Learning to Program Mobile Robots in the ROS Summer School Series

Alexander Ferrein*, Stephan Kallweit*, Ingrid Scholl*, and Walter Reichert†
*Mobile Autonomous Systems & Cognitive Robotics Institute
FH Aachen University of Applied Sciences, Aachen, Germany
†Department of Mechanical Engineering and Mechatronics
FH Aachen University of Applied Sciences, Aachen, Germany
Email: {ferrein, kallweit, scholl, reichert}@fh-aachen.de

***Abstract*—The main objective of our ROS Summer School series is to introduce MA level students to program mobile robots with the Robot Operating System (ROS). ROS is a robot middleware that is used by many research institutions world-wide. Therefore, many state-of-the-art algorithms of mobile robotics are available in ROS and can be deployed very easily. As a basic robot platform we deploy a 1/10 RC cart that is equipped with an Arduino micro-controller to control the servo motors, and an embedded PC that runs ROS. In two weeks, participants get to learn the basics of mobile robotics hands-on. We describe our teaching concepts and our curriculum and report on the learning success of our students.**

I. Introduction

The main objective of our ROS summer school series is to introduce MA level students to program mobile robots with the Robot Operating System (ROS) [1]. As a basic robot platform we deploy a 1/10 RC cart that is equipped with an Arduino micro-controller to control the servo motors, and an embedded PC that runs ROS. Besides us being passionate about mobile robots, we believe that we enable students to deal with some key future technologies as we try and outline in the following.

Why are mobile robots important for future university graduates in Computer Science, Mechatronics, Mechanical Engineering?
Under the topic cyber-physical systems (physical entities that have a computing or network communication unit attached to it), mobile robots will be ever more important in the factories of the future. While the customization of products will become a huge issue in the future, mobile robots will have to step in and help with logistics tasks in the smart factories of tomorrow. It is sometimes called the fourth industrial revolution; the first came with the invention of the steam engine, the second came with the invention of the assembly line and the third came with the computer and the Internet. Now we are facing the fourth industrial revolution. Production is changing right now and is going to change dramatically in the near future. Besides mass production, customized products will be more and more important in the future. This has inter alia the effect that the designer of the product will be closer together with its manufacturer and it is believed, for instance, that by 2020 10–30 % of the products that the USA are importing from China today could be produced inland [2]. The new production will be supported by so-called cyber-physical systems. These systems combine computation with physical processes. They include embedded computers and networks which monitor and control the physical processes and have a wide range of applications in assisted living, advanced automotive systems, energy conservation, environmental control, critical infrastructure control, smart structure or manufacturing [3].

Why is it still a quite hard task to program a robot?
In [4], Brian Gerkey asks the question, why it is hard to write robot software: "The biggest obstacle to broader adoption of robotics is that only experienced roboticists can develop robotics applications. To make a robot reliably and robustly do something useful, you need a deep understanding of a broad variety of topics, from state estimation to perception to path planning. While few people in the world have this expertise, many people can write software. What we need is more of those software developers involved in the business of developing robotics applications. I say 'applications' to distinguish this work from that of developing new algorithms or core building blocks. Making an analogy to traditional software development, I don't need to understand how process schedulers, or file systems, or memory managers work in order to develop useful desktop applications. And I don't need to know the details of DNS, web servers, or web sockets to develop portable web applications. Knowing more about the underpinnings of the system will always be useful, of course. But the key is that, once the building blocks are established, understood, documented, and tutorialized, the barrier has been greatly lowered: you just need to be able to write code. [...]"

With ROS the task of developing robotic applications have become much easier. Many researchers world-wide contribute their research results as Open Source ROS packages. It was never that easy to download and run highly sophisticated robot software. However, to be able to run your robot with the different packages, still quite some expert knowledge is required. Not only do one need very good programming and system skills of the surrounding OS; to be able to set up and adapt all the parameters that come with a particular ROS package, quite some deep understanding of the subject matter is needed.

This is exactly what we are targeting with the ROS Summer School series at FH Aachen University of Applied Sciences. The participants of the Summer School get a 1/10 cart robot that is equipped with an IMU and an RGB-D camera. For controlling the servos, we make use of Arduino-based flight controllers developed for Unmanned Aerial Vehicles (UAVs). In two to three weeks of lectures and hands-on sessions, we lay the foundations for future mobile robot developers. In the morning, the theoretical background of advanced mobile robot topics is explained. This is done in a lecture style. Additionally, we invite renowned guest speakers to present their work—related to ROS. In the afternoons, we provide guided hands-on sessions where the participants need to apply the imparted knowledge practically with making certain related ROS packages run or with implementing several high-level robotic tasks. The goal in the end is to drive the cart autonomously around a track confined by wooden curbs. These will be used to detect road borders with the RGB-D camera. The last days of the competitions are reserved for a race competition where teams of participant compete against each other. A future goal is to develop the task of driving around the track into a scaled version of the Urban Challenge Scenario. During the 2014 edition of the summer school, we had about 48 participants from 10 different countries and at least three different disciplines (CS, Mechanical Engineering, and Mechatronics).

In the next section, we will outline the hardware system which we use for our summer schools. In Section III, we give an overview of the concept behind the ROS Summer School and the topics being taught, while in Section IV, we show the assessment of the summer school's participants and discuss ways to improve the curriculum for the future. We conclude with Section V.

II. Hardware Platform

There is a large number of robots that support the Robot Operating System (see, for instance, [5]). We decided however against using one of them for several reasons. For one, for a summer school with 40–50 participants, many of the available ready-to-run platforms are beyond the price scope. Considering that about 20 platforms (about 2 students share a platform) are required, the price becomes a major cost factor. For another, to show the participants that building a robot with available hardware components themselves is also valuable. For the selected hardware it is important that ROS drivers are available. Our decision of which hardware components to choose was based on the following criteria:

- Good Ubuntu OS support
- Low power consumption
- High processing power
- Multi-core architecture
- Open Source hardware components
- Low cost
- Robustness

ROS is best run under *Ubuntu OS*. There is a lot of support already for different Linux distributions, but a stable system is the easiest entry point for a beginner with binary ROS packages that run out of the box. As most binary packages are pre-compiled for Intel architectures, using one makes life much easier. On the other hand, Intel architectures usually have a higher *power consumption* for the mobile system and higher costs. When comparing Intel with ARM architectures, the processing power vs. power consumption is still remarkably better for ARM architectures. The price for ARM hardware is as well lower compared to Intel processors. In addition, a *multi-core architecture* with a large number of cores, e.g., from the used Odroid XU system, is well-suited for the distributed, fine grained ROS framework: most of the tasks are implemented in small programs (nodes) which are distributed over the eight cores of the used CPU. The description of the components for the mobile robot should be available to the students, so the hardware schematics and the software drivers need to be *Open Source*. Well-documented drivers for sensors commonly used in mobile robot applications such as the Asus Xtion RGB-D camera are also available. The complete system is targeted for a student budget, so it should not exceed a price of around US$ 600, so that students can build one on their own. There are of course very *low cost* robots available—mostly based on just one micro-controller like Arduino—but they do not have an interface to common hardware components such as web cams or RGB-D cameras and have too little *processing power*. These robots cannot run all the software algorithms we intend to teach during the summer school such as SLAM, visual odometry, or navigation. The hardware for the driving system needs to be as rugged and *robust* as possible and still affordable.

Instead of building our own platform with motors, electronic speed controllers, wheels and encoders, we decided to check for available *low-cost* RC cars for outdoor driving. Our hardware consists of:

- CPU, Odroid XU or Intel NUC
- Flight Controller, Crius[1] or PixHawk[2]
- RC Crawler chassis, Ridgecrest AX10[3]
- LiPo driving battery, 5800mAh
- RGB-D camera, Asus Xtion
- IR Ranger, Sharp GPD series
- Custom-designed mounting base

Figure 1 shows the assembled FH Rover based on the 1/10 RC Crawler. The RC Crawler has a quite rugged driving chassis, enough headroom and payload for carrying the CPU, the battery, the flight controller and the sensory devices. The springs can be adjusted or replaced for higher loads and the damper oil can be exchanged as well. There are carts available for a lower budget, but the ones we use are very reliable and worth the price. The used IMU is part of the flight controller design. The Crius has an on-board Invensense MPU6050 which is well documented and where a lot of source code is available to read out the raw data from the tilt and gyro sensors, for instance. The Crius is based on an Arduino design, so the implementation of the *rosserial* package can be easily shown via connecting an IR range sensor

[1] http://www.rctimer.com/
[2] www.3drobotics.com/pixhawk
[3] http://www.rccrawler.com/

to the I/O pins of the AVR2560 micro-controller. The *rosserial* package is an easy way for beginners to interface nearly any embedded system with ROS and is the right starting point for connecting additional hardware without explicit ROS drivers to a ROS-based system. As an alternative design for higher computational demands, we use an Intel NUC Core-i5 device together with the PixHawk 4 flight controller.

III. The FH Aachen ROS Summer School Concept

A. Challenges and Concepts

The basic idea is to give a general introduction to mobile robotics during the summer school. Our ROS Summer School runs for two weeks[4]; plenty of time, but still it is quite tough for teaching a comprehensive course in robotics. Robotics is a wide research field and as Gerkey, one of the main developers of ROS, pointed out in [4], a lot of experience is also required to design good robot software applications. The required qualifications range from solid mathematical understanding for dealing with noisy data, kinematics, or dynamics over good understanding of electronics or physics to distributed real-time software systems running—as in the case of ROS—under a Linux OS. Writing a robot application requires some knowledge in mechanical engineering, mechatronics, computer science and software engineering in general, computer vision and AI, in particular. The group of students attending the *ROS Summer School at FH Aachen* is quite mixed between CS and Mechatronics students, Electrical and Mechanical Engineering majors. Hence, the target group is quite diverse as are the prerequisites of the participants. Up till now, we did not have particular requirements apart from basic knowledge in programming and the Linux operating system.

Our teaching concept therefore foresees intensive workshop sessions with lectures in the morning and hands-on tutorials in the afternoon. For the afternoon sessions, we provide a number of skilled tutors who can individually help the participants with their particular problems. These range from problems with running a terminal under Linux via solving compiler and linker errors to discussing solutions to robotic problems and their implementation in ROS. We include social activities such as visits to close-by research institutions and universities that are active in the field of mobile robotics. To motivate the students beyond "ordinary" classwork, the last days of the summer school are reserved for a competition among teams of students. Our concept consists of the following five building blocks:

1) Lectures, invited talks, and hands-on sessions;
2) Lightning talks by participants;
3) Visits to research laboratories;
4) Industrial exhibitions and demos;
5) Final competition.

We briefly discuss them in the following.

[4]In the 2014 edition, we extended the duration to three weeks. While this was generally a good experience, we will switch back to two weeks for 2015.

1) Lectures, invited talks, and hands-on sessions: Lectures usually take a 2–3 hour span in the morning. The topics covered in the lecture series as well with hands-on tutorials are:

- Basics of ROS;
- Working with proximity sensors under ROS;
- Basic image processing;
- Bayes filtering, localization and mapping;
- Implementing basic reactive control algorithms.

In the first week, we lay the foundations with a general introduction to ROS. We introduce the concepts of the *rosmaster*, *nodes*, *topics* etc. and the students get to know the basics of a distributed real-time system. In the hands-on sessions, the participants write their own first nodes, exchange data and write a joystick teleoperation node for the Rover. Furthermore, they get into touch with the Rover hardware, in real as in simulations. They learn how to exchange data with the attached Arduino MC and have to read sensor signals from the MPU 6050, an integrated 6D IMU. They have to attach an IR Ranger from Sharp and use the package *rosserial* to communicate with the ranging device.

After the first week, all participants are fairly familiar with ROS and are able to learn higher concepts. Therefore, we focus on higher-level functionalities in the second week. We start with an introduction to kinematic modeling, local navigation and visual odometry, inertial navigation, and the data integration with an Extended Kalman Filter; on the practical side, we introduce the packages *tf*, *rviz*, *rqt_plot*, *robot_pose_ekf*, and the *ccny* package. Additionally, we introduce the Bayes Filter and have sessions on related ROS packages. Another focus lies on image processing. We use the RGB-D camera for basic image processing tasks such as color segmentation in 2D, and obstacle detection with 3D data. Fig. 2 shows the local obstacle map from the track that was extracted from the point cloud provided by the RGB-D camera. We have brief introductions to OpenCV and the bridge into ROS [6], [7] as well as the Point Cloud Library (PCL) [8]. We also introduce the general concepts of high-level control and prepare the students to get the carts ready for the final competition. During the preparation time for the competition, the participants have to code a simple control strategy that keeps the Rover in the middle of the road. Fig. 3 shows part of the competition track for the final competition. The detection of the road borders is done with the depth image that is projected to the ground plane to provide a 2D occupancy grid of the current camera frame.

We rounded up our lecture series by three invited talks with guest lecturers from University Nuremberg, Freiburg University and German Aerospace (DLR). They are all experts in specific fields, like e.g. Rescue Robotics, SLAM algorithms and UAVs. All of them are using ROS extensively and gave a lot of hints and advices how to start with mobile robotics and even participating in RoboCup [9].

2) Lightning talks by participants: As some participants already had some experience with ROS or robotics applications, we encouraged lightning talks by participants for

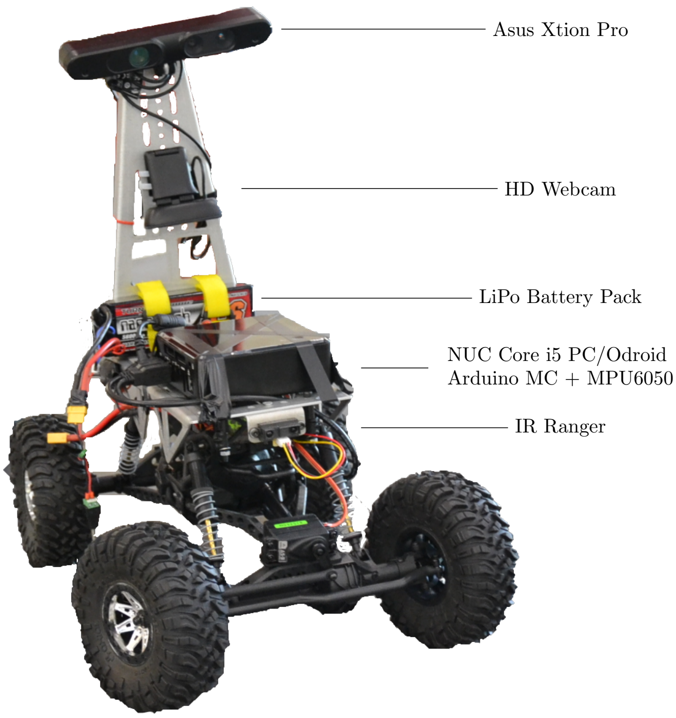

Figure 1. The FH-Rover 1/10 carts

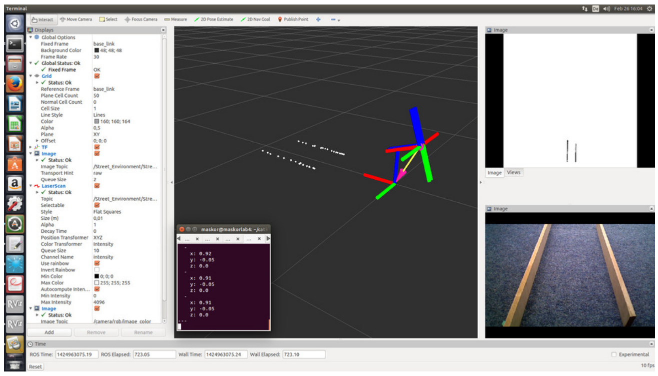

Figure 2. Occupancy scans from the track.

Figure 3. Competition track.

participants. We had some interesting talks on mutli-copter control and an industrial application, where a laser cutting machine was loaded and unloaded with a simulated PR2 and a real programmable logic controller (PLC).

3) Visits to research laboratories: We organized several social activities around the ROS summer school. In the beginning we had the usual get-together and a guided city tour. At the end of the first week, we visited the Aldebaran Laboratory[5]. We were given some presentations on the Nao Robot. At the end of the second week, we visited the Biorobotics Laboratory[6]. While many of the researchers were still attending the 2014 RoboCup Championships in Brazil with their Domestic Service Robot team, we got an excellent tour through their laboratory. It was in particular insightful to show that much of the material covered in ROS Summer School was also deployed on their robots. This underlined the significance of our teaching activities.

4) Industrial exhibitions and demos: As we pointed out in the introduction, the material covered by the ROS Summer School is not only relevant for mobile robotics which is mainly done by research units throughout the world. With the advent of cyber-physical systems in manufacturing, also an important impact for the automation industry can be expected. To guide the student in that direction, we organized a Special Session for industry partners to show where ROS would be used for their products. We had a presentation by FESTO Didactics giving a demonstration of the new Robotino 3 robot. Another partner from industry was the German distributor of Universal Robots. They showed their latest developments in compliant robot manipulators and the future impact of the taught methods w.r.t. human-machine collaboration. Here, a brief overview of industrial applications was given by mentioning *ROS Industrial*; so the learned skills can be even applied to industrial robots.

5) Final competitions: For the final competition we gave the participants some leeway regarding lectures and theory. They had three days time to develop a controller that would be able to autonomously drive the vehicle around the track. To keep everyone in the competition, we defined three competition levels with varying complexity. To make it a fun event and to make everybody be able to participate, the first stage was to teleoperate the robot around the track with a joystick. At the second stage, the robot had to perform the same task

[5]Aldebaran, Paris, France

[6]Biorobotics Laboratory, Delft Robotics Institute, Delft University of Technology, Delft, The Netherlands

Figure 4. Impressions from the 2014 ROS Summer School.

autonomously. For the third stage, the robot had to obey a street light that was at some unknown position on the race track.

B. Financial Background and International Relations

An effort to organize a summer school for up to 50 participants with hardware, accommodation, catering etc. would not be possible without substantial financial support. Some of the costs could be covered with the participant's registration fees of € 550. Given that, for the 2014 ROS Summer School, this included travel expenses for trips to Paris and Delft, catering and drinks, this is a quite reasonable price.

In addition, the ROS Summer School is approved for the DAAD scholarship program university summer courses in Germany for 2015. This program will support the ROS Summer School for 2015 and the following two years.

The ROS Summer School can also be rated as a success w.r.t. international relations. With participants from 10 countries, it was indeed an international summer school. The Summer School yields also good international exposure for participants of our partner universities and other institutions as well as for our local participating students. In particular, they get some international experience without having to stay away for a whole semester. Additionally, they can earn credit points for the course outside our semester times. This unravels the quite crowded study plans of the students or offers the opportunity to speed up their studies. The scholarships provided by the DAAD scholarship program will support international students to participate in the ROS Summer School also in the future. The 2015 ROS Summer School event promises to be even more international, since we already have about 50 application in total, with some of the applicants coming from Pakistan, Algeria, Tunisia, Egypt and Saudi Arabia.

IV. Improving the Summer School

In total, a number of 48 students registered for the 4th ROS Summer School, which was held between July 21 and August 8, 2014. About half of the students were international students from Brazil, China, Hungary, India, Portugal, Russia, Slovenia, Taiwan, and the USA. The other half were local students from FH Aachen who enrolled for Computer Science, Mechatronics, or Mechanical Engineering. The course was evaluated through our center of university didactic. The students answered a standardized evaluation form and estimate the following features:

Indicator	**Evaluation** μ	σ
Global Indicator	1.86	0.83
Structure and commitment of lecturer	1.88	0.84
Realization	1.75	0.67
Relevance of the lecture	2.06	0.89
Social interaction	1.62	0.86
Exercises, seminars, preparation for exams	2.03	0.91
Study success	2.04	0.84
Overall rating	1.67	0.78

Table I
Results from the evaluation form.

structure and commitment of lecturer, realization, relevance of the lecture, social interaction, exercises, seminars, preparation for exams, study success and overall rating. At the end of the evaluation form some open questions can be answered as free text, i.e. what did the student like most, how could the lecture be improved and how can be described the lecture in one sentence. Table I summarizes the students overall rating in a range from 1 to 5. While it was quite a challenge to host such a variety of different backgrounds and nationalities, overall, we got very positive feedback from the students.

Many students acknowledged the practical experience coupled with more theoretically oriented lectures. In particular, they liked being given enough time to find solutions to the given exercises with the support of our tutoring team. Although in 2014, the summer school was three weeks, some participants found it even too short. Another lesson we learned from the assessment of the participants was that we need to hand out more preparation material before the Summer School starts. For the next edition, we will make some virtual hard drives available where the participants can use the ROS simulation environment Gazebo [10] and prepare already for the summer school with our cart model in the simulation. As we mentioned above is the background of the participants quite diverse. There are students from Computer Science, Mechanical Engineering, Mechatronics with very different skills w.r.t. programming or working with Linux. In the last editions we did not have particular participation requirements for the summer school apart from the mentioned basic ones. As more and more students have used ROS before and therefore already have some working knowledge in ROS, we need to adapt our curriculum. Therefore for the 2015 edition of the ROS summer school, we will set up two parallel tracks. One *beginner track* for participants without prior knowledge, and an *advanced track* for students with good working knowledge in ROS and robotics. The reason for this split is not to bore the more advanced students with basic stuff they already know. On the other hand in the beginner track, we could focus more on the basics of robotics and on ROS, in particular. For the advanced track, we also equip our carts with the Intel NUC to have more processing power available.

For both tracks, we will stick to the final competition. The participants will start in two groups. The task for the beginner will be according to the gathered skill. As it turned out in the 2014 ROS summer school, writing a reactive controller to drive around the track is challenging but doable during the summer school. For the advanced track participants, this task could easily be made more complicated by, say, obeying street light and street signs, do some global path planning on a more complicated track layout with crossings and stop signs. In the future, the scenario of the race could be extended towards a small-scale Urban challenge scenario with street crossings, traffic lights, and street signs. Then, also more advanced robotics tasks such as localization, path planning, object recognition (sign and object detection) can be tackled. For 2015, we will offer students the possibility to buy the FH-Rover. Then, they could also continue their work on the rovers and deepen their skills with using ROS.

V. Conclusion

In this paper, we summarized the ideas and the background of the ROS Summer School. The basic idea is to teach the basics of mobile robotics and the programming of robotics applications with the Robot Operating System as a software basis. This has the advantage that many state-of-the-art algorithms and methods are available quite out of the box. To use such algorithms the right way, however, some theoretical knowledge in mobile robotics is required. Therefore, we combine thorough theoretical lectures teaching the concepts and fundamentals of mobile robotics with hands-on tutorials where participants can try out different algorithms and have to solve fundamental problems of mobile robotics. If not more, participants get a good overview of the problems in robotics. We think that this is very important to prepare our students for some future technological challenges coming from cyber-physical system and the future in automation.

With a varied programme consisting of lectures, hands-on tutorials, invited guest lectures by robotic experts, visits to robot institutions, we offer an interesting and challenging programme for our participants. While, until now, we did not require special prerequisites apart from basic Linux and programming skills, for future editions we will provide two tracks in the programme, one for beginners in mobile robotics and one for advanced users of ROS.

At the end of each summer school there is a competition among the participants. They have to program the rover carts so that they are able to drive around a track autonomously. For the future, we want to increase the complexity of this domain to end up with a small-scaled Urban Challenge scenario with street crossings, street signs where the robot have to stop at red traffic lights and overtake other vehicles.

Acknowledgment

The authors want to thank the number of students who supported the effort throughout the year with developing software and hardware for the rovers, preparing lectures and tutorials and, of course, doing a great job as tutors at the event. In particular, we want to thank Heiko Engemann, Christoph Gollok, Kai Krückel, David Masternak, Sebastian Schmidt, Marcel Stüttgen, Patrick Wiesen, Daniel Zwilling. Without an excellent organization team caring for all the organizational details and raising funds, such and effort would not be possible. We want to express our gratitude to Grit Breuer and Nathalie Kanj.

The authors acknowledge the financial support from FH Aachen; the FH Aachen Department of International Affairs-the DAAD (German Academic Exchange Service) and the German Federal Foreign Office. We thank the anonymous reviewers for their useful comments.

References

[1] M. Quigley, K. Conley, B. P. Gerkey, J. Faust, T. Foote, J. Leibs, R. Wheeler, and A. Y. Ng, "ROS: an open-source Robot Operating System," in *ICRA Workshop on Open Source Software*, 2009.

[2] The Economist, "The third industrial revolution," , vol. Vol. 12, no. 16, 2012, http://www.economist.com/node/21553017.

[3] E. Lee, "Cyber Physical Systems: Design Challenges," in *11th IEEE Int. Symp. on Object Oriented Real-Time Distributed Computing (ISORC)*, may 2008, pp. 363–369.

[4] B. Gerkey, "What is the single biggest obstacle preventing robotics from going mainstream," 2013, last visited June 10, 2013.

[5] "Robots using ros," http://wiki.ros.org/Robots, 2014, last visited 28/02/2015.

[6] G. Bradski and A. Kaehler, *Learning OpenCV: Computer vision with the OpenCV library*. O'Reilly Media, Inc., 2008.

[7] B. Cyganek and J. P. Siebert, *An introduction to 3D Computer Vision - Techniques and Algorithms*. Wiley, 2009.

[8] R. B. Rusu and S. Cousins, "3d is here: Point cloud library (pcl)," in *Proceedings of the 2011 IEEE International Conference on Robotics and Automation (ICRA-2011)*. IEEE, 2011.

[9] H. Kitano, M. Asada, Y. Kuniyoshi, I. Noda, and E. Osawa, "RoboCup: The Robot World Cup Initiative," in *Proc. 1st Int. Conf. on Autonomous Agents*, 1997.

[10] N. Koenig and J. Hsu, "The many faces of simulation: Use cases for a general purpose simulator," in *Proc. of the ICRA'13 Workshop on Developments of Simulation Tools for Robotics & Biomechanics*, 2013.