RoboEireann Team Description RoboCup 2015 Standard Platform League

Rudi Villing[1], John McDonald[2], Simon O'Keeffe[1], Guillaume Gales[2], and Robert Kelly[2]
[1]Department of Electronic Engineering [2]Department of Computer Science
Maynooth University
Maynooth, Co. Kildare, Ireland
http://www.roboeireann.ie

1 Introduction

1.1 Team

RoboEireann is a Robocup team from Maynooth University, and is currently Ireland's only RoboCup Standard Platform League team. RoboEireann is a collaborative effort between the staff and students of the Computer Science and Electronic Engineering Departments of Maynooth University, both of which have strong research records in the wider areas of computer vision, signal processing, control, robotics, and intelligent systems. In previous years we have also hosted visiting researchers from other teams and collaborated with Queen's University Belfast.

We first participated in RoboCup in 2008 as part of the Standard Platform League team *NUManoids* which was a joint effort of Maynooth University and the University of Newcastle, Australia. In that year, the first in which the Aldebaran NAO platform was used, we were Standard Platform League's overall winners. Since 2009 we have competed as an individual team under the name *RoboEireann*.

As RoboEireann we have competed in RoboCup 2009, 2010, 2011, and 2013. In RoboCup 2013, Eindhoven, we reached the quarter finals of the competition for the first time. In RoboCup 2011, Istanbul, we achieved first place in the technical challenge for our open challenge "Localisation without goal posts". The approach which was based on an extension to the work of Cox [1] was published in [8] and [9]. We have also competed in the RoboCup German Open every year from 2010 to 2014, achieving 2^{nd} place overall in 2012 and 2^{nd} in the best player award for the drop-in player competition in 2014.

For RoboCup 2015, the team is comprised of the following staff, graduate students, and undergraduate students:

Staff members: Rudi Villing (Team Leader), John McDonald, Guillaume Gales.
Students: Simon O'Keeffe, Robert Kelly, Daniel Moraes, Daniel Trusca, Adil Mahmood, Jadhe Ferreira, Tajinderpal Gill.

1.2 Impact

RoboEireann has made a number of technical contributions to the standard platform league since 2008. The distance and accuracy of our strong kick design was a notable contribution in early years. Our localization system developments based on Kalman filters and subsequently based on extensions to the line based registration algorithm due to Cox also contributed to progress within the league. Other technical contributions by the team include: early development of the b-script system for specifying behaviours, development of a light weight modular architecture, and kernel fixes to the Aldebaran Linux kernel to deal with a number of camera driver issues[1]. Over the years we have also hosted members from other RoboCup teams for extended research visits at our lab.

RoboCup and the standard platform league provide an excellent training and development environment for students in the areas of robotics and real time software systems. In Maynooth University we have a number of research groups active in the area of robotics and RoboEireann provides an excellent means for engaging the CS and EE undergraduate community with those groups. Every year, academic staff associated with RoboEireann have supervised undergraduate projects and internships that expose students to both the practical and cutting edge aspects of of robotic software development. The association with RoboCup is a very positive motivating factor that greatly affects the students' desire to get involved and perform well. Since our initial involvement in RoboCup we have had a number of undergraduate students who, as a consequence of their involvement in the team, have completed Masters and PhDs in robotics in our respective labs.

In addition to engaging students within the university, our involvement in RoboCup has played a key role in the successful running of a three year Summer Internship in Autonomous Robotics (SIAR) programme [2] and outreach activities to promote engagement of the Irish public with science, technology, engineering, and maths as part of the Discover Programme. Both programmes received financial support from Science Foundation Ireland (SFI). The SIAR programme funded 30 summer internships over its duration for both national and international students at Maynooth University. The Discover Programme has funded outreach activities for robot soccer demonstrations seen by hundreds of children, families, and the wider public as part of the National Science Week.

2 Architecture and key subsystems

The RoboEireann software architecture is based around two main threads: the MotionAgent and the CognitionAgent as shown in Fig. 1. The CognitionAgent runs at the vision frame rate (i.e. 2 cameras×30Hz) and implements all the "intelligence" of the robot. The MotionAgent runs at the Naoqi DCM rate (i.e. 100Hz) and implements the low-level motion control. The MotionAgent interfaces with

[1] https://github.com/mp3guy/linux-aldebaran/commits/release-1.12/geode

[2] http://www.eeng.nuim.ie/robocup/siar/siar.php

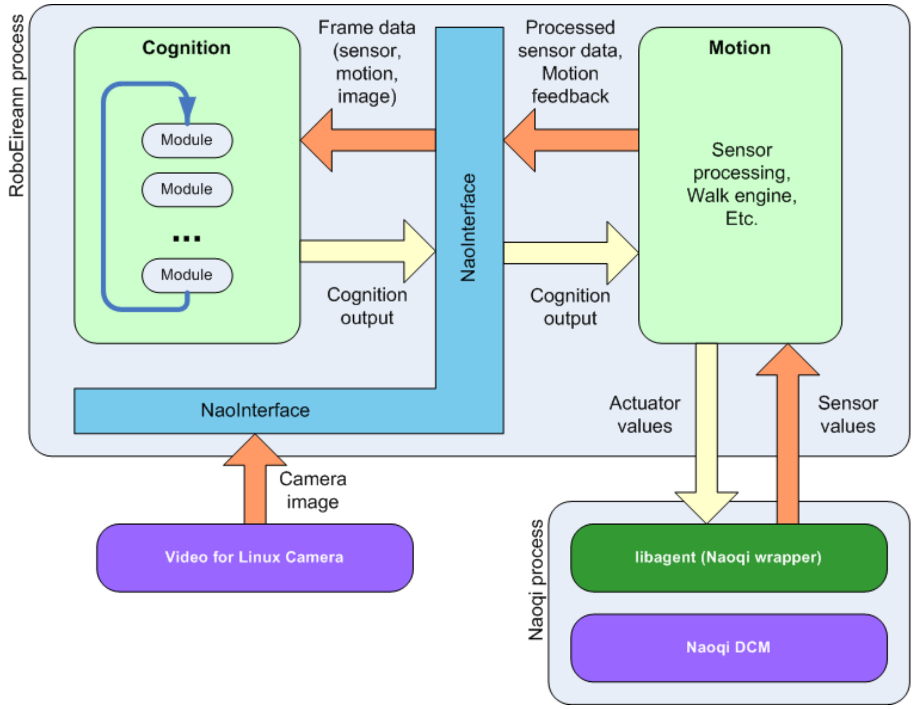

Fig. 1: RoboEireann Architecture

the robot hardware through a separate *libagent* process which in turn manages all low level direct communication with Naoqi and isolates Naoqi from crashes in the robot control code. The libagent implementation is a ported version of the equivalent libbhuman component from B-Human [6].

The design of the system is based on an original blackboard style micro module framework that has been implemented completely in-house. Each independent piece of information that is shared between several modules is represented in a single *MemoryUnit*. Separate modules of functionality are implemented as independent classes. Modules can read data from and write data to the memory units in the shared blackboard but they must declare their dependencies in advance. Each module can write to one or more memory units, but only one module (the owner) is permitted to write to a given memory unit. A module may also declare a read dependency on one or more memory units. The framework is responsible for ensuring that memory units to be read have previously been written so that they contain valid data.

Each thread in the architecture implements the superloop architecture for real time systems such that all activities for a given iteration of the loop must be completed before the loop period expires. In the case of the cognition agent, this loop period is approximately 16 ms. Within the superloop, the module framework calls modules sequentially to perform one frame worth of processing.

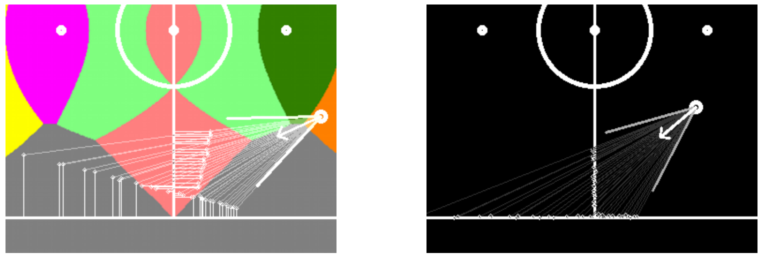

Fig. 2: Modified Cox Algorithm (MCA) processing showing the use of the precomputed voronoi diagram for optimised correspondence estimationg and error calculation (*left*), and resulting optimised robot pose (*right*). [9]

These calls are scheduled in a fixed round robin order determined during framework initialization to ensure that dependencies between modules which read and write individual memory units are satisfied.

2.1 Robot Perception

Vision in the RoboCup environment presents many key challenges. Our vision system is built around a colour segmentation and labelling approach. To maximise the performance of the labelling stage we have developed techniques for automatic camera setting calibration using a genetic algorithm to optimise the camera parameters such that the difference between all colours of interest is maximised. This is similar to the approach taken by Grillo et al. [2].

We also have preliminary experience with image correlation based algorithms for line/goal detection [4]. Image correlations have the advantage of being highly robust to noise and occlusions in classifying fixed field objects.

Up to 2012 we primarily used an extension of Cox's algorithm for vision based localisation in conjunction with a single model unscented Kalman filter [7]. Cox's algorithm, effectively an early variant of the iterative closest point (ICP) algorithm, obviates the need for line determination and clustering, a central strength of our vision system in our team's earlier participation. A highly optimised implementation coupled with a novel means of determining point correspondences (via a precomputed voronoi diagram) made the technique not only robust but also computationally feasible [9] (see Fig. 2).

In 2013 as part of our effort to make our localisation system more robust we developed a multi-hypothesis unscented Kalman filter (MHUKF) localisation approach. This system utilised a percept based vision pipeline. This year work has focused on combining the strengths of both these systems, whereby the output of Cox algorithm will be input to the MHUKF based localisation. This is an important development in the context of this year's rule change to white goals, given the Cox algorithm's robustness to such issues.

Finally we note that the computational performance of a vision system is always a principal concern in any RoboCup league and hence we have exploited as much of the hardware CPU capabilities as possible in our codebase, including the use of assembly and MMX instructions [9].

2.2 Motion

Walking We originally used the Aldebaran walk, but found that it was rather slow in a RoboCup context and had some difficulties stopping near and approaching the ball. Since RoboCup 2011 we have instead been using a modified version of the 2010 B-Human walk [5]. The B-Human walk engine was modified where necessary to utilise features already available in our code in preference to adding unnecessary duplicated code from the B-Human code base.

Kicking Our primary kick engine was designed around carefully designed poses for backswing, mid strike, foot lift and recovery. In all cases, key aspects of torso, arm and leg movements were considered in the design so that maximum kicking force and accuracy could be achieved. Using a small number of main kick designs we could independently specify the kick direction within a range of ball placements by interpolating between different kick parameter sets. This kick performed well at numerous RoboCup and Open competitions where tests showed that we could kick more than the length of the old 6×4m field. Our kick design is still relatively unique, but other teams have subsequently achieved similar performance with alternative designs.

As the RoboCup soccer game has improved in recent years, the importance of dealing with the ball quickly has increased. For this reason, in 2013 we started using fast and short in-walk kicks that were part of the B-Human walk engine. Our behaviour system now applies heuristics and a probabilistic element to decide which kick to perform when the ball is within range.

Balance In 2013 we introduced a basic balance engine. This balance engine can compute a static balance solution for any feasible target leg position relative to the current support foot. Balance is achieved by moving the torso and adjusting the target foot position (relative to the torso) to compensate for the constrained hip-yaw-pitch joint of the NAO robot. This version of the balance engine also allows the target leg to move through simple trajectories and computes the required joint configuration to ensure the robot remains statically balanced throughout.

2.3 Localisation and World Modeling

During the earlier years of our participation in RoboCup our localisation was mostly based on a single model unscented Kalman filter (UKF). As part of this early work we also evaluated alternative approaches including Particle Filtering [9], [7].

More recently we have adopted a multi-hypothesis UKF (MHUKF) approach where each robot runs multiple (e.g. 10) UKF's each of which tracks a separate hypothesis. During each localisation iteration a standard predict/update cycle is applied to each UKF independently, with each resulting estimate ranked based on a probabilistic scoring. The final estimate is selected as that of the UKF with the highest rank.

Since 2012 we have employed a number of techniques to overcome the symmetric field problem introduced by the rule change to uniformly coloured goals. Rather than taking a heuristic approach we simply allow each UKF to evaluate the probability of the observed posts given the robot's current estimate and chose the most likely possibility. This can be seen as splitting the model (two ways for a two post perception and four ways for a single post perception) and choosing the most likely based on the current estimate. We have found that this approach worked well for us as our incremental localisation is typically quite strong due to the method we use to integrate white field marking perceptions into the UKF [9]. However there are always points at which robot kidnappings occur. For this reason we introduced a shared world model that allows an agent to "flip" their world estimate given information from team mates. If an robot's estimate of the ball location is not consistent with two or more ball locations reported by other robots, the ball location is evaluated as if the world model was flipped. If this flipped world model is consistent with the location shared by many other robots, the flipped world model is accepted as correct.

2.4 Behaviour and Team Play

In 2011 our behaviour was based on hierarchical state machines. Since 2012 we replaced this with an early variant of the b-script behaviour engine initially developed in Ireland and subsequently refined in Germany [3]. This system uses a custom scripting language to specify behaviours using hierarchical generators.

Our current behaviour design is based on robot agents which can switch between a number of roles. In five-player SPL teams the roles have typically been striker, supporter, forward, defender, and goalie. Each agent determines their own role based on information they perceive themselves and information broadcast from their team members in the standard SPL packet. We have begun to modify this system to support greater flexibility and less hard switching between roles. This is particularly applicable to the the drop-in player competition (where behaviour should be more adaptive to team mates and the overall situation).

3 Research and development

For RoboCup 2015, we have focussed on a number of key areas. First, we have ported our current code to the NAO V5 robots. In addition to tuning the code for the new platform, our ongoing work includes making use of improvements to the NAO V5 robots in the areas of sensing (for example 3 axis gyrometer to facilitate yaw detection and better microphones) and further exploitation of the available computational resources to improve vision, perception, planning, locomotion, and world modelling.

Having had experience of both hierarchical state machines and hierarchical generators we have observed that each has different advantages which it would be useful to combine. Our ongoing work is also looking to incorporate probabilistic behaviour which can merge high-level reasoning for actions with low-level processing for perception, giving improved behaviour when dealing with uncertainty.

We are adding to our existing behaviours to account for various situations we encountered in competition, to account for rule changes, and to provide more intelligent and robust gameplay in general. Where possible, we intend that our new behaviours will preform more robustly in the drop-in challenge competition without role switching as other agents are not operating from the same set of roles.

We are continuing our research into the use of audio for team positioning and low bit rate communications based on wireless communications systems techniques.

A central focus of the 2015 research within the team has been the automation of all steps involved in the calibration of the robots. In the first instance this is to minimise the overhead and setup time required prior to gameplay, but our overall objective here is to evolve the current system to a point whereby the robots can adaptively adjust calibration parameters in an online fashion. Such capabilities should serve to significantly increase the robustness of our current system to variations in the kinematic configuration of the robot (e.g. due to impacts), or variations in ambient illumination (e.g. due to overhead skylights). This year we have a number of ongoing projects as part of this effort including (i) automatic kinematic calibration using AprilTags[3], and, (ii) automatic colour calibration using MeanShift segmentation (see Fig. 3).

Furthermore we are currently working on a visual compass approach to orientation estimation, and an improved visual localisation approach based on a sliding window feature tracking and mapping system which combines features over multiple video frames.

4 Conclusions

We have presented the RoboEireann Robocup Standard Platform League team, the software architecture and sub-systems, and the ongoing research and development work both for Robocup 2015 and beyond. This year our main effort has been in making the system more robust, extending existing algorithms to cater for 2015 rule changes (notably the use of white goals), porting our system to the new NAO V5 platform, and implmenting a number of enhancements to our calibration and diagnostic tools.

Our goal beyond Robocup 2015 is to further develop our codebase along each of these lines, and in particular to improve performance by automating as

[3] http://april.eecs.umich.edu/wiki/index.php/AprilTags

Fig. 3: Examples of current research and development: AprilTag kinematic calibration (*left*), MeanShift colour calibration (*right*).

much of the kinematic and sensor calibration as possible. Ultimately our aim is to develop a system that self-calibrates and is therefore capable of adpating online to both internal changes (such as degradation of actuation and sensing capabilities) and external environmental changes.

References

1. Cox, I.J.: Blanche-an experiment in guidance and navigation of an autonomous robot vehicle. Robotics and Automation, IEEE Transactions on 7(2), 193–204 (Apr 1991), http://dx.doi.org/10.1109/70.75902
2. Grillo, E., Matteucci, M., Sorrenti, D.: Getting the most from your color camera in a color-coded world. In: RoboCup 2004: Robot Soccer World Cup VIII, Lecture Notes in Computer Science, vol. 3276 (2005)
3. de Haas, T.J., Laue, T., Röfer, T.: A scripting-based approach to robot behavior engineering using hierarchical generators. In: ICRA. pp. 4736–4741 (2012)
4. Inam, W.: Particle filter based self-localization using visual landmarks and image database. In: Proc. IEEE International Symposium on Computational Intelligence in Robotics and Automation (December 2009), daejeon
5. Röfer, T., Laue, T., Müller, J., Burchardt, A., Damrose, E., Fabisch, A., Feldpausch, F., Gillmann, K., Graf, C., de Haas, T.J., Härtl, A., Honsel, D., Kastner, P., Kastner, T., Markowsky, B., Mester, M., Peter, J., Riemann, O.J.L., Ring, M., Sauerland, W., Schreck, A., Sieverdingbeck, I., Wenk, F., Worch, J.H.: B-human team report and code release 2010 (2010), only available online: http://www.b-human.de/downloads/bhuman10_coderelease.pdf
6. Röfer, T., Laue, T., Müller, J., Fabisch, A., Feldpausch, F., Gillmann, K., Graf, C., de Haas, T.J., Härtl, A., Humann, A., Honsel, D., Kastner, P., Kastner, T., Könemann, C., Markowsky, B., Riemann, O.J.L., Wenk, F.: B-human team report and code release 2011 (2011), only available online: http://www.b-human.de/downloads/bhuman11_coderelease.pdf
7. Whelan, T., Stüdli, S., McDonald, J., Middleton, R.H.: Line point registration: A technique for enhancing robot localization in a soccer environment. In: Proc. RoboCup Symposium (July 2011), istanbul
8. Whelan, T., Stüdli, S., McDonald, J., Middleton, R.: Line point registration: A technique for enhancing robot localization in a soccer environment. RoboCup 2011: Robot Soccer World Cup XV pp. 258–269 (2012)

9. Whelan, T., Stüdli, S., McDonald, J., Middleton, R.: Efficient localization for robot soccer using pattern matching. In: Leveraging Applications of Formal Methods, Verification, and Validation. Communications in Computer and Information Science (2012)