USAR robot system and allow deployment in real disasters in the future. In this section, directions and preliminary results of ongoing work are described.

5.1 Active Gaze Control

For victim detection, it is crucial to have reliable sensor coverage as to not miss any victims. It is crucial that the robot covers the scenario thoroughly in a minimal amount of time. Intelligent camera control thus can be a major factor for ensuring sufficient performance. Solutions for next best view point and coverage path planning on mobile robots must avoid high computational effort like computing all possible view points. We use a coverage mapping approach

based on the octomap framework [12]. We extend it to include the viewing angle α_n that measures the deviation of a view ray from a surface normal. We consider a frontal view of a surface as the most useful in a victim detection scenario. The score of a cell being sensed by a certain sensor $S(c_i)$ is initialized with zero for all cells and updated for every incoming sensor reading according to

$$S(c_i)_{new} = \begin{cases} S(c_i)_{old} + \gamma(1 - (\frac{\alpha_n}{\pi})^2) & \text{if } \alpha_n < \frac{\pi}{4} \\ S(c_i)_{old} + \gamma & \text{otherwise} \end{cases} \quad (3)$$

such that views near a frontal viewing angle result in a better coverages score than views under a steeper angle. Here, γ denotes the update factor and all cells having a $S(c_i)$ value higher than a certain threshold are considered as already been covered.

The information provided by this coverage map can be used to compute the next gaze direction for the robot. Given the current pan and tilt of the camera, a randomized set of n possible next gaze points is sampled. Each point is weighted by the effort to look at it and the number of uncovered points that can be found in the surroundings. The point with the highest weight is chosen as the next gaze direction for the robot.

Tests in simulation have shown several benefits of intelligent gaze control. The robot is able to adapt sensor motion to the environment and therefore does not miss unexplored areas which might happen when following a fixed sensor motion pattern.

We plan to extend this approach for future use and consider using the coverage map not only for the control of the sensor head but also for path planning similar to the approach proposed in [13].

5.2 3D Mapping

We have been using the previously mentioned highly robust *hector_slam* approach in the past. With a 2D internal map representation, the capability for full 3D state estimation from LIDAR data is limited. Thus, although this system provides great reliability and robustness in various scenarios, it provides limited capability for use in multi-story scenarios.

For this reason, we are currently evaluating 3D SLAM approaches allowing deployment of tracked vehicles in challenging terrain, including significant roll and pitch angles and angular rates of the platform.

The state of the art *loam_back_and_forth* [14] and *ethzasl_lic_mapper* [15] ROS packages were selected for initial tests in simulation. These tests confirmed that major challenges are the tight spaces of USAR scenarios, high (roll/pitch) velocities of the platform and sensor occlusion when close to obstacles. We will investigate tackling these challenges in future work.

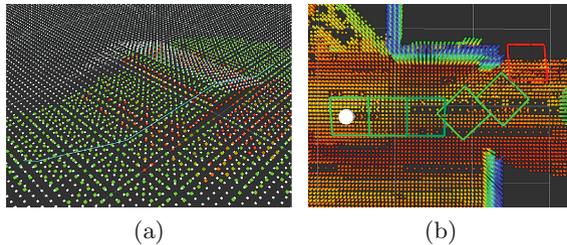


Fig. 8. Planning over a ramp (a) with low (green dots) and high (red dots) risk costs. The path (cyan) and the expanded states are projected onto the ground. (b) Checking a 2D path for stability, green poses stable, red pose instable (Colour figure online).

5.3 3D Path Planning

While reduction of 3D data to a 2D representation for planning can be computationally advantageous, it only allows avoiding harsh terrain, but not planning its traversal.

The problem of planning a path from the robot position to a goal point can be reduced to finding a path in a graph. We base our planner on the SBPL library [16]. The set of nodes is a lattice of pose states $s(x, y, \theta)$ and the edges are the transitions connecting them. The available transitions are a discretized subset of possible robot movements. The Anytime Repairing A* (ARA*) algorithm [17] finds an initial solution and incrementally improves it. Therefore, it is well suited for real-time applications like ours.

We propose the following cost function that consists of a basic cost $c_b(s, s')$ for the movement and a factor $f_r(s') \geq 1$ for the risk associated with reaching the target position:

$$c(s, s') = f_r(s')c_b(s, s'). \quad (4)$$

f_r is computed with the Force Angle Stability Metric (FA) [18], which is based on the supporting polygon, i.e. the convex hull of the ground contact points.

Each of its edges is considered a possible tip-over axis. FA is computed for them based on the gravity-induced moments. The value of the least stable axis serves as the stability estimate of the whole system.

To apply FA, the contact points need to be computed. To avoid the high computational cost associated with physics simulation, we developed an algorithm providing a sufficiently accurate estimation by imitating the physical behavior with geometric operations. It imitates dropping the robot onto the requested position and is based on a 3D octomap model generated from onboard sensors.

Evaluation shows that this planning approach enables the computation of safely traversable paths in 3D terrain (Fig. 8a). We also use FA for checking planned paths for their feasibility (Fig. 8b).

6 Conclusion

Autonomous capabilities for robots in USAR scenarios offer a huge potential of improvements for responders, but are not yet available in practice. In this paper, we describe the research approaches and technical achievements (including open source developments) underlying the autonomous USAR robots of Team Hector which in 2014 became the first team with main focus on autonomous robot capabilities to win the RoboCup Rescue Robot League championship.

Acknowledgements. This work has been supported in parts by the German Research Foundation (DFG) within the Research Training Group 1362 “Cooperative, Adaptive and Responsive Monitoring in Mixed-Mode Environments” of Technische Universität Darmstadt.

References

1. Meyer, J., et al.: A semantic world model for urban search and rescue based on heterogeneous sensors. In: Ruiz-del-Solar, J. (ed.) RoboCup 2010. LNCS, vol. 6556, pp. 180–193. Springer, Heidelberg (2010)
2. Kohlbrecher, S., Meyer, J., von Stryk, O., Klingauf, U.: A flexible and scalable SLAM system with full 3D motion estimation. In: Proceedings of IEEE International Symposium on Safety, Security and Rescue Robotics (SSRR), Kyoto, Japan, 1–5 November 2011, pp. 155–160. IEEE (2011)
3. Kohlbrecher, S., Meyer, J., Graber, T., Petersen, K., Klingauf, U., von Stryk, O.: Hector open source modules for autonomous mapping and navigation with rescue robots. In: Behnke, S., Veloso, M., Visser, A., Xiong, R. (eds.) RoboCup 2013. LNCS, vol. 8371, pp. 624–631. Springer, Heidelberg (2014)
4. Kohlbrecher, S., Meyer, J., Graber, T., Petersen, K., Klingauf, U., von Stryk, O.: RoboCupRescue 2014 - Robot League Team Hector Darmstadt (Germany). Technical report. Technische Universität Darmstadt (2014)
5. Kohlbrecher, S., Romay, A., Stumpf, A., Gupta, A., von Stryk, O., Bacim, F., Bowman, D.A., Goins, A., Balasubramanian, R., Conner, D.C.: Human-robot teaming for rescue missions: team ViGIR’s approach to the 2013 DARPA robotics challenge trials. *J. Field Robot.* **32**, 352–377 (2014)
6. 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), pp. 1–5 (2007)
7. Rasmussen, C.E.: Gaussian Processes for Machine Learning. MIT Press, Cambridge (2006)
8. Gallier, J.: Curves and Surfaces in Geometric Modeling: Theory and Algorithms. Morgan Kaufmann, San Francisco (1999)
9. Gärtner, T., Driessens, K., Ramon, J.: Graph kernels and gaussian processes for relational reinforcement learning. In: Horváth, T., Yamamoto, A. (eds.) ILP 2003. LNCS (LNAI), vol. 2835, pp. 146–163. Springer, Heidelberg (2003)
10. Foster, L., Waagen, A., Aijaz, N., Hurley, M., Luis, A., Rinsky, J., Satyavolu, C., Way, M.J., Gazis, P., Srivastava, A.: Stable and efficient gaussian process calculations. *J. Mach. Learn. Res.* **10**, 857–882 (2009)

11. Eilers, P.H.C., Marx, B.D.: Flexible smoothing with B-splines and penalties. *Stat. Sci.* **11**(2), 89–102 (1996)
12. Hornung, A., Wurm, K.M., Bennewitz, M., Stachniss, C., Burgard, W.: OctoMap: an efficient probabilistic 3D mapping framework based on octrees. *Auton. Robot.* **34**(3), 189–206 (2013). <http://octomap.github.com>
13. Dornhege, C., Kleiner, A., Kolling, A.: Coverage search in 3D. In: Proceedings of the Symposium on Safety, Security and Rescue Robotics (SSRR), pp. 1–8, October 2013
14. Zhang, J., Singh, S.: LOAM: lidar odometry and mapping in real-time. In: Robotics: Science and Systems Conference, July 2014
15. Pomerleau, F., Colas, F., Siegwart, R., Magnenat, S.: Comparing ICP variants on real-world data sets. *Auton. Robots* **34**(3), 133–148 (2013)
16. Likhachev, M.: Search-based planning with motion primitives (2009)
17. Likhachev, M., Gordon, G.J., Thrun, S.: ARA*: anytime A* with provable bounds on sub-optimality. In: Advances in Neural Information Processing Systems (2003)
18. Papadopoulos, E., Rey, D.A.: The force-angle measure of tipover stability margin for mobile manipulators. *Veh. Syst. Dyn.* **33**(1), 29–48 (2000)