RoboCupRescue 2013 - Robot League Team

Hector Darmstadt (Germany)

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

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 2013 RoboCup Rescue 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. We give 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. 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 (<u>He</u>terogeneous <u>C</u>ooperating <u>T</u>eam <u>o</u>f <u>R</u>obots) 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) 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

---

* 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"

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, improving every year. The team won both the overall and the Best in Class Autonomy competition at the RoboCup German Open 2011 and 2012. 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.

Recently, a initiative within the RoboCup Rescue community to establish an open source framework for USAR robotics is gaining increasing interest and support [1]. Contributing to this effort, 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).

Our ground robots are based on the R/C model Kyosho Twin Force (later on referred to as "Hector UGV", Fig. 1). The vehicles are mechanically modified for better autonomous handling and equipped with onboard computers and several sensors. The main sensor for mapping and navigation is a Hokuyo laserscanner (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 vision extension, including a visual, a thermal and a depth camera mounted on a pan/tilt unit. The control box can be used as a stand-alone component for testing or can be attached to another robot to enable autonomous exploration and victim detection.

Based on experience from previous RoboCup competitions, several improvements have been made to the chassis of the robot. Compared to the original design, the steering system uses less connection rods and stronger digital servos, yielding more direct control of the steering angles even on rough terrain or when wheels are blocked for some reason.

The major additions and changes compared to the system as used in 2012 are:

- Increased use of 3D sensing and modeling. This includes the use of the octomap [2] library to generate 3D environment maps as well as sensor coverage mapping and active gaze control for the victim detection sensors.
- Upgrade of the main onboard computer system to use a Core i7 CPU in order to permit the use of computationally demanding 3D mapping approaches.
- Possible deployment of multiple cooperating UGVs at the same time with map merging.

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`.

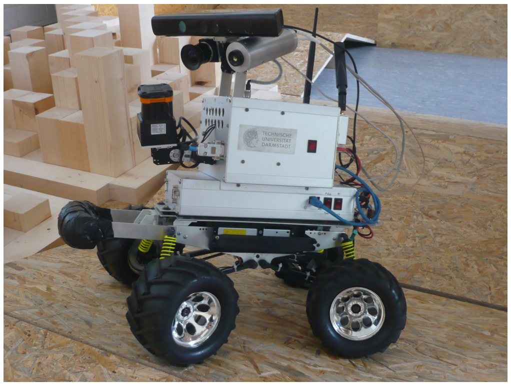


Fig. 1. Umanned Ground Vehicle "Hector UGV".

1 Team Members and Their Contributions

- Stefan Kohlbrecher: Team Leader, SLAM, GUI
- Karen Petersen: Behavior, HRI, Team Cooperation
- Johannes Meyer: Hardware, Navigation and Control, Simulation
- Thorsten Graber: Point Cloud Processing
- Florian Kunz: ROS Software Integration/Infrastructure
- Mark Sollweck: Exploration/Global Path Planning
- Konstantin Fuchs: Victim Detection/Thermal Imaging
- Johannes Simon: Development of an arena designer GUI for gazebo
- Florian Berz: Exploration based on First Responder methods
- Clemens Wronski: Hardware Design
- Dorothea Koert: 3D Sensor Coverage Mapping
- Gregor Gebhardt: Multi-Robot Communication
- Oskar von Stryk: Advisor
- Uwe Klingauf: Advisor

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

The system consists of one or more lightweight Hector UGVs capable of autonomous or tele-operation via a laptop. All of the control equipment easily fits into a standard backpack and the Hector UGV(s) can be carried by hand. 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

Our communication concept is based on two different channels. A common wireless network 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. For data exchange with lower bandwidth demands the vehicle is additionally equipped with a 802.15.4 radio modem. This low-bandwidth link is used for telemetry and basic manual control of the vehicle and enables the operator to take over even when the onboard computer is no longer operational. The operator station is connected to a modified wireless access point which interfaces both networks, 802.11a/g and 802.15.4.

Rescue Robot League Hector Darmstadt (Germany)			
Technology	Frequency (selectable)	Power	Bandwith (nominal)
2.4 GHz – 802.11g	channel 1-13	32 mW	54-300 MBit/s
5.0 GHz – 802.11a	channel 36-54	32 mW	54-300 MBit/s
2.4 GHz – 802.15.4	channel 11-26	100 mW EIRP	115 kBit/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 robots' 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 with the Extensible Agent Behavior Specification Language XABSL [3], but because of the easier integration into ROS the system will be ported to use *smach*.

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 the human's SO can be omitted, thus reducing the required bandwidth [4]. The robots can transfer critical decisions to the supervisor by sending queries, which is in full autonomy mode 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 [5].

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: Since we started using ROS as middleware, the *rviz* visualization tool scan be used for visualizing maps and other arbitrary data, and for sending waypoints to the robots. As a second important tool we use *rqt_gui*, 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_gui*, 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 [6]. Our 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 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 [7].

To enable autonomous cooperative deployment of multiple robots on missions, a feature based map merging system has been developed. Each robot detects SURF features [8] for the estimated map and these are exchanged among teammate robots. A registration approach is then used to arrive at a common coordinate frame for all robots.

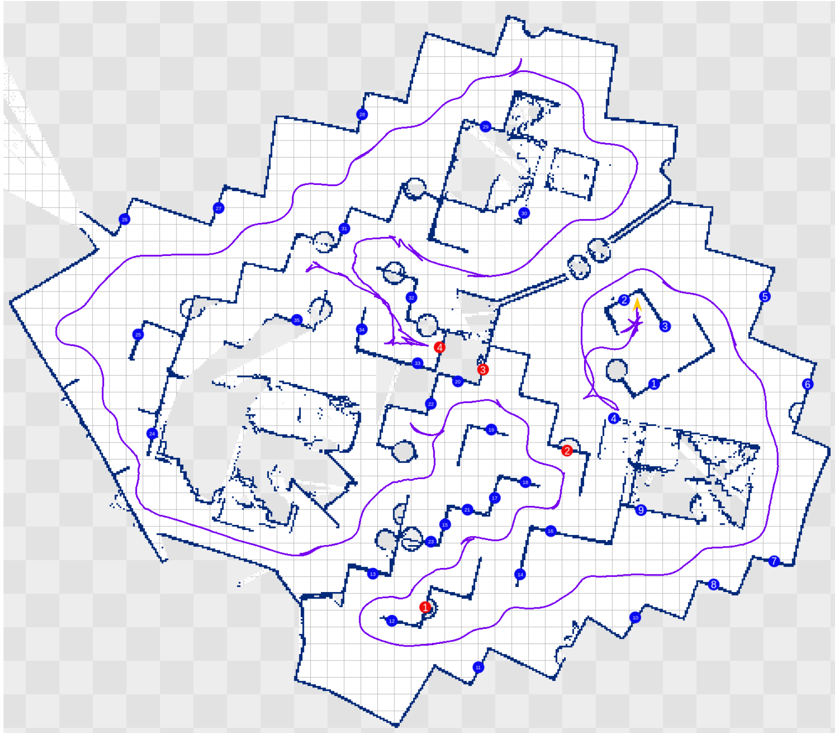


Fig. 2. Map learned using the *hector_slam* system at the final mission of RoboCup 2012. The robot started at the right middle position and autonomously explored the majority of the arena, finding 3 victims (red markers). The fouth victim was found using tele-operation on the last 4 meters of the path.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

To better negotiate the increasingly rough terrain in the rescue arena, we added a RGB-D camera mounted on the pan/tilt unit of the robot to acquire point clouds, then build a 2.5D height map and classify the terrain into passable and impassable grid cells. Our software makes use of the *Point Cloud Library* (PCL) which is available as a ROS package.

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. Most of the software described in this section is available and documented as open source software in the *hector_slam* stack for ROS.

6 Sensors for Navigation and Localization

Wheel Encoders: To measure the translational and rotational speed of the vehicle, all four wheels are equipped with incremental optical encoders. This odometry data is used for low level speed control, but due to noise additional feedback from other sensors is needed.

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 control 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 LIDAR can be mounted on the back of the vehicle. Pointing towards the ceiling, this LIDAR allows the acquisition of 3D data about the environment of the robot.

RGB-D Camera: We use a RGB-D camera 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 vehicle, it is 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 [6]. 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 *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 *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 [9].

Vision-Based Recognition of Victims and Hazmat Symbols: The recognition of the objects is performed by using a combination of visual cues based on the gradients of image intensity. Such cues can be efficiently captured by a descriptor based on the histograms of oriented gradients (HOG, see Fig. 3 for illustration). First, the gradient magnitude and orientation are computed densely in the image. The local distributions of the gradient orientation are then captured by the histogram. Such histograms are then grouped with their neighbors and jointly normalized. The normalization and local pooling of gradient information significantly improves the stability of the description to viewpoint changes, noise, and changes in illumination.

It has been demonstrated that visual information represented in this way combined with powerful machine learning techniques can be successfully applied to recognition of people in realistic conditions [10]. While showing good performance this approach also requires significant processing power. The on-board computer (Fig. 4) with an nVidia graphics card allows real-time feature computation and recognition with an implementation based on [11].

We use the recognition system for detection of hazmat symbols at the victim sites (Fig. 3). The same system, but trained on the images of human body parts, is used to recognize victims parts.

In further work relevant to the USAR scenario, we examined people detection from UAVs [12]. We improved part-based people detection algorithms for detecting people in arbitrary poses with partial occlusion by projecting the images to the ground plane, adding a scale prior, and combining the two best-performing algorithms. This leads to an equal error rate (EER) of 66%, compared to EER of 21.9% of the upper-body HOG detector. Some examples for both detectors can be seen in Fig. 5. However, because the objects we want to detect in the RoboCup scenario are not articulated humans, but rather rigid objects like baby dolls and hazmat signs, the HOG detector is sufficient, while requiring less computational power.

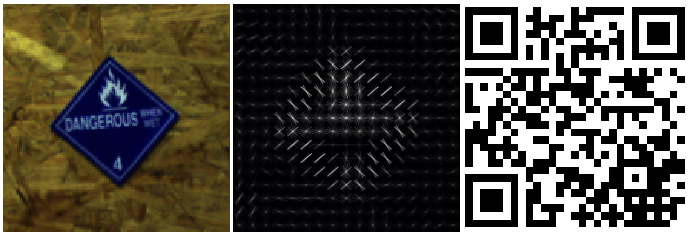


Fig. 3. Original Image (left), histogram of oriented gradients (middle) and an example for a QR code (right).

Fig. 4. Our mobile computing platform with CUDA capable GPU (left) and uEye camera (middle), and a picture taken by the camera at RoboCup German Open 2009 in Hannover (right).

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.

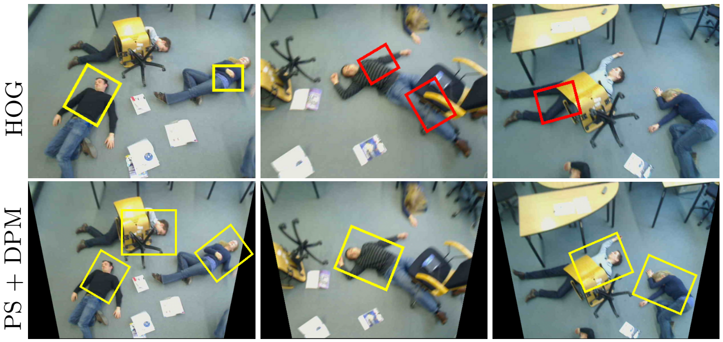
In most cases images provided by the thermal camera are very helpful for identifying possible victim locations. As a drawback of a thermal camera the thermal images often contain not only victims but also other warm objects, such as radiators or fire, so that thermal and visual recognition systems will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

8 Robot Locomotion

The vehicle is based on a Kyosho Twin Force RC model with a powerful drive train optimized for high velocities. For navigation in USAR scenarios the drive train, steering and suspension systems have been modified because of the much higher mass compared to the original vehicle.

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



(a) (b) (c)

Fig. 5. Several examples of detections at EER obtained with a HOG detector [10] (1st row), and the combined detector augmented with scale prior [12] (2nd row). True positive detections are plotted with yellow and false positives with red color.

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.

9 Other Mechanisms

9.1 Hardware Modularity

The complete hardware structure of the Hector UGV vehicle is shown in Fig.6. The intrinsic sensors and actuators are connected to an interface board which communicates with a PC/104 computer, which enables 6DoF navigation and allows basic autonomous driving. Most of the extrinsic sensors are connected to a separate on-board computer which is equipped with a state-of-the-art Intel

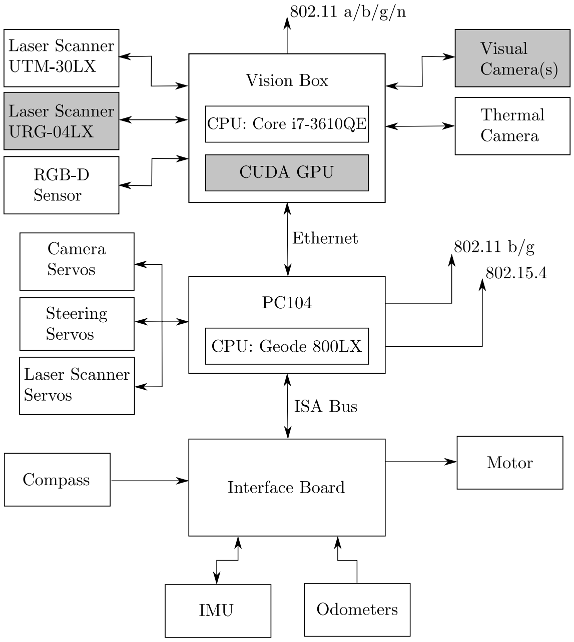


Fig. 6. Structure of hardware components. Grey boxes indicate optional components that can quickly be added/removed as needed.

Core i7 mobile CPU and an optional high-performance GPU for parallel computing. This "vision box" fulfills the more demanding tasks of mapping and visual detection of victims and hazmat symbols.

The separation of both components, even on the hardware layer, simplifies independent testing and offers a high degree of flexibility. The vision computer can easily be mounted on other robots or used as a separate instrument for the evaluation of computer vision algorithms. The robot itself can and has been used in various indoor and outdoor scenarios as a flexible and lightweight research platform.

9.2 Handheld Mapping System

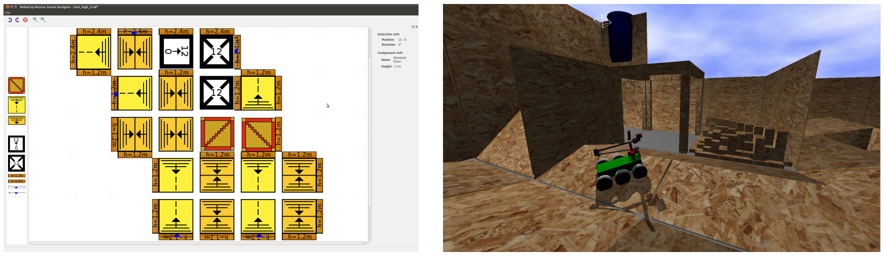
As the SLAM system described in Section 5 does not require odometry data, it can also be used in a handheld mapping system weighting around 1kg. This system can easily be carried around by a person and be used to learn a map of the environment in realtime [13]. A video of the system being used in the RoboCup 2011 Rescue Arena is available online [14]. 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 [15].

9.3 Smartphone User Interface

Dedicated operator stations often have a large footprint and need dedicated hardware and space for setup. In contrast to this, state of the art smartphones are ubiquitously available and sufficiently powerful for tasks like displaying a video stream and sensor data. For this reason, we developed a proof-of-concept teleoperation interface that can be used for teleoperation of unmanned systems.

9.4 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 [16] 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 investigate the feasibility of using the *gazebo* simulator widely used within the ROS community for USAR scenarios. Among our efforts in this direction is the development of a tool that permits the fast and user friendly creation of simulated disaster scenarios using elements of the NIST standard test arenas for response robots (Fig. 7).



(a) (b)

Fig. 7. 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

9.5 Small, Low Cost UGV System

A low cost, lightweight UGV system based on the commercially available "Wild Thumper" robot kit was first tested at the RoboCup 2011 competition [17]. A sensor arm extending the range of this system is currently in development. Fig. 8 shows a preliminary version. The system is very lightweight ($< 5kg$) and of small size (width 30cm and length 43cm) as to fit into confined spaces while still providing good mobility.

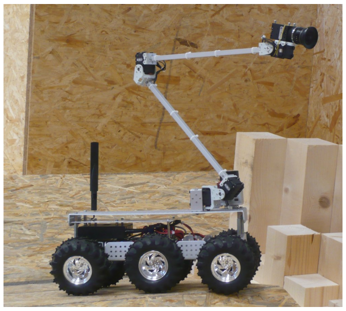


Fig. 8. Unmanned Ground Vehicle "Hector Lightweight UGV".

9.6 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 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 [18]

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 teleoperation.

11 Possibility for Practical Application to Real Disaster Site

The Hector UGV is a flexible vehicle that allows for precise and versatile locomotion. The low weight is a big advantage for fast and flexible setup of the whole system. The most critical points are movement in very rough terrain and sensitivity against some basic environmental factors like humidity.

The strength of our approach is the elaborate reusable software, which is a reliable base for developing and extending our system. For practical application to real disaster sites we have to improve abilities in (partial) autonomy and plan to combine the system with other existing systems like an UAV (Quadrotor) and (mobile) sensor nodes. We hope to be able to give useful, flexible assistance to operators in managing disaster scenario within a few years.

12 System Cost

Vehicle

Component	**Model**	**Price**
Chassis	modified Kyosho Twin Force	300 EUR
Navigation Computer	Lippert Cool LiteRunner	250 EUR
Steering Servos	Robotis RX-28	300 EUR
Odometer	Selfmade	200 EUR
Interface Board	Selfmade	200 EUR
IMU	ADIS16350	300 EUR
Magnetometer	HM55B	25 EUR
Laser Scanner	URG-04LX	1900 EUR
Ultrasound Rangers	SRF05/SRF08	150 EUR
Power Supply	picoPSU-120 + Misc.	100 EUR
Batteries	6 Cell LiPo 5000mAh	240 EUR
Miscellaneous		300 EUR

Vision Extension

Vision Computer	Intel Core i7-3610QE	700 EUR
Visual Camera	uEye UI-2230RE	700 EUR
Thermal Camera	ThermalEye 3600AS	3100 EUR
RGB-D Camera	Kinect Sensor	130 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 Cost		**13715 EUR**

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. 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`.
3. 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.
4. 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.
5. 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.
6. 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.
7. Video of Hector UGV during Best in Class Autonomy Challenge on YouTube. `http://www.youtube.com/watch?v=nI1DWboC73w`, 2010.
8. H. Bay, A. Ess, T. Tuytelaars, and L. Van Gool. Speeded-Up Robust Features (SURF). *Computer Vision and Image Understanding*, 110:346–359, June 2008.
9. 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.
10. N. Dalal and B. Triggs. Histograms of Oriented Gradients for Human Detection. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR'05)*, 2005.
11. C. Wojek, G. Dorkó, A. Schulz, and B. Schiele. Sliding-Windows for Rapid Object Class Localization: A Parallel Technique. In *DAGM-Symposium*, pages 71–81, 2008.
12. 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.
13. 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.
14. Video of Handheld Mapping System used in RoboCup 2011 Rescue Arena on YouTube. `http://www.youtube.com/watch?v=F8pdObV_df4`, 2011.
15. Datasets available for download. `http://code.google.com/p/tu-darmstadt-ros-pkg/downloads/list`, 2011.
16. 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.
17. Videos of HLUGV being tested used in the RoboCup 2011 Rescue Arena on YouTube. `http://www.youtube.com/playlist?list=PL4D55AB35D5DEEF9C`, 2011.
18. 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.