b-it-bots@Work
Team Description Paper

Shehzad Ahmed, Devvrat Arya, Oscar Lima, Arka Mallick, Alexander Moriarty, Ashok Sundaram, Santosh Thoduka, Rhama Dwiputra, Frederik Hegger, Nico Hochgeschwender, Sven Schneider, Gerhard K. Kraetzschmar

Bonn-Rhein-Sieg University of Applied Sciences
Department of Computer Science
Grantham-Allee 20, 53757 Sankt Augustin, Germany
Email: <first_name>.<last_name>@inf.h-brs.de
Web: http://www.b-it-bots.de

Abstract. This paper presents the b-it-bots RoboCup@Work team and its 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 and object recognition.

1 Introduction

The b-it-bots RoboCup@Work team at the Bonn-Rhein-Sieg University of Applied Sciences (BRSU) was established at the beginning of 2012. The team consists of Bachelor, Master and PhD students, who are advised by one professor. The results of several research and development (R&D) as well as Master theses projects are going to be integrated into a highly-functional robot control software system. Our main research interests includes mobile manipulation in industrial settings, omni-directional navigation in unconstrained environments, environment modeling and robot perception in general.

2 Robot Platform

The KUKA youBot is the applied robot platform of our RoboCup@Work team. It is equipped with a 5 DoF manipulator, a two finger gripper and an omni-directional platform. In the front and the back of the platform, two Hokuyo URG-04LX laser range finders are mounted to support robust localization and navigation. A sensor tower on the back platform hosts a ASUS Xtion Pro Live RDB-D camera for common perception tasks in 3D, like scene segmentation and object recognition. Further, a Microsoft color camera is mounted on the gripper for visual servoing purpose. The standard internal computer of the youBot has been replaced by a Intel Core i5 processor in order to perform also high computational perception tasks. Finally, the BRSU youBot is equipped with a custom made gripper (see Figure 1) which allows the robot to grasp a wider range of objects from 10 mm up to 65 mm.

Fig. 1. BRSU hardware configuration based on the KUKA youBot

3 Robot Software Framework

The underlying software framework is based on the ROS, the Robot Operating System. We use its communication infrastructure based on topics, services and actions to pass information between the functional components. The framework also provides interfaces to common hardware devices like laser scanners or cameras. The wide range of various tools are utilized for visualization, testing and debugging the whole system. The currently deployed ROS distribution is Hydro.

4 Navigation

Several components have been developed and integrated to move the robot from one place to another in different situations and even cluttered or narrow environments.

4.1 Map-based Navigation

The map-based navigation is based on the ROS navigation stack which uses an occupancy map together with a global and local path planner. For the local path planner a Dynamic-Window-Approach is deployed which plans and executes omni-directional movements for the robots base. This enhances the manoeuvrability especially in narrow environments. The data from the front and rear Hokuyo laser scanners are considered for (dynamic) obstacle avoidance. The map-based navigation component is used to move to different but predefined locations.

4.2 Base Positioning in Front of a Workspace

To enhance the position and orientation accuracy especially in front of a workspace, a simple and effective approach has been used to align the robot perpendicular and in a certain distance to a workspace. A linear regression model is applied to fit a line to each front laser scan. The resulting orientation and distance to this line is fed forward to a closed loop controller which tries to minimize both "errors" by commanding linear and angular velocities to the robot's base. This component is used to compensate for the insufficient accuracy of the ROS navigation.

4.3 Relative Base Controller

In some situation the robot is required not to move in a global reference frame but in a local frame w.r.t the robot, e.g. for visual servoing tasks. Therefore a relative base controller has been implemented which allows to move the robot's base in directions such as forward/backward, shift to left/right and rotating clockwise/anti-clockwise for a given distance or angle. For this movements only odometry information are considered and can be even executed without a map.

5 Object Perception

For the object recognition subtask a three-stage pipeline (see Figure 2) has been devised which involves data accumulation over several consecutive frames.

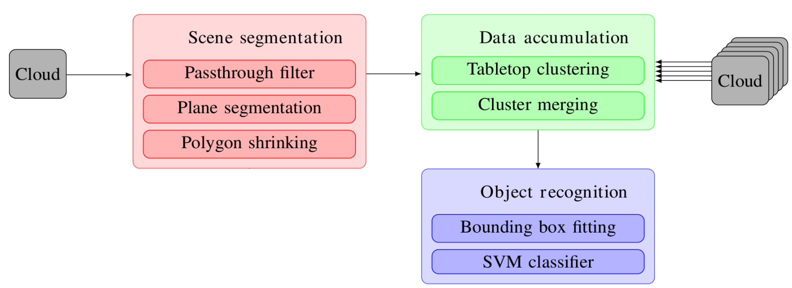

Fig. 2. Object perception pipeline

The first stage is concerned with scene segmentation, or, more precisely, finding the workspace. We capture a single point cloud and apply a passthrough filter to restrict the FOV, which removes irrelevant data and reduces the computational burden. Next we perform plane segmentation with a region-growing algorithm. It is possible that the algorithm outputs more than one planar polygon. In this case we apply orientation constraints to remove irrelevant (e.g. non-horizontal) planes. Among the remaining we select the one with the maximum area. Finally, we shrink the polygon by several centimeters to make sure that it does not include the edge of the workspace.

The second stage is data accumulation. We filter each frame to keep only the points above the workspace polygon, which we then merge into an occupancy octree. Our experiments have shown that 30 frames is a reasonable tradeoff between the running time and the amount of information accumulated.

The final stage is object recognition. We partition the accumulated point cloud into clusters and fit minimal bounding boxes around them. The dimensions of the bounding box, the number of points in the cluster, and the average color of the points serve as an input to a classification neural network, which we have trained beforehand. Based on this information it outputs the predicted object type.

6 Object Manipulation

In order to grasp objects reliably, several components are have been developed and integrated on the robot.

From the object recognition, the position of the desired object is retrieved. Based on this information a Cartesian pre-grasp pose is calculated. The motion is then planned with MoveIt![1], a motion planning framework, based on the youBot model (to avoid self-collisions) and in future also based on sensor information from a 3D camera (to avoid collisions with the environment).

Once the pre-grasp pose is reached, the operation mode is switched to visual servoing. In this stage a blob detection component is used to track the object in the image of the camera mounted on the arm. Approaching the gripper to the object is achieved with a combined arm-/base Cartesian controller which moves both components simultaneous towards the object until it can be grasped.

7 Conclusion

In this paper we presented the modification applied to the standard youBot configuration as well as the functional core components of the current software architecture for the KUKA youBot robot. Besides the development of new functionality, we also focus on developing new and even existing in such away that is robot independent and can be reused for a wide range of other robots with different configuration.

Acknowledgement

We gratefully acknowledge the continued support of the team by the B-IT Bonn-Aachen International Center for Information Technology and the Bonn-Rhein-Sieg University of Applied Sciences.

[1] `http://moveit.ros.org/`