RoboCup Rescue 2016 Team Description Paper Tedusar

Konstantin Lassnig, and Stefan Loigge

Info

Team Name:	Tedusar
Team Institution:	Graz University of Technology
Team Leader:	Konstantin Lassnig
Team URL:	http://www.tedusar.at

RoboCup Rescue 2016 TDP collection:
https://to-be-announced.org

***Abstract*—This paper describes the TEDUSAR Rescue Robot team and their robots developed for the participation in the RoboCup 2016. The team was established as part of the Technology and Education for Search and Rescue Robots (TEDUSAR) project. TEDUSAR was a cross-border project between Slovenia and Austria funded by the European Regional Development Fund (EFRE) and the local governments some years ago. Today the team involved in this project still continues its work in the area of urban search and rescue (USAR). This includes the development of a search and rescue robot, the research devoted to improve this robot, as well as cooperations with first responders.**

***Index Terms*—RoboCup Rescue, Team Description Paper, Tedusar, Autonomy**

I. Introduction

THE TEDUSAR rescue robot team was established as part of the Technology and Education for Search and Rescue Robots project (TEDUSAR). The project was started in September 2011 to foster education, development and research in robotics and artificial intelligence.

TEDUSAR started as a cross-border project between Slovenia and Austria and was funded by the European Regional Development Fund (EFRE) and the local governments. Project partners were the Institute for Software Technology at Graz University of Technology and the Faculty of Electrical Engineering and Computer Science at University of Maribor. The central topic of the project was the promotion of education, research and deployment in the area of urban search and rescue robots.

Since the project launch two rescue robots, *Zaphod* and *Wowbagger*, have been developed and the team has successfully participated in RoboCup German Open 2012, 2013, 2014, 2015 and RoboCup Championship 2014. The main research aim is to develop reliable software for fully autonomous robotic systems. Wowbagger is the major competition platform and designed for exploration and victim detection. The robots are capable to operate fully autonomous by receiving higher level goals from the operator.

A. Improvements over Previous Contributions

The current state of the team's main robot differs primary in software improvements. The team further worked on making the robot more reliable and therefore included a new powerboard which is not only more efficient and universal than the old one, but also offers the capability to diagnose hardware problems. It can measure currents and voltages and every single component can be switched off and on again, which offers the possibility to repair a large amount of hardware issues.

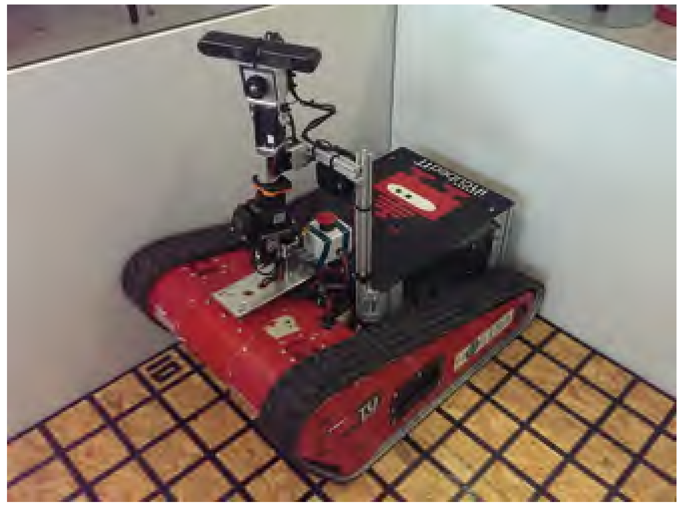

Fig. 1. Wowbagger

Fig. 2. Forbot

Beside observing and diagnosing the hardware, also the software is observed and diagnosed. The observations and diagnoses allow the robot to react on failures. Additionally, statistics and data dumps can be produced for post mortem analysis. Thus useful improvements for future versions of the robot can be derived from this data. Another major advancement is the localization and mapping in 3D, which is essential to detect all kinds of obstacles. The 3D environment is further processed to detect objects of interest which are important to understand the environment. Last but not least some new operator capabilities were added for easily commanding the robot for special movements.

Apart from these improvements, a second robot was developed which will demonstrate manipulation capabilities.

II. System Description

This section includes detailed descriptions of used hardware and software components of both robots.

A. Hardware

As stated before, two different robots are used. The first one is called *Wowbagger* and the component list can be found in table I. It is the major platform and used for almost all tasks except manipulation. The other robot has a different component table II and is specialized for manipulation on even terrain, based on the Forbot platform and equipped with a Schunk robotic arm.

1) Mesa Element Robot: *Wowbagger* is based on the MESA Element tracked robot. It is able to overcome steps which is an advantage compared to the *Forbot*. Special tank-shaped tracks allow it to pass stairs and overcome obstacles of up to 170 mm. The robot easily traverses most structures in the NIST arena because of his low center of mass. The robot weighs 35 kg and can reach a speed of up to 3.2 km/h. A payload of up to 55 kg can be mounted on the robot. It is able to carry a robotic arm. The original tracks had to be changed because they where not durable enough.

2) Forbot: The second robot is based on the Roboterwerk Forbot, which is a 6-wheel, skid-steered platform and weighs 20 kg.

3) Manipulation: The rescue robot *Forbot* is equipped with a robotic arm manufactured by Schunk called Powerball. The Powerball arm is a 6 DOF robotic arm and can handle up to 6 kg of payload. Currently the team is working on a semi-autonomous manipulation software to perform the shoring task and this can further be used to open doors and to deliver goods. The software is based on the ROS framework MoveIt [1].

4) Sensors for Navigation and Localization: The navigation system is based on multiple different sensors. The measurements are analyzed to estimate the 6D position of the robot and find the best path for exploration.

- Odometry:
  The robots have built in wheel encoders. They are used to calculate current velocity of each track of the robots to perform low level motion control. The wheel encoders are used in conjunction with other measurements for localization.
- Laser Range Finder:
  In order to generate a map and to perform local obstacle avoidance the laser range finder (LRF) sensor Hokuyo UTM-30LX with maximum range of 30 m is used. The LRF is part of the so called laser alignment system. It allows to align the laser scan in 2 dimensions. The team has build an aluminum construction comprising two smart servo motors (Robotis Dynamixel RX-24F) that provide roll and pitch joints for the alignment of the laser scan. Using the laser alignment system ensured to receive consistent and reliable measurements from the laser scanner. Details on the laser alignment system and on experimental evaluation can be found in [2].
- RGB-D Camera:
  Both robots are equipped with the Asus Xtion. The sensor is mounted on pan-tilt unit also called sensor head system which offers better coverage of the area independent of the robot position. The RGB-D sensor is used for victim detection, distance measurements, obstacle detection and providing visual information of the environment. It is also feed into the elevation mapping algorithm to find drivable ground.
- Inertial Measurement Unit:
  For gaining information of robot rotations, the inertial measurement unit (IMU) from XSense version MTi is used. It gives precise information on robot roll, pitch and yaw rotations. The information is used for horizontal alignment of the laser range finder.

5) Model Based Diagnosis: Dependability and robustness are major issues in safety-critical applications such as urban search and rescue. Therefore, *Wowbagger* includes a hardware diagnosis board controllable by ROS-based robot systems. This board is designed to monitor the power consumption of individual system parts. It can measure current and voltage of each branch, and it is possible to turn each branch on or off. Using such a board allows to remotely initiate a power down/up cycle on different parts of the robot, e.g. sensor, base. Therefore, the robot is able to hard reset a crashed sensor autonomously.

B. Software

As an operating system, Ubuntu 14.04 and ROS Indigo are used, due to their proven stability and wide range of package availability. The full list of external software used can be seen in table VII.

1) Map generation/printing: Maps are generated using the Simultaneous Localization And Mapping (SLAM) open-source ROS package hector_slam [3]. The package provides a fast implementation of a solution to the SLAM problem based on laser scan measurements. It does not depend on odometry measurements and works very robustly in the RoboCup rescue environments.

Arena features and found victims are stored and tracked by a self-made object detection system. It handles autonomously found or manually added features during the run. The system is described in more detail in Section II-B2.

The map, detected objects, and found victims are stored as a GeoTIFF file using the hector_geotiff node. Special plug-ins extend the node to connect it to the object detection system.

2) Sensors for Victim Identification: The object detection system was developed to reliably detect victims, regions with higher risk and other relevant objects in the rescue arena. It is able to handle different sensor sources. Moreover, it deals with uncertainties in the measurements and the position estimation using filter approaches [4]. In real disaster areas the rescue robot is used to reduce risks of human rescue teams and increase their performance. The robot's ability to detect different objects and record them in a map plays an important role. With this knowledge, rescue forces are better able to prepare themselves and use appropriate equipment to rescue victims.

The object detection system comprises the components feature detector, feature registration and tracking, object fusion and object classifier.

- Feature Detection:
  The first step in the object identification process is the feature detection. The output of different sensors is analyzed to detect different signs of life and other interesting properties of objects, called features. The position of each detected feature is estimated and a scoring is computed. Currently feature types based on vision, thermal-image, depth-image and gas are supported.
  In order to cover the area of the arena in a smart way the robot uses an algorithm to focus on objects of interrest by using the sensor head. The custom made sensor head has two degrees of freedom (DOF) to extent the field of view of the robot. It consists of two powerful smart servo motors from Robotis (Dynamixel RX-64). This pan-tilt unit can also be moved manually by the operator using a game-pad.
- Vision:
  The rescue robot is equipped with the RGB-D sensor Asus Xtion, placed on top of the sensor head. The images are analyzed to detect objects of interest like hazmat signs, QR codes and victim related features like faces and skin color. Hazmat labels are detected using color histograms and SURF matching as described in [5]. When the robot is further evaluating one image e.g. for deciding if a victim is in sight or not, an external deep convolutional neural network is triggered. This CNN was trained with over 1500 victim images and based on the pretrained Overfeat CNN [6].
- Thermal Image:
  Thermal images are used to detect the body temperature of victims. To measure heat the thermal camera Mirco-Epsilon TIM-160 (a small thermal camera, which allows to measure exact temperature) is used. The images from the thermal camera are analyzed to detect heated areas. The position and the size of the detected heat source is computed using the knowledge about the structure of the environment defined by the mapping algorithms.
- Depth Image:
  The RGB-D sensor Asus Xtion collects 3D information about the environment. This information is used to analyze the structure of the arena. The detection of voids inside the arena allows the robot to improve the search for buried people. In addition the depth image is used to detect movements from potential victims and to build up 3D map.
- Gas Measurements:
  Gas measurements are taken by the Dynament RN 6894A gas sensor. It is able to detect carbon dioxide, propane and methane. Gas measurements are used for the life-sign detection and to identify highly explosive regions. The ability to measure different gases makes the detection of a human victim more accurate. Therefore, the gas sensor measures over the whole operational time of the robot. This allows the robot to detect higher gas concentrations in the area. In relation to certain limits the system is capable to identify highly explosive regions, regions with a too low oxygen level and indications for human respiration.
- Feature Registration and Tracking:
  In the second step all detected features are registered and tracked over time [7]. Every detected life sign is assigned to an already existing feature track or added as a new track. The score, position, size, volume and likelihood of the feature track is updated. The feature tracker is designed to deal with different types of environment models including moving or static features. Due to the fact that victims are not moving in the RoboCup Rescue arena the tracker is configured to assume static features.
- Feature Fusion and Object Classification:
  The last step is the feature fusion and object classification. The feature fusion combines nearby feature tracks to objects based on their scoring and following a few simple rules. Some feature types (e.g. heat) can be shared among multiple objects, other feature types (e.g. face) can only belong to a single object.
  The combined objects are analyzed in the object classification. Based on its feature tracks each object is assigned to one of the defined types. The output of this final computation are different types of objects. This allows the system to distinguish between objects which should be stored autonomously and signed into the map (e.g. qr codes), detections which should be examined in detail (e.g. heated-void) and targets that should be reported to the operator (e.g. victims, dangers). Victims are a collection of different life signs. The operator interface is able to report and visualize found features and objects.

3) Model Based Diagnosis: To guarantee dependability and robustness the robot uses different observers and a diagnosis engine for determining errors and repair plans. Observers are software entities that observe particular properties of the system. A diagnosis model, which is a logic-based description of the systems correct behavior, is used needed for the diagnosis engine. The diagnosis engine combines the observed information and the diagnosis model for model-based reasoning to be able to derive the root cause of problems. The obtained diagnosis together with the observations is used by the repair engine to autonomously recover from failure.

C. Communication

The communication between the operator station and the robot is established using a wireless network. The connection is established between the access points on the robot and the operator's computer. The used hardware is able to work within the 2.4 GHz and the 5 GHz frequency range. The properties of the communication can be seen in table IV.

D. Human-Robot Interface

The Human-Robot Interface (HRI) is used for teleoperation and supervised control of the robot. The interface is based on the RQT visualization framework. This ROS framework allows to easily load different plug-ins to customize a clear user interface. The visualization includes the map, the robot state and the position of objects of interest. In addition two image streams are visualized, showing the current thermal and RGB camera view. The camera streams are overlaid by additional perception information, this is helpful to see things faster and to understand what the robot does. Different buttons are provided to interact with the robot and adapt the operation mode from full teleoperation to full autonomy. In full autonomous mode buttons to confirm, ignore or go closer to victims are available. In this mode the robot starts the mission and only reports back if really necessary. In teleoperation mode the operator can use the interface to register victims on the map and directly drive the robot with the game-pad.

III. Application

A. Set-up and Break-Down

The operator station consists of one compact case including operator laptop, game-pad, headset and mouse. This allows for easy transport and fast set-up / break-down of all components. After switching on the laptop, the operator can establish a connection to the rescue robot and start the mission. The operator station can be operated with external power supply or in battery mode. The case includes a power cable to enable an easy connection to external power sources.

B. Mission Strategy

The robot systems are designed to be used in two operational modes. The first mode is the autonomous mode. In real disaster scenarios there is a lag of rescuers and autonomous robots can help saving life where time is crucial. Autonomous robots are more independent and one operator can operate multiple robots in parallel on the disaster site. Second one is the teleoperation mode where multiple robots can be operated by a single operator. Teleoperation is designed to allow the operator to teleoperate multiple robots with the same operator station. The goal was to design a teleoperation that is easy to use even without any knowledge on robotics.

C. Experiments

The robot teleoperation mode was tested by fire fighters, this helped us to understand where usability problems arise. The team performs training for RoboCup competitions in NIST rescue arena in our laboratories. Monthly testing and training days are performed under competition conditions.

D. Application in the Field

An active cooperation with the responders has allways been a central aim of the TEDUSAR project. The exchange between researchers and responders is important to understand the responders requirements on technology and to increase the acceptance of new technologies. As part of this cooperation team TEDUSAR with their robots participated in different emergency respond exercises organized by the responders. During the exercises different impressions of potential practical application of robots and their sensors at real disaster sites are received.

The robot Zaphod participated in the international forest fire exercise at Soboth, Austria in 2012. The exercise was organized by the EU-founded project GOAL [8]. The robot was used to explore the forest and detect remaining blazes as well as smoldering fires.

In 2013 the team participated in a tunnel emergency exercise. The selected scenario was an accident of a minivan transporting hazardous materials and a coach in a tunnel. The exercise was conducted in the street tunnel Loiblpass located at the border between Slovenia and Austria. The task was to provide remote reconnaissance from the disaster site inside the tunnel.

Another tunnel emergency exercise was arranged inside the Plabutsch mountain in 2014. Together with colleagues from the FH Aachen the team performed a tunnel mapping exercise.

With the current project R$\hat{3}$ the team also contributes to the integration of robotics for already established emergency force institutions.

IV. Conclusion

Team TEDUSAR was established as part of a cross-boarder project between Slovenia and Austria. Today the team continues to work in the field of urban search and rescue and participates in RoboCup events as well as in emergency exercises with first responders. In this paper, the current state for *Wowbagger* (the major research robot) and the new *Forbot* robot extended for manipulation, is shown in detail.

In the last years of RoboCup, we learned a lot about software and hardware design. These days it's not difficult to get a robot up and running, but making it reliable is very hard to achieve. We also learned that a good foundation is necessary for further development, otherwise you will always be distracted by problems burried somewhere in the system. Our team is very curious about the changes to the competition and is hoping to see an ongoing improvement of robotics for rescue purposes.

Appendix A
Team members and Their Contributions

- Konstantin Lassnig — Team Leader, Software Design
- Jakob Auer — Arm Control, Manipulation
- Alexander Buchegger — Software Design
- Stefan Imlauer — Software Design
- Stefan Loigge — Mechanical, Electrical, Software Design
- Peter Lorenz — Image Processing
- Clemens Mühlbacher — Software Design

TABLE I
GROUND SYSTEM

Attribute	Value
Name	Wowbagger
Locomotion	tracked
System Weight	40 kg
Weight including transportation case	55 kg
Transportation size	0.8 x 0.6 x 0.8 m
Typical operation size	0.76 x 0.53 x 0.75 m
Unpack and assembly time	30 min
Startup time (off to full operation)	10 min
Power consumption (idle/ typical/ max)	? / 200 / 300 W
Battery endurance (idle/ normal/ heavy load)	? / 120 / 90 min
Maximum speed (flat/ outdoor/ rubble pile)	1 / 1 / - m/s
Payload (typical, maximum)	10/ 20 kg
Support: set of bat. chargers total weight	7.5 kg
Support: set of bat. chargers power	800 W (100-240 V AC)
Support: Charge time batteries (80%/ 100%)	- / 180 min
Support: Additional set of batteries weight	6 kg
Any other interesting attribute	?
Cost	50,000 USD

TABLE II
MANIPULATION SYSTEM

Attribute	Value
Name	Forbot
Locomotion	wheeled
System Weight	? kg
Weight including transportation case	? kg
Transportation size	? x ? x ? m
Typical operation size (Depending on arm pose)	0.7 x 0.5 x 0.9 m
Unpack and assembly time	? min
Startup time (off to full operation)	? min
Power consumption (idle/ typical/ max)	? / ? / ? W
Battery endurance (idle/ normal/ heavy load)	? / ? / ? min
Maximum speed (flat/ outdoor/ rubble pile)	4 / 1 / - m/s
Payload (typical, maximum)	? / ? kg
Arm: maximum operation height	1.15 m
Arm: payload at full extend	? kg
Support: set of bat. chargers total weight	? kg
Support: set of bat. chargers power	325 W (100-240V AC)
Support: Charge time batteries (80%/ 100%)	? / ? min
Support: Additional set of batteries weight	? kg
Any other interesting attribute	?
Cost	? USD

- Julia Nitsch — High Level Control
- Gerald Steinbauer — Advisor

APPENDIX B
LISTS

A. Systems List

Specifications of the robots Wowbagger and Forbot are listed in table I and table II. Information about the operator box and the communication to each robot can be found in table III and table IV.

B. Hardware Components List

Detailed information about the hardware used for Wowbagger are listed in table V, and about Forbot in table VI.

C. Software List

Both robots are using the same operating system and basic software components, see table VII.

TABLE III
OPERATOR STATION

Attribute	Value
Name	
System Weight	4.5 kg
Weight including transportation case	4.5 kg
Transportation size	0.5 x 0.4 x 0.15 m
Typical operation size	0.5 x 0.4 x 0.15 m
Unpack and assembly time	1 min
Startup time (off to full operation)	5 min
Power consumption (idle/ typical/ max)	- / - / 40 W
Battery endurance (idle/ normal/ heavy load)	- / - / 3 h
Any other interesting attribute	?
Cost	2000 USD

TABLE IV
RADIO COMMUNICATIONS USED

Frequency	**Channel/Band**	**Power (mW)**
2.4 GHz - 802.11b/g/n	1-13 / 54-240 Mbit/s	40
5.0 GHz - 802.11a/n	36-54 / 54-240 Mbit/s	40
5.0 GHz - 802.11a/n	100-140 / 54-240 Mbit/s	40

TABLE V
HARDWARE COMPONENTS LIST WOWBAGGER

Part	Brand & Model	Unit Price	Num.
Robot base incl.	mesa robotics element	EUR 30,000	1
Drive motors,		-	
Drive gears,		-	
Drive encoder,		-	
Motor drivers		-	
Batteries		-	4
Battery Chargers		-	1
DC/DC	tracopower	EUR 75	7
Computing Unit	Spectra Mini PC	EUR 1,600	1
WiFi	Netgear R6300	EUR 150	2
IMU	xsens MTi	EUR 1,000	1
Cameras	Asus Xtion	EUR 100	1
Infrared Camera	thermoIMAGER TIM 160	EUR 4,000	1
LRF	HOKUYO UTM-30LX	EUR 4,500	1
CO_2 Sensor	Dynament RN6894A	?	1
Rugged Operator Laptop	Lenovo	EUR 1,600	1

TABLE VI
HARDWARE COMPONENTS LIST FORBOT

Part	Brand & Model	Unit Price	Num.
Robot base incl.	Roboterwerk Forbot	EUR ?	1
Drive motors,			2
Drive gears,			2
Drive encoder,			2
Motor drivers,			1
Battery chargers,			1
Batteries			4
DC/DC converter	tracopower	EUR 75	1
CAN interface	PEAK PCAN-USB	EUR 195	2
Computing Unit	Lenovo laptop	EUR ?	1
WiFi		EUR ?	1
LRF	SICK LMS100	EUR 3,500	1
6-Axis Robot Arm	Schunk Powerball	EUR ?	1
Gripper	Schunk WSG 50	EUR ?	1
Rugged Operator Laptop	Lenovo	EUR 1,600	1

TABLE VII
SOFTWARE LIST

Name	Version	License	Usage
Ubuntu	14.04	open	
ROS	indigo	BSD	
PCL [9]	1.7	BSD	Perception
Hector SLAM [10]	0.3.4	BSD	2D SLAM
Overfeat [6]	?	?	Perception
OpenCV	2.4.10	BSD	Perception

ACKNOWLEDGMENT

The team at their work has received great support from the Graz University of Technology's Institute for Software Technology. Also, several companies have sponsored the team with money and technology.

REFERENCES

[1] S. Chitta, I. Sucan, and S. Cousins, "Moveit!" *IEEE Robotics Automation Magazine*, vol. 19, no. 1, pp. 18–19, 2012.
[2] P. Lepej, J. Maurer, G. Steinbauer, and S. Uran, ""Analysis of Laser Sensor System for a Rescue Robot"," in 21*st International Electrotechnical and Computer Science Conference*, Portoroz, Slovenia, 2012.
[3] S. Kohlbrecher, J. Meyer, O. von Stryk, and U. Klingauf, "A Flexible and Scalable SLAM System with Full 3D Motion Estimation," in *Proc. IEEE International Symposium on Safety, Security and Rescue Robotics (SSRR)*. IEEE, November 2011.
[4] S. Thrun, W. Burgard, and D. Fox, *Probabilistic Robotics*. The MIT Press, 2005.
[5] D. Gossow, J. Pellenz, and D. Paulus, "Danger sign detection using color histograms and surf matching," in *Safety, Security and Rescue Robotics, 2008. SSRR 2008. IEEE International Workshop on*, Oct 2008, pp. 13–18.
[6] "Overfeat cnn," http://cilvr.nyu.edu/doku.php?id=software:overfeat:start.
[7] D. B. Reid, "An algorithm for tracking multiple targets," *Automatic Control, IEEE Transactions on*, vol. 24, no. 6, pp. 843–854, 1979.
[8] "Si-at goal," http://http://www.goal-siat.eu.
[9] R. B. Rusu and S. Cousins, "3D is here: Point Cloud Library (PCL)," in *IEEE International Conference on Robotics and Automation (ICRA)*, Shanghai, China, May 9-13 2011.
[10] S. Kohlbrecher, J. Meyer, O. von Stryk, and U. Klingauf, "A flexible and scalable slam system with full 3d motion estimation," in *Proc. IEEE International Symposium on Safety, Security and Rescue Robotics (SSRR)*. IEEE, November 2011.