RoboCupRescue 2015 - Robot League Team Hector Darmstadt (Germany)

Stefan Kohlbrecher[1], Johannes Meyer[2], Thorsten Graber[2], Karen Kurowski[1], Oskar von Stryk[1⋆]

Department of Computer Science (1) and Department of Mechanical Engineering (2),
Technische Universität Darmstadt,
Karolinenplatz 5, D-64289 Darmstadt, Germany
E-Mail: rescue@sim.tu-darmstadt.de
Web: www.gkmm.tu-darmstadt.de/rescue

Abstract. This paper describes the approach used by Team Hector Darmstadt for participation in the 2015 RoboCup Rescue Robot League competition. Participating in the RoboCup Rescue competition since 2009, the members of Team Hector Darmstadt focus on exploration of disaster sites using autonomous Unmanned Ground Vehicles (UGVs). The team has been established as part of a PhD program funded by the German Research Foundation at TU Darmstadt and combines expertise from Computer Science and Mechanical Engineering. In 2015, the team won both the champion title and the Best in Class Autonomy award of the RoboCup Rescue Robot League competition.
We provide an overview of the complete system used to solve the problem of reliably finding victims in harsh USAR environments. This includes hardware as well as software solutions and diverse topics like locomotion, SLAM, pose estimation, human robot interaction and victim detection. In 2015, the team focuses on improving the rough terrain motion capabilities of used platforms as well as manipulation capabilities. As a contribution to the RoboCup Rescue community, major parts of the used software have been released and documented as open source software for ROS.

Introduction

Team Hector Darmstadt (Heterogeneous Cooperating Team of Robots) has been established in late 2008 within the Research Training Group GRK 1362 "Cooperative, Adaptive and Responsive Monitoring in Mixed Mode Environments" (http://www.gkmm.tu-darmstadt.de) funded by the German Research Foundation (DFG). This program addresses two exciting and challenging research areas: (1) navigation and coordination of multiple autonomous vehicles to perform a common task possibly together with a human mission manager; and (2)

⋆ This research has been supported by the German Research Foundation (DFG) within the Research Training Group 1362 "Cooperative, adaptive and responsive monitoring in mixed mode environments"

monitoring in mixed mode environments that are characterized by the heterogeneity of their components in terms of resources, capabilities and connectivity. Driven by the goal of using heterogeneous hardware and software in disaster environments, a successful participation in RoboCup Rescue is an important milestone for these efforts. The interdisciplinarity of our Research Training Group allows us to combine established knowledge and elaborate tools from different disciplines to develop new solutions for search and rescue applications.

The team successfully participated in RoboCup Rescue 2009 for the first time. At the RoboCup German Open 2011, 2012 and 2013 the team won both the overall ranking and the Best in Class Autonomy competition (three times in a row). At RoboCup 2012, the Best in Class Autonomy award, the second place in the overall competition and a special Commendation of Innovation award were achieved, the latter for sharing software with the community. At RoboCup 2013, the team again won the Best in Class Autonomy award. The 2014 season was the most successful for the team yet, winning both the overall competition and the Best in Class Autonomy at RoboCup German Open 2014 and the RoboCup World Championships in Brazil.

Contributing to a initiative within the RoboCup Rescue community to establish an open source framework for USAR robotics [1], the team has released many of the software modules used to achieve top scores at the RoboCup competition as open source software for ROS to facilitate progress and reduce the need for re-inventing the wheel (for example every team having to develop their own SLAM system from scratch).

Ground robots used during previous RoboCup competitions are based on the R/C model Kyosho Twin Force (later on referred to as "Hector UGV"). The vehicles are mechanically modified for better autonomous handling. They are equipped with a modular autonomy box with a onboard computer and several sensors. The main sensor for mapping and navigation is a Hokuyo UTM-30LX LIDAR with a range of 30m which can be rotated around the roll and pitch axis to keep the scan plane parallel to the ground plane. For victim detection and identification we developed a pan/tilt unit with a RGB-D and thermal camera is mounted. The autonomy box can be used as a stand-alone component for testing or can be attached to other robots to enable autonomous exploration and victim detection.

Major additions and changes compared to previous years participation are:

- Improved performance of tracked vehicle naviation
- Motion planning and control capabilities for traversing rough terrain autonomously or semi-autonomously.
- Active gaze control for victim search
- Adapting approaches for semi-autonomous control and manipulation developed within the scope of Team ViGIR [2] and participation in the DARPA Robotics Challenge for use with tracked USAR robots.

A comprehensive overview of recent research can be found in [3]. An overview of released open source packages for mapping and navigation is available in [4]. A video of the operator station screen at the RoboCup German Open 2014

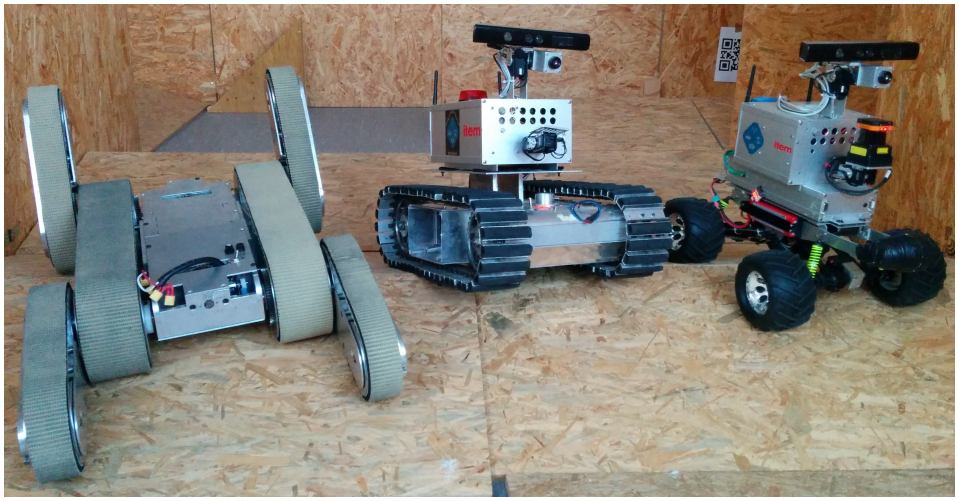


Fig. 1. Umanned Ground Vehicles used by Team Hector: Highly mobile tracked platform with Flippers, tracked platform with adapted autonomy box, Hector UGV used in previous competitions.

Rescue Robot League final mission by Team Hector is available online [5]. It shows how the components described throughout this paper enable Best in Class autonomous behavior.

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`.

1 Team Members and Their Contributions

- Stefan Kohlbrecher: Team Leader, SLAM, GUI, Planning
- Alexander Stumpf: 3D Perception/Motion Planning
- Florian Kunz: ROS Software Integration/Infrastructure
- Dorothea Koert: 3D Sensor Coverage Mapping
- Christian Rose: Systems Integration, Mechatronics
- Kevin Daun: Rough Terrain Motion Planning
- Johannes Schubert: Rough Terrain Motion Planning
- Paul Manns: Motion Control
- Philipp Overath: Hardware Design/Mechatronics
- Vladimir Rezanov: Mechanical Design
- Marvin Just: Mechanical Design
- Benedikt Wartusch: 3D Motion Planning
- Viviane Seidel: Tracked Vehicle Simulation
- Oskar von Stryk: Advisor

2 Operator Station Set-up and Break-Down (10 minutes)

The system consists of one or more robots capable of autonomous or tele-operation via a laptop computer. All of the control equipment easily fits into a standard backpack and depending on the robots used, robots can be carried by hand (wheeled Hector UGV) or should be carried by two persons (tracked vehicles). To start a mission, the robots and the laptop have to be switched on, and the operator can connect to the robots via Wireless LAN.

3 Communications

A COTS wireless network system is used for high-bandwidth data like video images or map information. Currently we use a 2.4 GHz 802.11g/n network, but our hardware also allows 5 GHz or 802.11a/n operation if necessary. The operator station is connected to a wireless access point to which the robot(s) connect.

Rescue Robot League Hector Darmstadt (Germany)			
Technology	Frequency (selectable)	Power	Bandwith (nominal)
2.4 GHz – 802.11g/n	channel 1-13	32 mW	54-300 MBit/s
5.0 GHz – 802.11a/n	channel 36-54	32 mW	54-300 MBit/s

Table 1. communication channels used

4 Control Method and Human-Robot Interface

Our research focus is on autonomy and human-supported autonomous systems, therefore we did not develop sophisticated operator interfaces, but instead concentrated on application-independent methods for better autonomous (team-) behavior and human supervision of autonomous systems. Also when using only a single robot, we use methods that are applicable to robot teams, to ensure general and extensible solutions.

Mission Control: The mission is modeled as a collection of independent tasks. New tasks can be generated during runtime by any module of the control software. For each task a cost value is calculated based on the expected required resources (e. g. time, energy) and the expected benefit (e. g., chance to find a victim), which is dependent on the current configuration of each robot and the knowledge about the environment. A task allocation algorithm (currently a simple greedy algorithm, which can be easily exchanged for, e. g., a market-based solution) assigns a suitable task to each robot based on the calculated costs.

The execution of each different task type is modeled using hierarchical state machines. This is currently realized using the *smach* because of easy integration with other ROS-based components. Previously, the Extensible Agent Behavior Specification Language XABSL [6] was used.

Monitoring and Human Supervision: The robots support the human supervisor in obtaining situation overview (SO) by providing events about their current status, health and progress. In that way, it is possible to send only data that contain relevant information, while other data that do not advance operator SO can be omitted, reducing required bandwidth [7]. The robots can transfer critical decisions to the supervisor by sending queries. In full autonomy mode this is only used for confirming victims. In general, the level of autonomy can be adjusted by transferring more decisions to the supervisor.

The supervisor can actively control the mission flow in two ways: On the one hand, new tasks can be added to the mission, and existing tasks can be modified or deleted. On the other hand, the allocation of tasks to robots can be influenced by systematical modifications to the calculated task costs, which enables the supervisor to directly assign tasks to robots, let groups of tasks be executed with higher priority, or temporarily forbid execution of specific tasks or task groups [8].

The mission modeling and task allocation as described in the previous paragraph, and the control concept for the supervisor can be applied to single robots as well as robot teams, and therefore allow to easily extend our approach to heterogeneous robot teams.

Teleoperation: In cases supervisory control is not sufficient (especially in difficult terrain in the orange and red arena), all vehicles can also be fully teleoperated using a gamepad, joystick, or the keyboard. In this case the operator uses the map and video-streams to obtain situation awareness.

Graphical User Interface: Using ROS as middleware, the *rviz* visualization tool can be used for visualizing maps and other arbitrary data, and for sending waypoints to the robots. As a second important tool we use *rqt*, which includes graphical dialogs for publishing and receiving messages, calling services, and visualizing the interactions between all active nodes. Additional plugins can be written in Python or C++ and can be loaded by *rqt*, thus providing an integrated GUI for mission control as well as for debugging.

5 Map Generation/Printing

The Simultaneous Localization And Mapping (SLAM) problem is solved by using a 2D grid map representation that gets updated using a scan matching approach [9]. The approach has low runtime requirements and can run with an update rate of 40Hz while requiring less than 15% CPU time on a Core 2 Duo setup, freeing resources for other computation. The system does not require odometry data, as the scanmatching approach is very robust. The input used for solving the SLAM

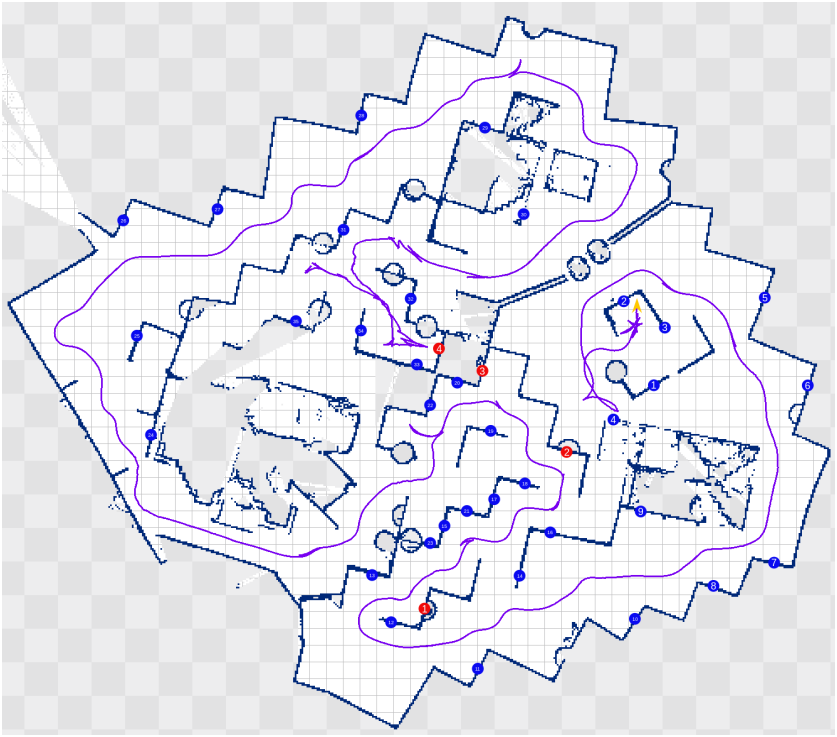


Fig. 2. Map learned using the *hector_slam* system at the final mission of RoboCup 2012. The robot started at the right middle position and autonomously explored the majority of the arena, finding 3 victims (red markers). The fourth victim was found using tele-operation on the last 4 meters of the travelled path. Blue markers indicate the positions of 35 QR codes that were detected autonomously by the robot.

problem are laser scans and the robot state as estimated by the navigation filter (cf. Section 6). Data provided by the navigation filter is used for transformation of laser scans to take into account the attitude of the laser scanner and vehicle during acquisition of scans. Figure 5 shows a map learned using the *hector_slam* system. A video is available online [10].

The map can be manually or automatically annotated with information about victims and other objects of interest. It can be saved in the GeoTIFF format using the *hector_geotiff* package. The described software is available and documented as open source software in the *hector_slam* stack for ROS, which is widely used within the RoboCup Rescue League and beyond.

To enable autonomous cooperative deployment of multiple robots on missions, a feature based map merging system has been developed. Each robot detects ORB features [11] for the estimated map and these are exchanged among teammate robots. A registration approach is then used to estimate a common

coordinate frame for all robots. A map stitching command line tool is available in the *mapstitch* package.

To better negotiate the increasingly difficult terrain in the rescue arena, a RGB-D camera is mounted on the pan/tilt unit of the robot. This allows to acquire point clouds, build a 2.5D height map and classify the terrain into passable and impassable grid cells. A 3D map of the environment is generated using a modified version of the *octomap* mapping package [12]. Raycasting into the 3D map is used to determine distances to objects of interest detected with imaging sensors. The 3D map also serves as the basis for a active gaze control approach that keeps track of observed parts of the environment and can control the gaze (pan/tilt motion) of camera sensors accordingly.

6 Sensors for Navigation and Localization

Wheel/Track Encoders: To measure the translational and rotational speed of vehicles, all used vehicles are equipped with encoders measuring wheel or track motion. This odometry data is used for low level speed control. Due to frequent slippage in the considered scenarios, it is not used in the SLAM system.

Laser Scanner: The vehicle is equipped with a Hokuyo UTM30-LX LIDAR. It is mounted on a roll/tilt unit at the front of the autonomy box and is mainly used for 2D mapping. The LIDAR system can be stabilized to stay close to the intended scan plane regardless of vehicle attitude.
Optionally, a Hokuyo URG-04LX or SICK TiM300 LIDAR can be mounted on the back of the vehicle. Pointing towards the ceiling, this LIDAR allows the acquisition of additional 3D data.

RGB-D Camera: A RGB-D camera is used for environment perception tasks like traversable terrain detection, 3D mapping and also for victim verification. This camera is mounted on the pan/tilt unit that is also used for the camera. We currently use the Microsoft Kinect sensor, but might exchange this for a smaller solution like the PrimeSense SDK 5.0 sensor or ASUS Xiton Pro Live.

Ultrasound Range Finders: Additionally to the LIDAR, a set of ultrasound range finders mounted at the back of the vehicle enables autonomous reactive collision avoidance when moving backwards, as the LIDAR only covers a 270 degrees field of view.

Inertial Measurement Unit: To measure the attitude of the platform, vehicles are equipped with a 6DoF inertial sensor ADIS16350 by Analog Devices which measures accelerations and angular rates (IMU).

Navigation filter: Information from the IMU and the scan matcher is fused to get an overall estimate of position, velocity and attitude of the vehicle using an Extended Kalman filter. This is realized in the *hector_localization* stack. Although Kalman filtering is a common and simple approach for robot navigation problems, it suffers amongst others from the resulting unimodal representation of the belief state. On the other side, the feedback from map-based localization as described in section 5 can lead to ambiguities which contradict the Gaussian assumption. Our approach is to combine these two sources of information in a loosely-coupled way in order to achieve a robust navigation solution [9]. The attitude estimate of the navigation filter is used to stabilize the LIDAR and camera system.

7 Sensors for Victim Identification

Finding human victims under difficult conditions of unstructured post-disaster environments is one of the main goals of RoboCup Rescue. Significant progress in visual object recognition and scene understanding allows us to apply state of the art computer vision methods. To tackle this problem we use a multi-cue victim detection system supporting optical image cues like RGB, thermal and depth images. This complementary information can be used to increase reliability.

Once the detector has recognized a victim or other object of interest this detection is forwarded to the *hector_object_tracker* which keeps track of known objects and updates this model based on positive and negative evidence. The separation of object detection and modeling enables the flexible integration of different sensory sources for various classes of objects. The position and pose of each object is tracked using a Kalman Filter. The *hector_object_tracker* is the only interface between perception and control, e.g. for the creation or modification of tasks or the manipulation of model state due to operator interaction.

A comprehensive overview of our approach to semantic mapping using heterogenous sensors such as thermal and visual cameras can be found in [13].

Thermal- and Depth-Based Victim Detection: In addition to visual victim detection we use a thermal and also a RGB-D camera to verify vision-based hypotheses.

In most cases images provided by the thermal camera are very helpful for identifying possible victim locations. As a drawback of a thermal camera the thermal images often contain not only victims but also other warm objects, such as radiators or fire, so that thermal and visual recognition systems will deliver complementary information.

To further reduce false-positives we use point clouds from the RGB-D camera to evaluate the environment of the victim hypotheses. False-positive victim hypotheses can be identified by the shape of the environment or by missing depth measurements at the victim location.

QR Code Detection As a step towards more exhaustive sensor coverage of the environment and future detection of additional objects of interest, QR codes are placed in the RoboCup Rescue arena, with points being awarded for their successful detection and localization in the map. This task is supported by two PS Eye cameras pointing out of the left rear and right rear of the autonomy box. Using the *hector_qrcode_detection* package, QR code detection is by default running for the Kinect RGB-D camera and both PS Eye cameras, for maximum coverage of the robots surroundings. Detections are used as input for the *hector_object_tracker* as described above for the victim detection case.

8 Robot Locomotion

8.1 Hector UGV

The Hector UGV vehicle is based on a Kyosho Twin Force RC model originally optimized for high speeds. The following modifications have been implemented to permit reliable autonomous operation.

4-Wheel Drive: The 4-wheel drive of the vehicle has one differential gear per axis and no middle differential gear. This ensures that the vehicle is able to move when only some of the wheels have ground contact. To reduce the maximum speed for autonomous operation in harsh terrain and to increase torque a 1:5 reduction gear was added.

4-Wheel Steering: The steering angle of front and rear wheels can be controlled independently, providing advantages over normal 2-wheel steering:

- a smaller minimum turn radius (half of 2-wheel steering),
- the ability for the rear wheels to use the same trajectory as the front wheels (if both steering angles are the same)
- the ability to move sidewards (up to 35 degrees to the longitudinal axis of the vehicle).

Usually, the rear wheels are set to the same steering angle as the front wheels, so that the resulting trajectories are identical and the risk of obstacle contact is reduced.

8.2 Tracked Vehicles

The tracked vehicles use DC motors for driving their tracks. The medium size UGV without flippers uses a roller chain track using metal brackets fitted with rubber elements mounted to each chain track element. The flipper equipped robot uses toothed belt system to reduce track maintenance requirements. The left and right side flippers are linked, but the front and back pair can be actuated independently.

9 Other Mechanisms

9.1 Hardware Modularity

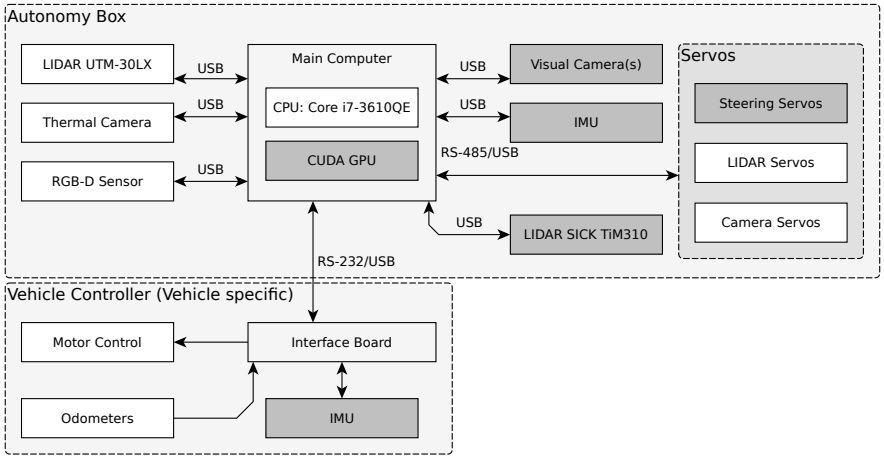


Fig. 3. Structure of hardware components. Dark grey boxes indicate optional components with the capability to quickly add/remove them as needed.

The complete hardware structure of the Hector vehicle setup is shown in Fig.3. To achieve interoperability between different vehicle platforms, the autonomy box is connected to the vehicle base using a single USB connection (or possibly Ethernet in the future). The sensor and software setup can thus be largely (if not completely) agnostic to the underlying platform employed for locomotion. The autonomy box is equipped with a state-of-the-art Intel Core i7 mobile CPU and an optional high-performance GPU serves and serves as the main control system.

The separation of vehicle base and autonomy box, even on the hardware layer, simplifies independent testing and offers a high degree of flexibility. The perception system can easily be mounted on other robots or used as a separate instrument for the evaluation of test scenarios. Both autonomy box and the wheeled robot base can and have been used in various indoor and outdoor scenarios as a flexible and lightweight research platform.

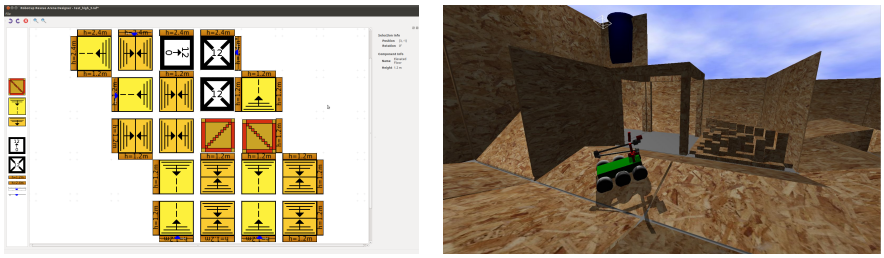
9.2 Handheld Mapping System

As the SLAM system described in Section 5 does not require odometry data, it can also be used in a handheld mapping system weighting around 1kg. This system can easily be carried around by a person and be used to learn a map

of the environment in realtime [14]. A video of the system being used in the RoboCup 2011 Rescue Arena is available online [15]. The system might form the baseline for a system usable by First Responders. ROS bagfiles acquired using the system for the RoboCup German Open 2011 and the RoboCup 2011 Rescue Arenas are both available for download [16].

9.3 USAR Scenario Simulation

With ROS getting used by multiple teams participating in the RoboCup Rescue League, a common framework for simulation is also desirable. USARsim [17] historically has been widely used for RoboCup Rescue related simulation (especially in the Simulation League), but currently has no integrated ROS support. For this reason, we employ the *gazebo* simulator widely used within the ROS community for the simulation of USAR scenarios. The *hector_nist_arenas_gazebo* ROS stack allows the fast and user friendly creation of simulated disaster scenarios using elements of the NIST standard test arenas for response robots (Fig. 4).



(a) (b)

Fig. 4. USAR scenario simulation: (a): Screenshot of the GUI tool for creating USAR scenarios showing an example scenario (b): The same scenario as simulated in *gazebo* simulation

9.4 Quadrotor UAV

In the context of monitoring in mixed mode environments, members of the team are also performing research on obstacle avoidance, localization and mapping and victim detection [18] using quadrotor UAVs. While the use in the RoboCup Rescue competition is not planned in the short term, this might become a possibility in coming years. To facilitate research in this direction, we made the *hector_quadrotor* stack available, which allows simulation of quadrotor UAVs using *gazebo*, allowing comprehensive simulation of the whole system including external sensors like LIDAR, RGB-D and camera sensors. A comprehensive description is available in [19]

10 Team Training for Operation (Human Factors)

The mission control dialog provides all crucial high level information about the ongoing mission to the operator, so unmanned vehicles can be supervised and controlled without detailed knowledge about their specific capabilities like kinematics and dynamics. UGVs classify terrain into passable and impassable sections, so they generally do not need external supervision for exploring the environment. High level control of multiple UGVs is thus possible without expert knowledge about the vehicles. However, depending on the situation, the autonomy level might has to be lowered, in which case an operator has more direct control of vehicles and thus needs to have more detailed knowledge about them. In the RoboCup Rescue scenario, we use only one operator, as the number of robots employed simultaneously is small. For other scenarios, different operators can be responsible for different autonomy levels and tasks.

We train operators in using the mission control interface and in teleoperation of robots. As mentioned before, the focus on our research lies in autonomy, so training in teleoperation is not as comprehensive as for many teams focusing on teleoperated robots.

11 Possibility for Practical Application to Real Disaster Site

The strength of our approach is the elaborate reusable software, which is a reliable base for developing and extending our system. With multiple mobile platforms available while keeping the same sensor and software setup, the mobility platform can be chosen depending on the task at hand. The Hector UGV system is lightweight and fast, while tracked vehicles are heavier but also more mobile in rough terrain conditions. For practical application to real disaster sites, hardening of hardware against environmental conditions (dust, water, humidity etc.) and improved robustness of sensor data processing in such environments are required.

12 System Cost

As we employ a modular system with the processing/perception unit being deployable on multiple platforms, overall system cost depends on the platform used.

Vehicle

Component	**Model**	**Price**
Chassis	Modified Kyosho Twin Force	2000 EUR
	Tracked vehicle w/o flippers	4000 EUR
	Tracked vehicle with flippers	10000 EUR

Autonomy Box

Vision Computer	Intel Core i7-3610QE	700 EUR
Thermal Camera	ThermalEye 3600AS	3100 EUR
RGB-D Camera	Kinect Sensor	130 EUR
Cameras	2 PS Eye Sensors	100 EUR
Laser Scanner	Hokuyo UTM-30LX	4200 EUR
Servos	Robotis RX-10/RX-28	320 EUR
Power Supply	M4 ATX	100 EUR
Miscellaneous		200 EUR
Total Vehicle Cost	Hector UGV	**10850 EUR**
	Tracked vehicle w/o flippers	**12850 EUR**
	Tracked vehicle with flippers	**18850 EUR**

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 Thanakit Wattakeekamthorn, Kamontip Wattakeekamthorn, Worawat Chaiwong, Chonlakarn Wongkhorsub and all other members of Team Stabilize for their support and cooperation in the acquisition of the tracked robot systems described in this paper.

References

1. S. Kohlbrecher, K. Petersen, G. Steinbauer, J. Maurer, P. Lepej, S. Uran, R. Ventura, C. Dornhege, A. Hertle, R. Sheh, , and J. Pellenz. Community-driven development of standard software modules for search and rescue robots. In *Proceedings of the 10th IEEE International Symposium on Safety Security and Rescue Robotics (SSRR 2012)*, 2012.
2. Stefan Kohlbrecher, David C Conner, Alberto Romay, Felipe Bacim, Doug A Bowman, and Oskar von Stryk. Overview of team vigir’s approach to the virtual robotics challenge. In *Safety, Security, and Rescue Robotics (SSRR), 2013 IEEE International Symposium on*, pages 1–2. IEEE, 2013.
3. Stefan Kohlbrecher, Florian Kunz, Dorothea Koert, Christian Rose, Paul Manns, Kevin Daun, Johannes Schubert, Alexander Stumpf, and Oskar von Stryk. Towards highly reliable autonomy for urban search and rescue robots. In *Proc. RoboCup Symposium 2014*, Lecture Notes in Artificial Intelligence (LNAI). Springer, 2014.
4. Stefan Kohlbrecher, Johannes Meyer, Thorsten Graber, Karen Petersen, Oskar von Stryk, Uwe Klingauf, K Petersen, A Kleiner, O von Stryk, S Kohlbrecher, et al. Hector open source modules for autonomous mapping and navigation with rescue robots. *Proc. RoboCup Symposium 2013*, pages 251–260, 2013.

5. Operator station recording of final mission at RoboCup German Open 2014 RRL final mission on YouTube. https://www.youtube.com/watch?v=Q_ChNVIsmeE, 2014.
6. M. Lötzsch, M. Risler, and M. Jüngel. XABSL - A Pragmatic Approach to Behavior Engineering. In *Proceedings of IEEE/RSJ International Conference of Intelligent Robots and Systems (IROS)*, pages 5124–5129, Beijing, China, 2006.
7. K. Petersen and O. von Stryk. An event-based communication concept for human supervision of autonomous robot teams. *International Journal on Advances in Intelligent Systems*, 4(3&4):357 – 369, 2011.
8. K. Petersen and O. von Stryk. Application independent supervised autonomy. In *Proceedings of the 10th IEEE International Symposium on Safety Security and Rescue Robotics (SSRR 2012)*, 2012.
9. S. Kohlbrecher, J. Meyer, O. von Stryk, and U. Klingauf. A flexible and scalable slam system with full 3d motion estimation. In *International Symposium on Safety, Security, and Rescue Robotics*. IEEE, November 2011.
10. Video of Hector UGV during Best in Class Autonomy Challenge on YouTube. http://www.youtube.com/watch?v=nI1DWboC73w, 2010.
11. Ethan Rublee, Vincent Rabaud, Kurt Konolige, and Gary Bradski. Orb: an efficient alternative to sift or surf. In *Computer Vision (ICCV), 2011 IEEE International Conference on*, pages 2564–2571. IEEE, 2011.
12. A. Hornung, K. M. Wurm, M. Bennewitz, C. Stachniss, and W. Burgard. OctoMap: An efficient probabilistic 3D mapping framework based on octrees. *Autonomous Robots*, 2013. Software available at http://octomap.github.com.
13. 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*, Lecture Notes in Computer Science, pages 180–193, 2011.
14. P. M. Scholl, S. Kohlbrecher, V. Sachidananda, and K. van Laerhoven. Fast indoor radio-map building for rssi-based localization systems. In *Demo Paper, International Conference on Networked Sensing Systems*, 2012.
15. Video of Handheld Mapping System used in RoboCup 2011 Rescue Arena on YouTube. http://www.youtube.com/watch?v=F8pdObV_df4, 2011.
16. Datasets available for download. http://code.google.com/p/tu-darmstadt-ros-pkg/downloads/list, 2011.
17. S. Carpin, M. Lewis, J. Wang, S. Balakirsky, and C. Scrapper. Usarsim: a robot simulator for research and education. In *ICRA*, pages 1400–1405. IEEE, 2007.
18. M. Andriluka, P. Schnitzspan, J. Meyer, S. Kohlbrecher, K. Petersen, O. von Stryk, S. Roth, and B. Schiele. Vision Based Victim Detection from Unmanned Aerial Vehicles. In *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, October 2010.
19. J. Meyer, A. Sendobry, S. Kohlbrecher, U. Klingauf, and O. von Stryk. Comprehensive simulation of quadrotor uavs using ros and gazebo. In *3rd Int. Conf. on Simulation, Modeling and Programming for Autonomous Robots (SIMPAR)*, page to appear, 2012.