RoboCup 2008 - homer@UniKoblenz (Germany)

Wolfram Hans, Simon Gräser, Florian Jarmer,
Simon Schmitt, Jan Bornemeier, Patric Lambrecht, David Gossow

Active Vision Group
University of Koblenz-Landau
Universitätsstr. 1
56070 Koblenz, Germany
hans@uni-koblenz.de
http://robots.uni-koblenz.de

Introduction

The *homer@UniKoblenz* is a team of researchers and students from the University of Koblenz-Landau, Germany. Together with its sister team *resko@UniKoblenz*, team *homer@UniKoblenz* will participate in the RoboCup@home league. Our mobile system called "Robbie X" was developed during the last three years. Five practical courses were held in which the robot hardware was assembled and the software developed. The robot was first used to guide persons to various places within the university, then as transport vehicle. After that, it started participating in the RoboCup Rescue league and, since RoboCup German Open 2008, in the @home league.

Fig. 1. Our Pioneer 3 AT robot "Robbie X"

1 Team members and their contributions

- Members of team *homer@UniKoblenz*
  - David Gossow: technical chief designer, object recognition, team leader
  - Simon Gräser: head of project team image processing
  - Wolfram Hans: scientific advisor
  - Ingo Hänlein: head of project team voice recognition and speech synthesis
  - Florian Jarmer: social events, computer setup, programming
  - Simon Schmitt: hardware assistant (mechanics), camera calibration, programming

  - Michael Dahl: hardware assistant (electronics), person tracking, programming
  - Jan Bornemeier: infrastructure, object recognition, feature extraction, programming
  - Patric Lambrecht: public relations, object recognition, programming
- The homer team is supported by the team *resko@UniKoblenz* with the members:
  - Johannes Pellenz: team leader, scientific advisor
  - Susanne Maur: quality management, mapping, programming
  - Robert Hofmann: head of project team rescue
  - Christian Feinen: mapping, exploration, programming
  - Denis Dillenberger: gripper control, programming
  - Marina Trierscheid: NIR camera

Present at the RoboCup competition will be David Gossow, Simon Gräser, Johannes Pellenz, Susanne Maur and Robert Hofmann.

2 Control method and Human-Robot Interface (HRI)

While the robot runs autonomously, a remote GUI runs on the operator laptop (see Fig. 2). The user interface is realized using Qt4 and OpenGL. The principal screen shows displays for various video streams (e.g. the ones provided by the cameras), a 3D-view representing the current sensor measurements and the generated map. Besides this, various dialogues for controlling the robot behaviour can be selected. Other screens are included in the software as well, like a tab used to edit the map

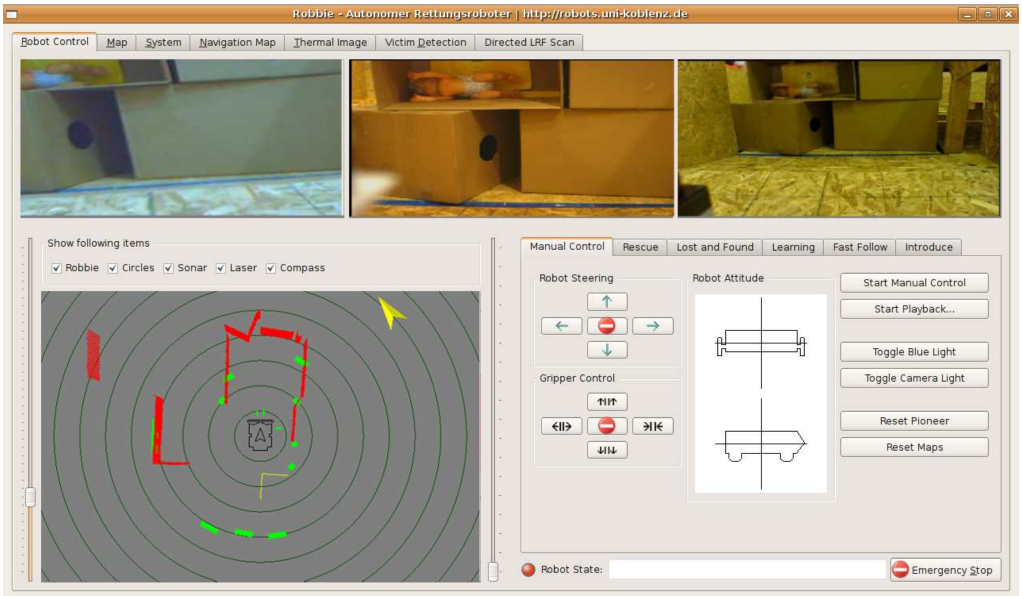

Fig. 2. The user interface of the operator station: Robot control frame

2.1 Rescue

Speech recognition Because the @home-rules forbid to communicate with the robot via mouse, keyboard or buttons, we can interact with Robbie via voice. We decided to use *pocketsphinx*[1] for speechrecognition and *espeak*[2] for speech output.

[1] `http://www.speech.cs.cmu.edu/pocketsphinx/`
[2] `http://espeak.sourceforge.net/`

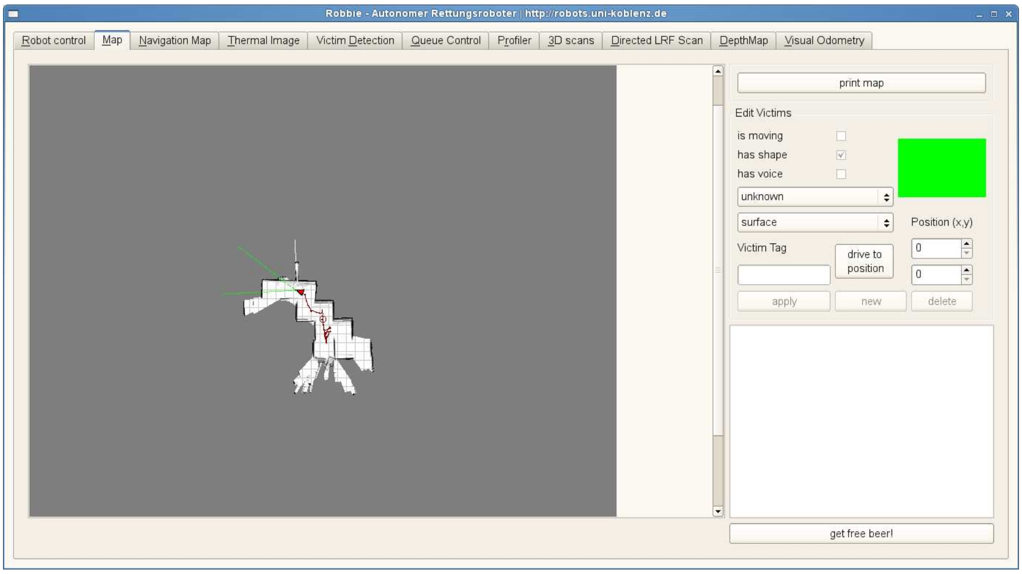

Fig. 3. The user interface of the operator station. Map frame

The incoming data is analysed using the Hidden Markov model (HMM) and feature vectors. The analysed data is compared with the entries in the dictionary and the word with the highest probability is the recognized word hypothesis.

3 Basic technology

3.1 Mapping

While exploring the environment, Robbie can automatically generate a map of the building. The map is based on the laser scans and the odometry information. Fig. 4 shows the result of the mapping process.

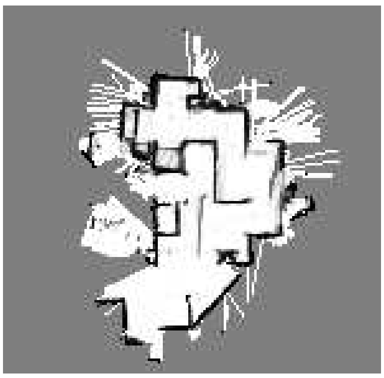

Fig. 4. Map of the RoboCup Rescue yellow arena (generated at the RoboCup German Open 2007)

The map is stored in two planes.One plane counts how often a cell was "seen" by a laser beam and a second plane stores the information how often a cell was seen as occupied. By dividing these two planes, the occupancy probability for a cell c_i is calculated as the following ratio:

$$p_{\text{occ}}(c_i) = \frac{\text{count}_{\text{occ}}(\text{c}_\text{i})}{\text{count}_{\text{seen}}(\text{c}_\text{i})} \tag{1}$$

To solve the SLAM problem, a simple implementation of a particle filter is used [IB98]. We use only 300 particles; each particle represents a hypothesis for the pose of the robot $(x, y, \Theta)^T$. The algorithm of the particle filter includes the following steps: *Resample*, *drift*, *measure* and *normalize*. The result is the most likely pose of the robot while the laser scan was taken. This pose is then used for the *map update*. The resulting map is displayed in realtime on the operator station. The map can be saved to disk. The mapping process is described in detail in [Pel07]

3.2 Object recognition

The object recognition in Robbie is based on histogram backprojection [Pau01] and the SURF (Speeded Up Robust Features) Algorithm [BTVG06] on stereo images in the YUV colorspace to detect and recognize known objects. Therefore the object recognition is divided in four steps:

1. Calculate the histogram backprojection to find regions of interest by color
2. Calculate of SURF-features for these regions
3. Match the SURF-features of the left and right camera image
4. Estimate the distance of the objects by using these correspondences

For the object learning, we take an image of the object in the YUV colorspace, which is supported by Robbies cameras. Each object is described by a two dimensional histogram of the U and V channel and a set of feature vectors from the SURF-algorithm. We choose the UV-channels because it is relatively lighting independend. The learned object features are stored in a file. So it is possible to load them at the game start.

The SURF-Algorithm [BTVG06] resembles Lowes SIFT-algorithm but offers advanced performance and more scalability. The SURF-Algorithm tries to find image correspondences for recognition. It works on an integral images for image convolutions and fast filtering. There are mainly three steps the SURF-algorithm is using.

1. Detect interest points at distinctive locations in an image by using a Hessian matrix-based measure (Interest points are discribed by corners or blobs)
2. Add a reproducible orientation based on information to the detected interest points from a circular region around them which makes the descriptor invariant for rotation

Then, SURF constructs a square region aligned to the selected orientation, and extracts the SURF descriptor from it (descriptor) which is a feature-vector based on haar wavelets weighted with a Gaussian (scale invariant). In the last step SURF matches the extracted feature-vectors of two images. For matching, we use the euclidian distance of the feature vectors. For fast indexing during the matching stage, the sign of the Laplacian (i.e. the trace of the Hessian matrix) for the underlying interest point is included [BTVG06]. This way we can compute scale and rotation invariant properties from objects. To improve object recognition from extreme different views we have extended SURF with a learning algorithm to learn an object from many perspective.

4 Sensors

4.1 Odometry data

The Pioneer robot has a built in system to report the robot pose $(x, y, \Theta)^T$, based on the readings from the wheel encoders. Anyway, our experiments have shown that the data is quite unreliable, and the error sums up over time. Therefore this data can only be used to make a rough guess about the robot location and orientation.

4.2 Sonar sensors

Our Pioneer 3 AT robot platform has two sonar rings (one scanning the front and one scanning the back) with eight sonar sensors each. Totally they cover 360°, but with larger gaps between the sensors. The sonar sensors are used for collision avoidance and the data is displayed in the GUI. The sonars are neither used for mapping nor for localization.

4.3 Laser scanner

The Hokuyo URG04-LX laser scanner generates 240 degree scans that measure the distance (up to 4 meters) of the closest objects. The resolution of the laser scanner is 0.36°. The operator interface shows the scans with respect to the robots position. The scanner is used to build the initial map and for the localization.

4.4 High Resolution Vision System

We have a vision system mounted at the robot's front. The two Sony cameras (one DFW-X700 and one DFW-X710) are adjusted into the direction the robot is moving. A LED light ring makes it possible to search also in dark areas. Additional the color images are used for the object recognition task.

4.5 Extra Camera

In addition to the high resolution vision system we have a single webcam mounted at the robot's front at a very low level. We use this camera to recognize small obstacles that are right in front of the robot or to watch the gripper.

4.6 Thermal Sensors

The two thermal sensors (TPA 81) are used in the RoboCup Rescue league only for detecting victims autonomously by their body heat.

5 Robot Locomotion

The Pioneer 3 AT[3] is a robot equipped with four air-filled tires. They are driven by 4 DC motors (12 volts). We control the robot by using the programming library (called ARIA[4]) provided by the manufacturer. Apart from the autonomous operation, we implemented control via gamepad, via keyboard and via voice control.

6 The Games

6.1 Introduce

Robbie will participate in the *Introduce*-test. The game can be roughly divided into three sections:

1. Driving onto the stage.
2. Introducing himself.
3. Leaving the stage.

We use the stored map and the Points of Interest (POIs) that are set manually for the robot navigation.

[3] `http://www.activerobots.com`
[4] `http://www.activrobots.com/SOFTWARE/aria.html`

6.2 Competetive lost and found

Robbie will also to participate in the RoboCup@home game *Competitive Lost and Found*. The basic task in this game is to find known objects in a partially changing environment. During the game there are two robots in the arena, competing against each other.
This game uses object recognition (see section 3.2) and navigation around moving obstacles. We use SURF [BTVG06] and histograms (U and V component of the YUV color space) for object recognition as explained in section object recognition. If the searched object has been recognized the distance of it is calculated and marked as a POI in the map.

6.3 Fast follow

The game *Fast Follow* is started by a speech-command. For tracking a person, Robbie uses the laser-scan-data and the object recognition for visual detection. Scanmatching is used to overlay two subsequent scans and to detect the motion of the legs. The information of the visual system and the scan data are combined by a Kalman filter, where as well a motion model, depending of the previous motion of the tracked person, is used to archieve higher precision.

6.4 Walk and talk

Another game is *Walk & Talk*. The focus lies on building a map of an unknown area and using this map for localization and navigation to given positions. The game also contains the aspects of interaction via speech recognition, human tracking and object recognition.
In the first phase the robot has to follow a human through an unknown environment while learning a set of objects or places. In the second phase the robot has to navigate to these objects/places.
To enable the Robot to take part in this game we plan to reuse techniques that have already been developed for other games or RoboCup Rescue. We will use the human tracking/following developed for the game Fast Follow, the speech recognition and synthesis that is also used in the other RoboCup@home games and the navigation and map building capabilities.

6.5 Fetch and carry

A optional task is to implement the *Fetch and Carry*-Test. The Fetch and Carry game starts with the object learning. At the game start Robbie awaits a speech-commmand with a position, sets a point of interest in the map and drives to it. Then he starts to grab the object.

The gripper, which is attached at the frontside of Robbie, gives the robot the ability to lift and move objects. It is only possible to take things right from the floor because the limited moving range of the gripper. The gripper can be controlled autonomously using the data from the webcam and the laserrangefinder. For getting the best grabbing position, the laserscanner is used to create a depthmap of the floor in front of the gripper.

References

[BTVG06] Herbert Bay, Tinne Tuytelaars, and Luc Van Gool. Surf: Speeded up robust features. *ECCV*, 2006.

[IB98] Michael Isard and Andrew Blake. Condensation - conditional density propagation for visual tracking. *International Journal for Computer Vision*, 29(1):5–28, 1998.

[Pau01] Dietrich Paulus. *Aktives Bildverstehen*. Der Andere Verlag, Osnabrück, 2001.

[Pel07] Johannes Pellenz. Robocup 2007 - robocup rescue team resko (germany). Technical report, Universität Koblenz-Landau, 2007.

Document date: 26.03.08