Bold Hearts Team Changes for RoboCup 2020 (Humanoid Kid Size League)

Rebecca Miko, Marcus Scheunemann, Alessandra Rossi, Sander van Dijk,
Roberto Figueiredo, and Daniel Polani

School of Engineering and Computer Science
University of Hertfordshire, Hatfield AL10 9AB, UK
boldheartsuh@gmail.com
https://robocup.herts.ac.uk

1 Hardware Changes

In RoboCup 2019, we changed the hardware to increase the robots' size and improve performance. The main processing unit had been replaced with an Odroid-XU4, and shins, legs, foot plates, head bracket and arms had been redesigned and 3D-printed. During the competition, we realised that the MX28 servos used were not powerful enough to support the new weight and height, and the robot was unable to perform basic behaviours. For this reason, we now use the MX64 servos for the robots' knees. Consequently, additional modifications to the design of the legs was required to support the new servos. The knees must be made bigger and the 3D-print must be stronger. The new legs have been designed and are now in the testing phase, with different 3D-printing materials being explored.

We also began to integrate the IKWalk[1] engine in our code[2], and began the first tests during the 2019 competition. The engine is promising, but a few hardware changes are needed for the robot. Most notably, the hip servos connected to the robot torso have too much of slackness to have a general engine solving the walk issue. We hypothesise that this can be solved by using a different material for the 3D printed legs. Currently, we are experimenting with the feasibility of several materials, including polycarbonate.

Lastly, in 2018 we improved the robot vision system by adopting new cameras (Logitech C920 Pro HD Webcam). As a result, the old head no longer fits our robots. Therefore, as an enticing starter project for our new students, we are in the process of designing a new head.

2 Robotic Operating System (ROS 2)

From 2013 until last year, we had developed and used our own software framework. However, our current software was limiting for the development of new behaviours. Therefore, we designed a more general and flexible framework for developing our robot software using ROS 2 [2].

[1] IKWalk engine by team Rhoban: https://github.com/Rhoban/IKWalk.

[2] The ROS 2 `ik_walk` package: https://gitlab.com/boldhearts/ros2_ik_walk.

Despite contributing a substantial amount of packages to the community[3], the transitioning to the new framework is not finished yet. For example, we were lacking a fluent provisioning system for the robots, entailing the most current changes for an upcoming game. However, we're close to finish the transitioning and providing a general framework for RoboCup teams. The new framework will provide several benefits for new students and for existing team members. Additionally, we expect that a ROS 2-based experience will be appealing for recruiting new members and creation of professional collaboration with researchers from both industry and academia.

We have developed a script that allows us to check out the latest changes of a set of branches and deploying those changes to a robot. Next step includes the development of a cross-compiling system and a build farm to reduce the delay due to the long building time of those packages on the robots.

3 Vision Advances

In 2019 we integrated our vision system [1] into ROS 2. As a consequence, we have developed several open-source ROS 2 packages, and have contributed several enhancements to standard ROS packages to better support similar hardware setups, such as support for the YUY2 encoding commonly used in webcams[4].

Firstly, we developed and released the `v4l2_camera`[5] package, which is now available through ROS 2's official repositories. Secondly, we developed an open-source ROS 2 package for integrating TensorFlow Lite[6]. With this setup we are able to run a semantic segmentation network at full resolution and a good framerate [1], which is retrained within 30–60 minutes on a modern laptop.

This year our focus is on extending the network to be able to handle more object classes, while still maintaining good performance. To achieve this, we are working to utilize both our hardware's GPU (Mali-T628 MP6), and TensorFlow Lite's quantization techniques to optimize the model's latency.

References

1. van Dijk, S.G., Scheunemann, M.M.: Deep Learning for Semantic Segmentation on Minimal Hardware. In: Holz, D., Genter, K., Saad, M., von Stryk, O. (eds.) RoboCup 2018: Robot World Cup XXII. Lecture Notes in Computer Science, vol. 11374, pp. 349–361. Springer International Publishing (Aug 2019). https://doi.org/10.1007/978-3-030-27544-0$_2$9
2. Scheunemann, M.M., van Dijk, S.G.: ROS 2 for RoboCup. In: Chalup, S., Niemueller, T., Suthakorn, J., Williams, M.A. (eds.) RoboCup 2019: Robot World Cup XXIII. Lecture Notes in Artificial Intelligence, vol. 11531. Springer International Publishing (2020)

[3] The full list of ROS 2 packages: https://gitlab.com/boldhearts.

[4] https://github.com/ros2/common_interfaces/pull/78 and https://github.com/ros-perception/vision_opencv/pull/309

[5] https://gitlab.com/boldhearts/ros2_v4l2_camera

[6] The ROS 2 `tflite` package: https://gitlab.com/boldhearts/ros2_tflite.