The RoboCup Rescue Team Deutschland1

Andreas Nüchter, Kai Lingemann, Joachim Hertzberg,
Oliver Wulf, Bernardo Wagner,
Kai Pervölz, Hartmut Surmann, Thomas Christaller

The RoboCup Rescue competition aims at boosting research in robots and infrastructure able to help in real rescue missions. The task is to find and report victims in areas of different grades of roughness, which are currently indoor. It challenges to some extreme the mobility of robot platforms as well as the autonomy of their control and sensor interpretation software.

In the 2004 competition, the Kurt3D robot was introduced, the first participant capable of mapping its environment in 3D and self-localizing in all six degrees of freedom, i.e., x, y, z positions and roll, yaw and pitch angles. In 2005, we have upgraded the system with more sensors, with a focus on speeding up the algorithms, and we have started to develop a tracked robot platform to cooperate with Kurt3D. This paper gives an introduction to the competition in general and presents main contributions of our Deutschland1 RoboCup Rescue team.

1 Background

RoboCup is an international joint project to promote AI, robotics and related fields. It is an attempt to foster AI and intelligent robotics research by providing standard problems where a wide range of technologies can be integrated and examined. Besides the well-known RoboCup soccer leagues, there is the Rescue league. Its real-life background is the idea of developing mobile robots that are able to operate in earthquake, fire, explosive and chemical disaster areas, helping human rescue workers to do their jobs. A fundamental task for rescue robots is to find and report injured persons. To this end, they need to explore and map the disaster site and inspect potential victims and suspicious objects. The RoboCup Rescue Contest aims at evaluating rescue robot technology to speed up the development of working rescue and exploration systems.

This paper introduces the RoboCup Rescue activities of the Universities of Osnabrück and Hannover and the Fraunhofer Institute of Autonomous Systems (AIS), forming the team *Deutschland1*. The research focuses on automatic 3D mapping and all terrain driving. Both subjects are highly relevant for rescue robots and in combination lead to robots with an enormous application potential. Besides rescue, these applications might include mining and industrial inspection robotics, facility management, architecture, and security tasks.

In this paper, we first explain the RoboCup Rescue contest and sketch the relevant state of the art, focusing on RoboCup Rescue platforms and 3D mapping. Then, we introduce our robots and algorithms used for 3D mapping, followed by a sketch of the recent work on the all terrain vehicle RTS Crawler. Our systems as described here have been on stage at RoboCup Rescue 2005 in Osaka, Japan.

1.1 The RoboCup Rescue Contest

In RoboCup Rescue, rescue robots compete in finding in limited time as many "victims" (manikins) as possible in a given, previously unknown arena and reporting their life signs, situations, and positions in a map of the arena, which has to be generated during exploration. The idea is that this map would, in a real-life application, help humans decide where to send rescue parties. The arena consists of three subareas (yellow, orange, red) that differ in the degree of destruction, and therefore, in the difficulty of traversal. In the "earthquake phase" between competition runs, the areas, including the distribution of the victims, get completely rearranged. Fig. 1 shows some examples.

The robots in RoboCup Rescue are remotely controlled or surveyed by one or more operators. The operator has

Figure 1: Rescue arenas at RoboCup 2004, Lisbon. Top row: Orange and red area. Bottom left: Operator station. Bottom right: Example of a victim in a yellow area.

strictly no direct view of the arena, only transmitted robot sensor data may be used for control. The degree of autonomy or telecontrol in the robots is at the team's discretion.

Scoring is based on an evaluation function that is modified between the competitions. This function incorporates the number of operators (the fewer the better), the map quality, the quality of the victim localization, the acquired information about the victim state, situation and tag, the degree of difficulty of the area, but also penalizes area bumping and victim bumping.

RoboCup Rescue is supported by the American National Institute of Standards and Technology (NIST) to promote mobility and mapping research. Furthermore, the human robot interaction (HRI) is evaluated [10]. Competitions take place annually at the RoboCup World Championship, AAAI, RoboCup American and German Open.

1.2 State of the Art in RoboCup Rescue

Current Rescue robots are quite divergent. The competition presents two main technical challenges: mobility and autonomy. Currently, no team solves both of them satisfactorily; in the 2004 and 2005 competitions, teams have typically focused on tackling one of them. A high degree of autonomous control and mapping typically comes on standard wheeled (though robust) robot platforms, such as iRobot or Activmedia; the focus is then on topics like online SLAM and sensor data interpretation. Due to both their restricted mobility and the limits of autonomous control, these platforms cannot go into the ragged areas. Highly mobile, i.e., all terrain, platforms are normally specially designed. They are normally closely teleoperated, so they can physically and in terms of control score in the very difficult areas.

Highly mobile platforms. The challenge is to have a platform able to cross very irregular ragged surfaces and climb up and down stairs without toppling over, yet able to move delicately in tightly confined space without bumping into objects. Different approaches are being considered, including large wheels and walking machines. However, the current trend is in flexible chain kinematics. A good example is the platform by Toin Pelican [10] (Fig. 2), the winner of the 2004 and 2005 contests. It has two cantilever arms at its front and rear part, which are also tracked. Driving on flat ground, the arms are folded, making the robot shorter and enabling sharp turns. Driving on uneven ground, including stairs, is made possible by unfolding the cantilever arms, nearly doubling the robot length, such that is does not tip over.

SLAM. State of the art for metric maps are probabilistic methods, where the robot has probabilistic motion models and uncertain perception models. Through integration of these two distributions with a Bayes filter, e.g., Kalman or particle filter, it is possible to localize the robot. Mapping is often an extension to this estimation problem. Beside the robot pose, positions of landmarks are estimated. Closed loops, i.e., a second encounter of a previously visited area of the environment, play a special role here: Once detected,

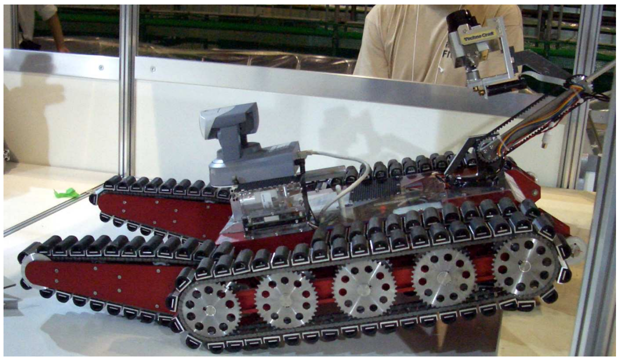

Figure 2: The Toin Pelican robot platform of the Toin University of Yokohama (Japan).

they enable the algorithms to bound the error by deforming the mapped area to yield a topologically consistent model. However, there is no guarantee for a correct model. Several strategies exist for solving SLAM. Thrun [12] surveys existing techniques, i.e., maximum likelihood estimation, expectation maximization, extended Kalman filter or (sparsely extended) information filter SLAM. FastSLAM [13] approximates the posterior probabilities, i.e., robot poses, by particles.

SLAM in well-defined, planar indoor environments is considered solved, and these algorithms are suitable for simple and structured rescue environments, like a yellow arena. Nevertheless, since rescue robots work in unstructured environments, in RoboCup Rescue 2004 and 2005 *none* of these probabilistic methods have been used. Teams with automatic mapping algorithms rely on local scan matching and map integration methods [4, 7]. Robot motion on natural surfaces has to cope with yaw, pitch and roll angles, turning pose estimation into a problem in six mathematical dimensions. In principle probabilistic methods are extendable to 6D. However, to our knowledge no reliable feature extraction mechanisms nor methods for reducing the computational cost of multihypothesis tracking procedures like FastSLAM (which grows exponentially with the degrees of freedom) have been published.

3D Mapping. Instead of using 3D scanners, which yield consistent 3D scans in the first place, some groups have attempted to build 3D volumetric representations of environments with 2D laser range finders. Thrun et al. [13] use two 2D scanners for acquiring 3D data. One laser scanner is mounted horizontally, the other vertically. The latter one grabs a vertical scan line which is transformed into 3D points based on the current robot pose. The horizontal scanner is used to compute the robot pose. The precision of 3D data points depends crucially on that pose and on the precision of the scanner.

A few other groups use highly accurate, heavy 3D laser range finders [6, 11]. The RESOLV project aimed at modeling interiors for VR and telepresence [11]. They used a RIEGL scanner on robots and the ICP algorithm for scan matching [3]. The AVENUE project develops a robot for modeling urban sites [6], using a CYRAX scanner. Nevertheless, in their recent work they do not use laser scanner data in their robot control architecture for localization [6].

Currently the teams of the Centre For Autonomous Systems (Australia), of the Toin University of Yokohama (Japan), of University of Tsukuba (Japan), University of Freiburg (Germany) and out own team Deutschland1 work on 3D mapping [16]. 3D scanners based on the SICK and Hokuyo URG scanner as well as the CSEM Swiss Ranger camera are studied.

2 3D Mapping with Kurt3D

Kurt3D is a mobile robot based on the KURT2 platform. The outdoor version has six 16 cm-wheels, where the two center wheels are shifted sideways/outwards to shorten the overall length of the robot. Two 90W motors are used to power the six wheels. Front and rear wheels have no tread pattern to enhance rotating. The robot has a C-167 microcontroller and two Centrino laptops for sensor data acquisition and sending.

As 3D scanner we are currently using the RTS/ScanDrive developed at the University of Hannover (cf. Fig. 3, middle). The scanning pattern that is most suitable for this rescue application is the yawing scan with a vertical 2D raw scan and rotation around the upright axis. The yawing scan pattern results in the maximal possible field of view (360° horizontal and 160° vertical). The scanner rotates continuously, which is implemented by using slip rings for power and data connection to the 2D scanner. This leads to a homogeneous radial distribution of scan points and saves the energy and time that is needed for acceleration and deceleration of panning scanners. Systematic measurement errors are compensated by sensor analysis and hard real-time synchronization, using a Linux/RTAI operation system. These optimizations lead to scan times as short as 2.3 sec for a yawing scan with 2° horizontal and 1° vertical resolution (181×161 points). For details on the RTS/ScanDrive see [15].

2.1 Scan Registration and Robot Relocalization

Multiple 3D scans are necessary to digitalize environments without occlusions. To create a correct and consistent model (cf. Fig. 4), the scans have to be merged into one coordinate system. This process is called registration. If the robot carrying the 3D scanner were precisely localized, the registration could be done directly based on the robot pose. However, due to the imprecise robot sensors, self-localization is erroneous, so the geometric structure of overlapping 3D scans has to be considered for registration. As a by-product, successful registration of 3D scans relocalizes the robot in 6D, by providing the transformation to be applied to the robot pose estimation at the recent scan point.

The following method registers point sets in a common coordinate system. It is called *Iterative Closest Points (ICP)* algorithm [3]. Given two independently acquired sets of 3D points, M (model set, $|M = \{\mathbf{m}_i\}| = N_m$) and D (data set, $|D = \{\mathbf{d}_i\}| = N_d$) which correspond to a single shape, we aim to find the transformation consisting of a rotation $\mathbf{R}$ and a translation $\mathbf{t}$ which minimizes the following cost function:

$$E(\mathbf{R},\mathbf{t}) = \sum_{i=1}^{N_m}\sum_{j=1}^{N_d} w_{i,j}\,||\mathbf{m}_i - (\mathbf{R}\mathbf{d}_j + \mathbf{t})||^2 . \qquad (1)$$

$w_{i,j}$ is assigned 1 if the i-th point of M describes the same point in space as the j-th point of D. Otherwise $w_{i,j}$ is 0. Two things have to be calculated: First, the corresponding points, and second, the transformation $(\mathbf{R},\mathbf{t})$ that minimizes $E(\mathbf{R},\mathbf{t})$ on the base of the corresponding points.

The ICP algorithm calculates iteratively the point correspondences. In each iteration step, the algorithm selects the closest points as correspondences and calculates the transformation $(\mathbf{R},\mathbf{t})$ for minimizing equation (1). The assumption is that in the last iteration step the point correspondences are correct. Besl et al. prove that the method terminates in a minimum [3]. However, this theorem does not hold in our case, since we use a maximum tolerable distance d_{max} for associating the scan data. Such a threshold is required though, given that 3D scans overlap only partially.

In every iteration, the optimal transformation $(\mathbf{R}, \mathbf{t})$ has to be computed. Eq. (1) can be reduced to

$$E(\mathbf{R},\mathbf{t}) \quad\propto\quad \frac{1}{N}\sum_{i=1}^{N} ||\mathbf{m}_i - (\mathbf{R}\mathbf{d}_i + \mathbf{t})||^2 , \qquad (2)$$

with $N = \sum_{i=1}^{N_m}\sum_{j=1}^{N_d} w_{i,j}$, since the correspondence matrix can be represented by a vector containing the point pairs.

Four direct methods are known to minimize Eq. (2) [8], the following one, based on singular value decomposition (SVD), is robust and easy to implement, thus we give a brief overview of the SVD-based algorithm. It was first published by Arun, Huang and Blostein [1]. The difficulty of this minimization problem is to enforce the orthonormality of the matrix $\mathbf{R}$. The first step of the computation is to decouple the calculation of the rotation $\mathbf{R}$ from the translation $\mathbf{t}$ using the centroids of the points belonging to the matching, i.e.,

$$\mathbf{c}_m = \frac{1}{N}\sum_{i=1}^{N}\mathbf{m}_i, \qquad \mathbf{c}_d = \frac{1}{N}\sum_{i=1}^{N}\mathbf{d}_j \qquad (3)$$

and

$$\begin{aligned} M' &= \{\mathbf{m}'_i = \mathbf{m}_i - \mathbf{c}_m\}_{1,\ldots,N}, &(4)\\ D' &= \{\mathbf{d}'_i = \mathbf{d}_i - \mathbf{c}_d\}_{1,\ldots,N}. &(5)\end{aligned}$$

After substituting (3) and (5) into the error function $E(\mathbf{R},\mathbf{t})$, Eq. (2) becomes:

$$E(\mathbf{R},\mathbf{t}) \propto \sum_{i=1}^{N} \left|\left|\mathbf{m}'_i - \mathbf{R}\mathbf{d}'_i\right|\right|^2, \quad \text{with} \quad \mathbf{t} = \mathbf{c}_m - \mathbf{R}\mathbf{c}_d. \quad (6)$$

The registration calculates the optimal rotation by $\mathbf{R} = \mathbf{V}\mathbf{U}^T$. Hereby, the matrices $\mathbf{V}$ and $\mathbf{U}$ are derived by the singular value decomposition $\mathbf{H} = \mathbf{U}\mathbf{\Lambda}\mathbf{V}^T$ of a correlation matrix $\mathbf{H}$, with a 3×3 diagonal matrix $\mathbf{\Lambda}$ without negative elements. The 3×3 matrix $\mathbf{H}$ is given by

$$\mathbf{H} = \sum_{i=1}^{N}\mathbf{m}'^T_i\mathbf{d}'_i = \begin{pmatrix} S_{xx} & S_{xy} & S_{xz}\\ S_{yx} & S_{yy} & S_{yz}\\ S_{zx} & S_{zy} & S_{zz}\end{pmatrix}, \qquad (7)$$

with $S_{xx} = \sum_{i=1}^{N} m'_{ix}d'_{ix},\ S_{xy} = \sum_{i=1}^{N} m'_{ix}d'_{iy},\ \ldots$ [1].

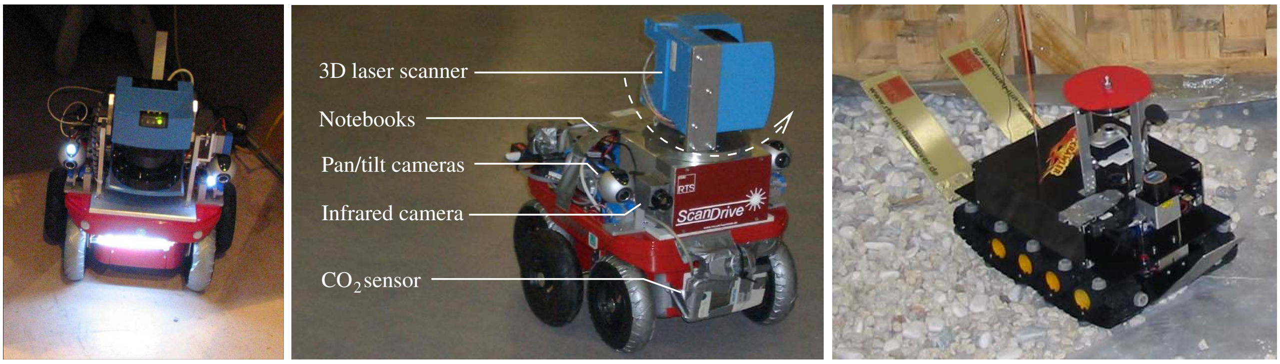

Figure 3: Left: The Kurt3D robot used at RoboCup Rescue 2004 in Lisbon, equipped with a tiltable scanner. Middle: Current Kurt3D robot with RTS ScanDrive. The 3D laser range finder rotates constantly around a vertical axis. Right: RTS Crawler.

2.2 Computing Point Correspondences

The time complexity of the algorithm described above is dominated by the time for determining the closest points (brute force search $\mathcal{O}(n^2)$ for 3D scans of n points). We have implemented kd-trees as proposed by Friedmann et al., using the optimized kd-tree version, i.e., the expected number of visited leafs is kept to a minimum [5].

Since the ICP algorithm extensively computes nearest neighbours, we have proposed a point reduction to reduce the number of nearest neighbour queries and the computing time spend on it. During scanning surfaces close to the scanner are sampled with more data points. These areas are subsampled using a median and reduction filter. Details of the algorithm can be found in [9].

Approximate Range Queries. To gain an additional speedup, approximating the nearest neighbours accelerates the algorithm. S. Arya and D. Mount introduce the following notion for approximating the nearest neighbor [2]: Given an $\varepsilon > 0$, then the point $\mathbf{p} \in M$ is the $(1+\varepsilon)$-approximate nearest neighbour of the point $\mathbf{p}_q \in D$, iff

$$||\mathbf{p} - \mathbf{q}|| \leq (1+\varepsilon) \, ||\mathbf{p}^* - \mathbf{q}|| \, ,$$

where $\mathbf{p}^*$ denotes the true nearest neighbour, i.e., $\mathbf{p}$ has a maximal distance of ε to the true nearest neighbour. Using this notation in every step the algorithm records the closest point $\mathbf{p}$. The search terminates if the distance to the unanalyzed leaves is larger than $||\mathbf{p}_q - \mathbf{p}|| \, / (1+\varepsilon)$.

Semantics-Based Scan Matching. While scanning with the RTS ScanDrive each point gets a supplementary attribute, describing its semantic position in space. 3D points on horizontal surfaces are labeled with the color blue or red, describing their location on the floor or ceiling respectively, while other 3D points, e.g. at walls, are labeled with yellow. The angle of the laser beam to the 3D points is decisive for this simple semantic classification. Fig. 5 shows the user interface for the Kurt3D robot. Its 3D view contains semantically labeled points.

A forest of kd-trees is used to search the point correspondences. For every color, i.e., semantic label, a separate kd-tree is created. The algorithm computes point correspondences according to the label. E.g., points belonging to the wall are paired with wall points of previous 3D scans. Using semantic information helps to identify the correct correspondences, thus the number of ICP iterations for reaching a minimum is reduced. In addition, maximizing the number of correct point pairs guides the ICP algorithm to the correct (local) minimum leading to a more robust algorithm. The influence of the simple semantic interpretation, i.e., horizontally and vertically distributed points to the ICP algorithm is more complex: Points in unknown structures, e.g., our lamps at the ceiling, are labeled as described, leading to exact and correct matches between the structures 'lamp'. Tab. 1 summarizes the computing time for an experiment in an office environment (cf. Fig. 5), pointing out the speed gain as a result of the semantic labeling.

Table 1: Computing time and number of ICP iterations to align two 3D scans (Pentium-IV-3200).

used points	search method	comp. time	iterations
all points	brute force	several hours	42
all points	kd-trees	35.43 sec	42
red. points	kd-trees	2.92 sec	38
red. points	apx. kd-trees	2.22 sec	39
red. points	semantic forest of apx. kd-trees	1.98 sec	34

2.3 Operation of Kurt3D

To cope with the whole rescue arenas, remote control as well as autonomous driving is required. Since the 2005 competition some parts of the area have to be crossed autonomously.

Remote Control. During RoboCup Rescue missions Kurt3D is controlled by an operator. Fig. 5 shows the user interface. The current 3D map (left side) and local virtual 2D scans ([14]) are always presented to the operator. Objects that are preferably used for situation awareness of the operator are walls and obstacle points. The first step to create this virtual 2D scan is to project all 3D points onto the plane by setting the height coordinate to zero. A virtual 2D scan that contains primarily walls can thereafter be assem-

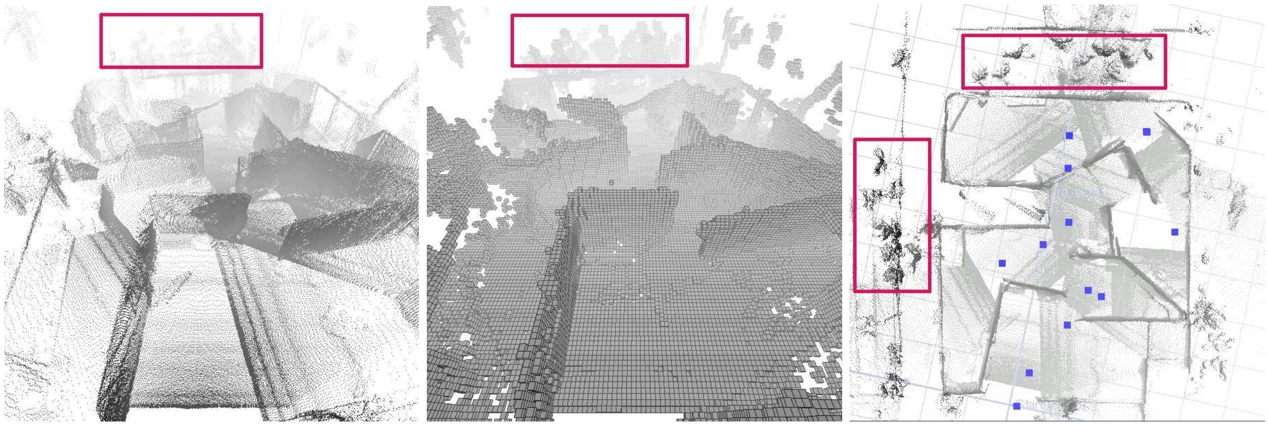

Figure 4: 3D maps of the yellow arena, recorded at the finals of RoboCup Rescue 2004. The 3D scans include spectators that are marked with a rectangle (red). Left: Mapped area as 3D point cloud. Middle: Voxel (volume pixel) representation of the 3D map. Right: Mapped area (top view). The points on the ground have been colored in light grey. The 3D scan positions are marked with squares (blue). A 1 m^2 grid is superimposed. Following the ICP scan matching procedure, the first 3D scan defines the coordinate system and the grid is rotated.

bled by taking one point out of each vertical raw scan. This point is chosen to be the one with the largest distance to the center of the robot. These points represent the outer contour in Fig. 5 right. The points closest to the 3D scanner, not labeled as floor or ceiling, are obstacle points and are shown as inner contour to the operator. The rotation frequency of the scanner is 0.43 Hz. Nevertheless, the operator gets virtual 2D scans with 2 Hz. Odometry is used to fuse the robot pose with the scan data [14].

Autonomous Control. A long-term objective of RoboCup Rescue is autonomous exploration and victim mapping. Autonomous driving is currently based on reactive fuzzy control, steering the robot into free space. Virtual 2D scans as described above are considerably better than ordinary 2D scans, since due to the absence of a fixed 2D horizontal scan plane, 3D objects with jutting out edges are correctly detected as obstacles. Nevertheless one has to think carefully about the distinction between solid obstacles just like walls, and movable objects like crumpled newspapers or curtains that may appear as obstacles in a virtual 2D scan. Autonomous victim detection is currently based on infrared camera data. While driving the robot is scanning the environment for heat sources with a temperature similar to human bodies. When such a heat source is detected, the robot drives autonomously into its direction, stops 80 cm in front of it and informs the operator.

3 The All Terrain RTS Crawler

In order to drive and locate victims in terrain that is not accessable by the wheeled robot, we developed a tracked robot platform to cooperate with Kurt3D.

The RTS Crawler (Fig. 3, right) is a high mobility platform with a size of 40 cm (length) x 26 cm (width) and a total weight of 7 kg including sensors and CPU. The GRP chains are driven by two 60 W DC-Motors. Special rubber knobs give grip on rubble and uneven ground. The crawler is operated either via a 40 MHz remote control or with the onboard embedded PC. To ease remote operation with a delayed feedback a speed controller for each chain is implemented. The platform is equipped with a number of sensors. The sensor most useful for human operation is a CCD camera pointing to an omnidirectional mirror. The image of this omnidirectional camera is distorted and has a relatively low resolution. On the other hand, it gives a good overview of the robot situation and the whole environment. This allows robot operation without camera panning. The 2D laser range sensor Hokuyo URG is used to measure distances to obstacles in the environment of the robot. Thereby it is possible to cope with narrow passages. For future application we intend to mount the sensor on a rotatable unit, acquiring 3D information to contribute to Kurt3D's map building. To navigate safely in uneven terrain and to reduce the risk of flipping over, the robot state is measured with a 3 DOF gyro. All sensor data is captured with an on-board embedded PC and transferred via WLAN to the remote operation terminal.

4 Conclusions and Future Work

The development of robots for urban search and rescue have started just recently. It is an exciting scientific area, including multidisciplinary research such as mechanics, electronics, control theory, artificial intelligence, computational geometry, and computer vision, to name only a few. There is still a great demand for reliable solutions, making RoboCup Rescue an attractive area.

Starting from our research in automatic robotic mapping we joined the community and presented working 3D metric mapping algorithms. The 3D mapping is based on fast 3D scanning in combination with precise registration algorithms. The registration uses ICP scan matching, combined with point reduction and a semantically motivated forest of approximated, optimized kd-tree. The remotely controlled robot Kurt3D provides good situation awareness to the operator by presenting a map and local virtual 2D scans in addition to camera images. Furthermore we briefly covered the

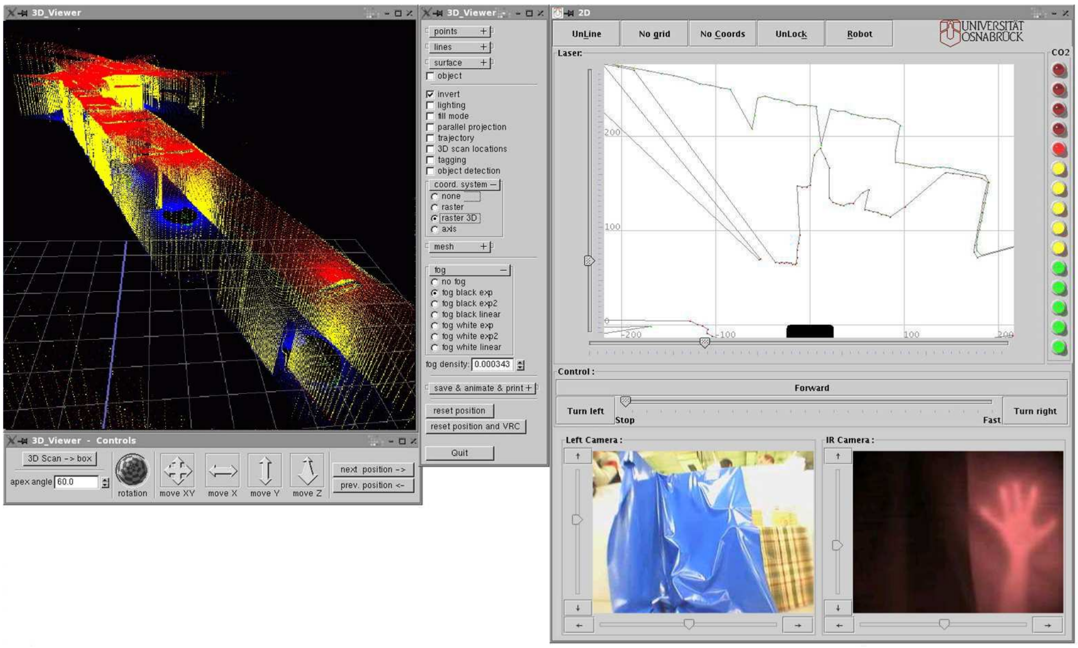

Figure 5: The user interface for controlling the mobile robot Kurt3D. The left part shows the 3D map, with semantically labeled points (blue for floor points, yellow for wall points, red for ceiling points) and the OpenGL controls. The right part shows a local virtual 2D scan, two camera images and the measurement of the CO_2 sensor. The left camera image corresponds to the left pan and tilt camera, the right image can be switched between camera and IR camera. The latter one is able to detect the hand hidden by plastic foil.

current development of our all-terrain platform. As future work we will combine the 3D mapping algorithm with this robot. We also plan to improve the system's autonomy and to change the scan process from the current stop-scan-go fashion to continuous scanning.

References

[1] K. S. Arun, T. S. Huang, and S. D. Blostein. Least square fitting of two 3-d point sets. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 9(5):698 – 700, 1987.

[2] S. Arya and D. M. Mount. Approximate nearest neigbor queries in fixed dimensions. In *Proceedings of the 4th ACM-SIAM Symposium on Discrete Algorithms*, 1993.

[3] P. Besl and N. McKay. A method for Registration of 3–D Shapes. *IEEE Transactions on PAMI*, 14(2), 1992.

[4] St. Carpin, H. Kenn, and A. Birk. Autonomous Mapping in the Real Robots Rescue League. In *Proceedings of RoboCup 2003: Robot Soccer World Cup VII*, 2004.

[5] J. H. Friedman, J. L. Bentley, and R. A. Finkel. An algorithm for finding best matches in logarithmic expected time. *ACM Transaction on Mathematical Software*, 3(3):209 – 226, September 1977.

[6] A. Georgiev and P. K. Allen. Localization Methods for a Mobile Robot in Urban Environments. *IEEE Transaction on Robotics and Automation (TRO)*, 20(5):851 – 864, 2004.

[7] G. Grisetti and L. Iocchi. Map Building in Planar and Non-Planar Environments. In *Online Proc. of the Second International Workshop on Synthetic Simulation and Robotics to Mitigate Earthqake Disaster (SRMED '04)*, 2004.

[8] A. Lorusso, D. Eggert, and R. Fisher. A Comparison of Four Algorithms for Estimating 3-D Rigid Transformations. In *Proc. of the 5th British Machine Vision Conf. (BMVC '95)*, pages 237 – 246, Birmingham, England, September 1995.

[9] A. Nüchter, H. Surmann, K. Lingemann, J. Hertzberg, and S. Thrun. 6D SLAM with an Application in autonomous mine mapping. In *Proc. IEEE ICRA*, USA, April 2004.

[10] National Institute of Standards and Technology. Intelligent Systems Devision, Performance Metrics and Test Arenas for Autonomous Mobile Robots, `http://robotarenas.nist.gov/competitions.htm`, 2005.

[11] V. Sequeira, K. Ng, E. Wolfart, J. Goncalves, and D. Hogg. Automated 3D reconstruction of interiors with multiple scan–views. In *Proc. of SPIE, Electronic Imaging*, 1999.

[12] S. Thrun. Robotic mapping: A survey. In G. Lakemeyer and B. Nebel, editors, *Exploring Artificial Intelligence in the New Millenium*. Morgan Kaufmann, 2002.

[13] S. Thrun, D. Fox, and W. Burgard. A real-time algorithm for mobile robot mapping with application to multi robot and 3D mapping. In *Proc. IEEE ICRA*, USA, 2000.

[14] O. Wulf, K. O. Arras, H. I. Christensen, and B. A. Wagner. 2D Mapping of Cluttered Indoor Environments by Means of 3D Perception. In *Proc. IEEE ICRA*, USA, April 2004.

[15] O. Wulf and B. A. Wagner. Fast 3D Scanning Methods for Laser Measurment Systems. In *Proc. of the Intl. Conf. on Control Systems (CSCS '03)*, Romania, July 2003.

[16] NIST Intelligent Systems Division - Performance Metrics and Test Arenas for Autonomous Mobile Robots. `http://www.isd.mel.nist.gov/projects/USAR/team_info_japan_2005.htm`, August, 2005.

Contact

Prof. Dr. Joachim Hertzberg
University of Osnabrück, Institute of Computer Science
Albrechtstr. 28, D-49069 Osnabrück, Germany
e-mail: `hertzberg@informatik.uni-osnabrueck.de`
WWW: `http://www.inf.uos.de/kbs/`