UvA@Work, Team Description Paper RoCKIn Camp 2015 - Peccioli, Italy

Valerie Scholten, Victor Milewski, Tessa Bouzidi, Celeste Kettler
and Arnoud Visser (a.visser@uva.nl)

Intelligent Robotics Lab, Universiteit van Amsterdam, Science Park 904, Amsterdam, The Netherlands

http://staff.fnwi.uva.nl/a.visser/research/atwork

I. Introduction

The UvA@Work team consists of four Artificial Intelligence (AI) bachelor students supported by a senior university staff member. It was founded in 2013 as part of the Intelligent Robotics Lab[1]. This initiative strives to involve students of the Universiteit van Amsterdam (UvA) in robotics. It also acts as a governing body for all the UvA's robotic teams such as UvA@Work, The Dutch NAO Team [1] (RoboCup Standard Platform League), the Amsterdam Oxford Joint Rescue Forces [2] (RoboCup Rescue Simulation League) and Dutch UAS (Wildlife Conservation UAS Challenge) to guarantee continuation of research, education and competition experience. It also enables sharing any gathered knowledge between the different teams.

After its first competition experience with the KUKA youBot at the RoCKIn Camp 2014 in Rome, the UvA@Work team is looking forward to participate in the RoCKIn Camp 2015 in Peccioli.

[1] http://www.dutchnaoteam.nl/index.php/irobolab/

II. Hardware

The KUKA youBot used by the UvA@Work team is equipped with Festo FinGrippers and a sensor suite consisting of an Asus Xtion Pro Live to create a disparity map, a Hokuyo UTM-30LX laser scanner for occupancy grid mapping and a Microsoft LifeCam HD-5000 for RGB vision. The Asus Xtion Pro is mounted well above the youBot platform to allow for a broad view of the local surroundings. These broad overviews can be processed as both, RGB and depth images. The Hokuyo UTM-30LX laser scanner is mounted in front of the youBot platform and provides a high precision depth plane over a 270 degree angle. Finally, the Microsoft LifeCam HD-5000 is mounted on top of the youBot's end effector to assist pick-up and placing tasks.

III. Software

A. Robot Operating System (ROS)

ROS is a broadly used framework for robotic application development. It supplies an easily extensible environment of basic components (nodes) which can be combined flexibly to form applications. ROS also provides a parameter server which allows nodes

to use shared parameters. In the current implementation these shared parameters are used to change the local workings of each node depending on the task that needs to be carried out. Furthermore, ROS comes with a range of packages, libraries, drivers and simulation programs that simplify the use of standard platform robots. For the development of implementations on the KUKA youBot, the team of UvA@Work uses the ROS Hydro Medusa version, published in public repository, as described in section III-G.

B. Image Processing Pipeline

The Image Processing Pipeline (IPP) currently consists of 7 nodes: webcam_controller, openni2, image_controller, image_thresholder, datamatrix_detector, tag_detector and object_detector.
The webcam_controller takes care of retrieving image streams from the different RGB cameras mounted on the robot. In the current implementation, only one camera can be used at a time due to limitations in data capacity of the USB-bus. To work around this limitation, the active camera can be assigned depending on the current task by using ROS parameters. For depth cameras the standard ROS package Openni2 is used. These two packages send their images to the image_controller node. Depending on the ROS parameters specified, the received input is passed on to the right detector and/or image_thresholder node. The detectors are used to locate data matrices, tags (colored sheets) or objects (as specified in the rulebook) in the image frame. These detectors also rely on the use of the OpenCV2 library. Finally, the detectors send obtained features to the action_controller.
By using this pipeline, each node can operate on a separate thread. Therefore the current implementation is only limited by the image throughput of the cameras used.

C. Manipulation Pipeline

Once localization and navigation tasks are completed, the next step is to fulfil the manipulation challenge. This could be picking up and precisely placing one of the industrial parts of the RoboCup and RoCKIn@Work challenges.

A prerequisite for pick-up is that the transformational function between perception-space and configuration-space of the robot arm is known. To obtain it, the Move-It module of ROS is extended with a path-planning algorithm developed by Dorst *et al* [3], preferable an anytime incremental variant like [4].

The advantage of using a path-planning algorithm in the configuration space is that evading obstacles in the environment becomes native in the planning. The challenge with this method is the transfer of real-world coördinate points to the configuration-space of the robot. Once the Inverse Kinematics of the KUKA youBot is calculated, the obstacles in the real world can be mapped onto the configuration-space. After this, the path-planning in the configuration space becomes a trivial problem that can be tackled by several search-algorithms like D*-lite [5] or Dorst's algorithm [3].

D. Inverse Kinematics

To solve the inverse kinematics, the configuration of the 7 joints of the youBot arm is now defined as a set of homogeneous transformations. Each joint represented with a translation matrix and optionally a rotation matrix that represents the rotation of the joint over the specified angle θ_i. The forward kinematics chain to compute the position of the gripper tip relative to the origin can be calculated in the following way:

$$\begin{aligned} T_0 = {} & J1T \cdot J2T \cdot J2R(\theta_2) \cdot J3T \\ & \cdot J3R(\theta_3) \cdot J4T \cdot J4R(\theta_4) \cdot J5T \\ & \cdot J5R(\theta_5) \cdot J6T \cdot J6R(\theta_6) \cdot J7T \end{aligned}$$

The inverse kinematics requires to find the five joint angles θ_i which correspond with a certain location T_0. The translation and rotation matrices are given in the appendix.

E. Human Robot Interaction (HRI)

In context of the RoCKIn Camp 2014 in Rome, the UvA@Work team started working on a Spoken Language Understanding (SLU) application for optimized customer-robot interaction. As SLU allows for a native communication with the robot, Human Robot Interaction is tailored for real-word use.

Spoken commands are recorded by an attached microphone and sent to the online Google voice recognition service. The returning result in form of written words is fed to a language model system, tagging words and applying a grammatical structure. As a next step, contained objects and actions are grounded in reference to a prebuilt database of environment and robot. These grounded objects and actions are than used to create a series of commands that can be send to the task_manager node within the ROS framework - thus conversed into a language that is "understood" by the robot. Tested in RoCKIn Camp Rome were basic directional commands as well as emergency "STOP".
This SLU interface can also be used as an alternative to the standard command-line @Work interface used to specify the task to be carried out.

F. Action Manager

The action manager takes care of all movement required by the KUKA youBot. It does so by listening to the task manager. All actions are partially depended on the current state of the robot. Therefore the action manager also keeps an updated version of the current joint states.

G. Reusability of the Software

During development of the code, UvA@Work focused on creating task-specific nodes that can be used as independent modules in the ROS framework. On top of that, various top-layer instances were created to centrally manage and control the modules. All information streams are assigned to unique default topics that can be subscribed to by other nodes where necessary. By doing so, modularity of code and re-usability of modules can be guaranteed.
The code of the UvA@Work team is available at github[2] where the reusable task modules will be made publicly available, following the initiative of SwarmLab[3].

[2]https://github.com/IntelligentRoboticsLab/KukaYouBot
[3]https://github.com/swarmlab/swarmlabatwork

IV. Research Areas

The University of Amsterdam has been active in the RoboCup since 1998 in form of varying teams such as the Windmill Wanderers [6], Clockwork Orange [7], Dutch Aibo Team [8], the Amsterdam Oxford Joint Rescue Forces [9] and the UvA Rescue Team [10].

Since its establishment in September 2013, the UvA@Work team has participated in the RoCKIn@Work Camps 2014 workshop in Rome, Italy [11]. This workshop offered both lectures and practical sessions on various topics such as navigation, object manipulation and human robot interaction. In the latter case, they have been awarded the title: "Best practical demonstration on human robot interaction."

A. Intelligent House Building

The new team members of UvA@Work have worked with the KUKA youBot in the first year of their study. They used the youBot to study how difficult it is to use a robot as construction device [12]. The result of their effort was that they were able to program the robot to construct two walls from construction material (see Fig. 1) with open loop control.

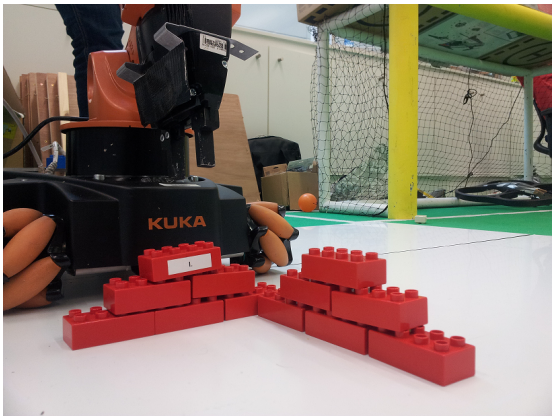

Figure 1. The result of the construction operation of the KUKA youBot.

A recording of the construction operation is available on YouTube[4]. As can be seen from this

[4]https://www.youtube.com/watch?v=M7llxA31NxY

recording, there is some variation in the placement of the material. This variation can only be reduced by introducing a closed loop based on torque or visual sensors, like the new version of the Microsoft LifeCam or the Asus Xtion Pro.

V. Task description

There are three tasks that can be performed at RoCKIn camp 2015. Out of these three the aim is to do both the tasks named Fill a Box with Parts for Manual Assembly as well as the task Prepare Assembly Aid Tray for Force Fitting. These tasks are more comparable to our previous work then the task about plate drilling. Because the robot is equipped with a Microsoft LifeCam HD-5000, the aim is to recognize the objects and the destination of the robot by RGB-vision. In a grid, with on one end the robot and on the other end the destination, there needs to be a search for objects. Hopefully we will be able to integrate found objects on the map generated by the laser scanner. With the A*-algorithm an optimal path without obstructions can be found. This way the robot can get to his destination as quick as possible. When it has arrived the robot puts the plate in its rightful place. This can be done with a computer-vision based approach.

The second task, Prepare Assembly Aid Tray, is an extension of the first task. The first thing to consider are the paths from the start point to the products that need to be assembled. The shortest path will be the first product our robot is going to fetch. When the robot has arrived and has localized the object with RGB-vision, it will pick this product up and put it on his platform. Then the robot calculates the paths to the rest of the products and again picks the shortest path. With use of the Hokuyo UTM-30LX laser scanner the environment is constantly scanned so even dynamic obstacles can be avoided.

VI. Conclusion

The RoCKIn Camp, as a weeklong hands-on school in robotics , will be a great opportunity to boost both the knowledge and experimental skills of our students. In addition, it will be great to share ideas with teams from other European countries. The expected result will be a large improvement of the performance of our system to recognize and manipulate objects in an industrial setting.

References

[1] P. de Kok, N. Girardi, A. Gudi, C. Kooijman, G. Methenitis, S. Negrijn, N. Steenbergen, D. ten Velthuis, C. Verschoor, A. Wiggers, and A. Visser, "Team description for RoboCup 2013 in Eindhoven, the Netherlands," Dutch Nao Team, Universiteit van Amsterdam & TU Delft, May 2013.

[2] N. Dijkshoorn, H. Flynn, O. Formsma, S. van Noort, C. van Weelden, C. Bastiaan, N. Out, O. Zwennes, S. S. Otárola, J. de Hoog, S. Cameron, and A. Visser, "Amsterdam oxford joint rescue forces - team description paper - virtual robot competition - rescue simulation league - RoboCup 2011," Universiteit van Amsterdam & Oxford University, July 2011.

[3] L. Dorst, I. Mandhyan, and K. Trovato, "The geometrical representation of path planning problems," *Robotics and Autonomous Systems*, vol. 7, no. 2, pp. 181–195, 1991.

[4] M. Phillips, A. Dornbush, S. Chitta, and M. Likhachev, "Anytime incremental planning with e-graphs," in *Robotics and Automation (ICRA), 2013 IEEE International Conference on*. IEEE, 2013, pp. 2444–2451.

[5] S. Koenig and M. Likhachev, "D* lite." in *AAAI/IAAI*, 2002, pp. 476–483.

[6] E. Corten and E. Rondema, "Team description of the windmill wanderers," in *Proceedings on the second RoboCup Workshop*, 1998, pp. 347–352.

[7] P. Jonker, B. van Driel, J. Kuznetsov, and B. Terwijn, "Algorithmic foundation of the clockwork orange robot soccer team," in *Algorithmic Foundations of Robotics VI*. Springer, 2005, pp. 17–26.

[8] A. Visser, J. Sturm, P. van Rossum, J. Westra, and T. Bink, "Dutch aibo team: Technical report robocup 2006," December 2006.

[9] V. Spirin, S. Soffia Otárola, and A. Visser, "Amsterdam oxford joint rescue forces - team description paper - virtual robot competition - rescue simulation league - robocup 2014, joão pessoa - brazil," 2014.

[10] M. Traichioiu and A. Visser, "Uva rescue - team description paper - agent competition - rescue simulation league - iran open 2014," 2014.

[11] S. Negrijn, S. van Schaik, J. Haber, and A. Visser, "UvA@Work - RoCKIn@Work camp 2014 - rome, italy," Team Description Paper, Intelligent Robotics Lab, Universiteit van Amsterdam, The Netherlands, November 2013.

[12] V. Scholten, V. Milewiski, T. Bouzidi, and C. Kettler, "Intelligent house builder," Project Report, Universiteit van Amsterdam, June 2014.

APPENDIX: THE TRANSLATION AND ROTATION MATRICES OF THE KUKA YOUBOT ARM

The forward kinematics chain to compute the position of the gripper tip relative to the origin can be calculated in the following way:

$$\begin{aligned} T_0 = {} & J1T \cdot J2T \cdot J2R(\theta_2) \cdot J3T \\ & \cdot J3R(\theta_3) \cdot J4T \cdot J4R(\theta_4) \cdot J5T \\ & \cdot J5R(\theta_5) \cdot J6T \cdot J6R(\theta_6) \cdot J7T \end{aligned}$$

These are the definitions of each of the matrices.

$$J1T = \begin{bmatrix} 1 & 0 & 0 & 0.143 \\ 0 & 1 & 0 & 0.0 \\ 0 & 0 & 1 & 0.046 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$J2T = \begin{bmatrix} 1 & 0 & 0 & 0.024 \\ 0 & 1 & 0 & 0.0 \\ 0 & 0 & 1 & 0.096 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$J2R(\theta_2) = \begin{bmatrix} \cos(170^o - \theta_2) & -\sin(170^o - \theta_2) & 0 & 0 \\ \sin(170^o - \theta_2) & \cos(170^o - \theta_2) & 0 & 0 \\ 0 & 0 & 1 & \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$J3T = \begin{bmatrix} 1 & 0 & 0 & 0.033 \\ 0 & 1 & 0 & 0.0 \\ 0 & 0 & 1 & 0.019 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$J3R(\theta_3) = \begin{bmatrix} \cos(\theta_3 - 65^o) & 0 & \sin(\theta_3 - 65^o) & 0 \\ 0 & 1 & 0 & 0 \\ -\sin(\theta_3 - 65^o) & 0 & \cos(\theta_3 - 65^o) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$J4T = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0.0 \\ 0 & 0 & 1 & 0.155 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$J4R(\theta_4) = \begin{bmatrix} \cos(146^o + \theta_4) & 0 & \sin(146^o + \theta_4) & 0 \\ 0 & 1 & 0 & 0 \\ -\sin(146^o + \theta_4) & 0 & \cos(146^o + \theta_4) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$J5T = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0.0 \\ 0 & 0 & 1 & 0.135 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$J5R(\theta_5) = \begin{bmatrix} \cos(\theta_5 - 102.5^o) & 0 & \sin(\theta_5 - 102.5^o) & 0 \\ 0 & 1 & 0 & 0 \\ -\sin(\theta_5 - 102.5^o) & 0 & \cos(\theta_5 - 102.5^o) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$J6T = \begin{bmatrix} 1 & 0 & 0 & -0.002 \\ 0 & 1 & 0 & 0.0 \\ 0 & 0 & 1 & 0.130 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$J6R(\theta_6) = \begin{bmatrix} \cos(167.5^o - \theta_6) & -\sin(167.5^o - \theta_6) & 0 & 0 \\ \sin(167.5^o - \theta_6) & \cos(167.5^o - \theta_6) & 0 & 0 \\ 0 & 0 & 1 & \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$J7T = \begin{bmatrix} 1 & 0 & 0 & +0.004 \\ 0 & 1 & 0 & 0.0 \\ 0 & 0 & 1 & 0.085 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$