The b-it-bots RoboCup@Work 2012 Team Description Paper

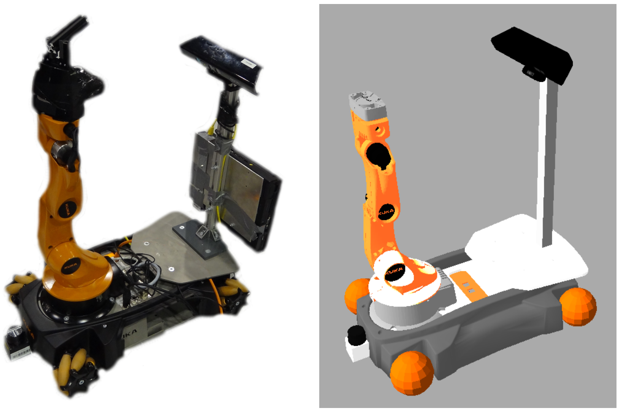
Matt Roscoe, Alexander Moriarty, Sergey Alexandrov, Praveen Ramanujam, Nirmal Giftsun,
Frederik Hegger, Nico Hochgeschwender, Jan Paulus, Gerhard K. Kraetzschmar

Bonn-Rhein-Sieg University of Applied Sciences
Department of Computer Science
Grantham-Allee 20, 53757 Sankt Augustin, Germany
<first_name>.<last_name>@inf.h-brs.de

***Abstract*— This paper presents the b-it-bots RoboCup@Work team and its current 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.**

I. Introduction

The b-it-bots RoboCup@Work team at Bonn-Rhein-Sieg University of Applied Sciences (BRSU) has been established in 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 well-functioning robot control software system. Through this kind of course modules, our RoboCup@Work team is strongly interwoven with the Master by Research course in Autonomous Systems, which is offered at the BRSU[1]. In the first RoboCup@Work competition (RoboCup 2012, Mexico City) our team placed third. Our main research interests includes mobile manipulation in industrial settings, omni-directional navigation in unconstrained environments, environment modeling and robot perception in general. Our approach is to first identify and evaluate in each subfield the state-of-the-art and the best practice solutions currently available, implement and integrate applicable algorithms, and to develop our own custom approaches with a focus on robustness in uncertain industrial environments.



(a) (b)

Fig. 1. The KUKA youBot in real world (a) and a simulated environment (b)

II. Robot Platform

The KUKA youBot[2] is the applied robot platform of our RoboCup@Work team. It is is equipped with a 5 DoF manipulator, a two finger gripper and a omnidirectional platform. In the front and in the back of the platform, two Hokuyo URG-04LX laser range finder are mounted to support robust localization and navigation. A sensor tower on the back platform hosts a Microsoft Kinect camera for common perception task, like scene segmentation and object recognition. Further, a Logitech webcam is mounted on the gripper for visual servoing. The internal computer is supported by a second high performance laptop which performs the computational expensive perception tasks.

[1] http://www.inf.h-brs.de/MAS.

[2] www.http://youbot-store.com

III. Robot Software Framework

The underlying software framework is based on the ROS framework[3]. We basically use its communication infrastructure based on publish/subscribe, service server/client and action server/client to pass information between our components. The framework also provides interfaces to various common hardware devices like laser scanners or cameras. Further, the wide range of various tools is utilized for visualization, testing and debugging the whole system.

Due to the deployment of ROS, we were able to migrate functionality from our Care-O-bot 3 robot to the KUKA youBot with less effort. All functional components are implemented as basic ROS nodes and can be connected to more cognitive capabilities to solve complex tasks. An example of such complex task in industrial robotics is "pick and place" (e.g. “bring a bolt to the production line and screw it on mounting point ’A’"). To perform this task, the robot needs to combine several actions such as navigation, object recognition, and object grasping. The actual scheduling is realized as a finite state machine (FSM). As the number of capabilities of the robot grows, more complex combinations of tasks can be performed. By virtue of this versatility, a scheduling by the developer is not feasible anymore. Hence, we will deploy an additional task planning component as replacement for the current FSM approach.

IV. Object Perception

Perception of objects relevant for the industrial environment is particularly challenging. The objects are typically small and often made of reflective materials such as metal. We use a Microsoft Kinect camera which provides both intensity and depth images of the environment. This enables effective scene segmentation and object clustering. But the spatial resolution is low even at the close range, and a significant degree of flickering corrupts the images. The information captured in a single frame is often not sufficient to reason about the object type. We have therefore devised a three-stage pipeline (see) which involves data accumulation over several consequtive frames.

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 computation burden. We next apply region growing–based plane segmentation. The algorithm may output several planar polygons, among which we pick horizontal ones with the maximum area. Finally, we shrink the polygon by several centimeters to make sure that it does not include the boundary of the workspace.

The second stage is data accumulation. We first filter each new point cloud to keep only the points above the workspace polygon. We then perform Euclidean clustering and create a separate occupancy octree for each cluster, which represents a hypothesis about an object. We then proceed with new frames, merging the new clusters in the appropriate octrees. This process is reapeated for a certain time. Our experiments have shown that 25 frames is a reasonable tradeoff between the time and the amount of information accumulated.

The final stage is object recognition. We analyze the obtained point clusters from the previous stage by fitting minimal bounding boxes around them. The dimensions of the bounding box and the number of points in the cluster serve as an input for a SVM classifier, which we trained beforehand. Based on this information it outputs the object type.

V. Object Manipulation

Industrial environments provide several challenges to grasping objects using a two-fingered grasper. The orientation and placement of objects is not guaranteed. An inverse kinematic based approach to grasp an object may not always find a solution for a given pose. But will almost always bring the arm close enough to the object to grasp it with a few minor tweaks. To deal with these difficulties we employ a solution that relies on visual servoing. Once the arm has been guided to the rough location of the object we use blob detection to find and track the object. Once we know where the object is located in relation to the arm, the system will then adjust the position of the base of the robot to better align the arm and gripper. To deal with possible rotational offsets the system will also align the object so that the gripper will be properly aligned to safely grasp the object. Once the object has been properly aligned the arm will continue with the procedure to grasp the object.

VI. Current and Future Research Activities

- Omni-directional path planning and execution
- Object tracking
- Arm motion planning
- Grasp planning
- Task planning

[3] http://www.ros.org

VII. Conclusion

In this paper we presented the functional core components of its current software architecture for the KUKA youBot robot. Beside the development of new functionality, we also focus on porting existing components from our Care-O-bot 3 robot to the youBot platform. The migration from one robot to another was and is still an exhaustive exercise. In our current EU FP7 funded project BRICS (Best Practice in Robotics)[4] we are exploring first steps towards an improved software development methodology in robotics. Among others we applied the component-oriented development approach defined in BRICS for our software development which turned out to be very feasible when several heterogenous components are composed into a complete system.

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. In addition, the research leading to these results has received funding from the European Community's Seventh Framework Programme (FP7/2007-2013) under grant agreement no. FP7-ICT-231940-BRICS (Best Practice in Robotics).

[4] www.best-of-robotics.org