LUHbots

Robocup@work 2015 Team Description Paper

Torben Carstensen (team leader), Simon Aden, Jan Carstensen, Andrej Dick, Johannes Gaa, Jens Hübner, Daniel Kaczor, Viet Nguyen, Sven Krause, Johann Wigger

Leibniz Universität Hannover
Mechatronik Zentrum Hannover
Appelstraße 11, 30167 Hanover, Germany
Email: info@luhbots.de
Web: http://www.luhbots.de

Abstract. This paper presents the LUHbots team and current hardware and functional architecture for the KUKA youBot robot. We describe the underlying software framework and the developed capabilities required for operating in industrial environments including features such as robust manipulation, navigation and object recognition.

1 Introduction

The LUHbots team has been founded in 2012 consisting of Diploma, Bachelor and Master Students of the faculty of mechanical engineering at the Leibniz Universität Hannover. Some of the team members have participated in the research inspired practical lecture RobotChallenge. Others started working on the robot in the context of various research projects. RoboCup@work is an opportunity for us to combine and extend the knowledge we gained from the work each one of us put into the lecture or his or her individual research topic. Over the last 3 years we accomplished tasks such as:

- remote control of the robot utilizing a space navigator with smart collision detection, based on the laser sensor
- basic object recognition by means of color, shape and features
- simple localization in a rectangular, known area, which is established by image processing of laser sensor data
- manipulation of simple objects (e.g. balls, cubes) and more complex objects such as bottles and RoboCup@work objects
- localization and mapping on the basis of particle filters
- motion planning considering static and dynamic obstacles
- autonomous object manipulation in known areas

2 The Robot Platform

Our robot is based on the mobile robot KUKA youBot. The robot consists of a platform with four omnidirectional wheels and five degrees of freedom (DoF) manipulator with an attached one DoF Gripper. They are equipped with two commercial laser range finders (hokuyo urg-04lx-ug01) at the platform's front and

back and a RGB-D camera mounted to the wrist of the manipulator. The standard internal computer of the youBot has been replaced by an Intel Core i7 processor. In addition, the robot is equipped with an emergency stop system, allowing for keeping the platform and the manipulator in the actual pose when activated.

3 Software

ROS in combination with the Linux distribution Ubuntu is used as software platform. 3D image processing is accomplished utilizing OpenCV. The whole range of functionality of the KUKA youBot can be simulated with the simulation software Vrep for testing and development. To simplify and coordinate the software development process, a GIT repository was set up on our local server with access for all team members, students and employees working with the robot. Furthermore we established a wiki as knowledge base and information exchange. The currently deployed ROS distribution is Indigo.

4 Navigation

The navigation is based on the standard ROS navigation stack. At first gmapping algorithms are used in combination with the two laser scanners to create a map of the environment in which the robot is going to move autonomously. With AMCL (Adaptive Monte Carlo Localization) the position of the robot is calculated.
The local and global path planners are improved by developments of the team to optimize the omni-directional movement of the youBot in close paths.

5 Object Recognition

We use the *Creative Senz3D* RGB-D camera as hardware for object recognition, which has two basic advantages over commonly used similar devices. Firstly it works at a very close range and secondly it is relatively small. Since the camera is not intended to be used in high precision tasks, the obtained 3D points are too noisy to be used directly for object recognition. Instead we use the 2D images of the infrared and RGB camera to segment the image, to extract features and then to classify the objects. The 3D points are then used to determine the object's position and orientation.

In order to get a robust vision system that can deal with misdetections and that can memorize detected objects, all detections are clustered using a modified version of DBSCAN. Each Cluster is weighed and after filtering the positions are averaged. Then the Clusters are classified as objects or as false positives and saved. The Observed Model is used to optimize scheduling and grasping.

6 Object Manipulation

During the last year we developed a completely new software system that can be seen as a software development kit (SDK) for manipulation tasks with the youBot. The aim was to facilitate the development of applications for the youBot by providing advanced functionality for the manipulator and the mobile platform combined with user friendly interfaces. Some of the features for the manipulator are: inverse kinematics, path planning, interpolated movement in joint- and taskspace, gravity compensation and force fitting. Features for the mobile platform include incremental movement, collision avoidance and movement relative to the environment. The provided interfaces contain a documented API and a graphical interface for the manipulator.
In the RoboCup we use this software e.g. to grab objects using inverse kinematics, to optimize trajectories and to create fast and smooth movements with the manipulator.

7 State machine

Our state machine is based on SMACH, with is a python library for building hierarchical state machines in ROS. Due to the capabilities of SMACH our state machine is modular and consists of the main components task planning, task execution, navigation and manipulation. The state machine acts as an action server, which sets the goals in navigation and manipulation to accomplish the tasks and receive feedback in case of issues.

8 Video

https://www.youtube.com/user/luhbots