B-Human

Team Description for RoboCup 2014

Thomas Röfer[1], Tim Laue[2], Judith Müller[2], Michel Bartsch[2], Jonas Beenenga[2], Dana Jenett[2], Tobias Kastner[2], Vanessa Klose[2], Sebastian Koralewski[2], Florian Maaß[2], Elena Maier[2], Paul Meißner[2], Dennis Schüthe[2], Caren Siemer[2], Jan-Bernd Vosteen[2]

[1] Deutsches Forschungszentrum für Künstliche Intelligenz, Cyber-Physical Systems, Enrique-Schmidt-Str. 5, 28359 Bremen, Germany
[2] Universität Bremen, Fachbereich 3 – Mathematik und Informatik, Postfach 330 440, 28334 Bremen, Germany

1 Introduction

B-Human is a joint RoboCup team of the University of Bremen and the German Research Center for Artificial Intelligence (DFKI), consisting of students and researchers from both institutions. Some of the team members have already been active in a number of RoboCup teams such as the GermanTeam and the Bremen Byters (both Four-Legged League), B-Human and the BreDoBrothers (Humanoid Kid-Size League), and B-Smart (Small-Size League).

We entered the Standard Platform League in 2008 as part of the BreDoBrothers, a joint team of the Universität Bremen and the Technische Universität Dortmund, providing the software framework, state estimation modules, and the get up and kick motions for the NAO. For RoboCup 2009, we discontinued our Humanoid Kid-Size League activities and shifted all resources to the SPL, starting as a single location team after the split-up of the BreDoBrothers. Since then, the team B-Human has won every official game it played except for the final of RoboCup 2012. We won all six RoboCup German Open competitions since 2009. In 2009, 2010, 2011 and 2013, we also won the RoboCup.

This team description paper is organized as follows: After this paragraph, we present all current team members, followed by short descriptions of our publications since RoboCup 2013. After that, our newest developments since the end of RoboCup 2013 are described in the areas of vision (cf. Sect. 2), modeling (cf. Sect. 3), behavior (cf. Sect. 4), and motion (cf. Sect. 5). Our solutions for the technical challenges are outlined in Sect. 6. Since several teams are using our codebase, we describe some infrastructural changes we did since our last code release as well as in the GameController in Sect. 7.

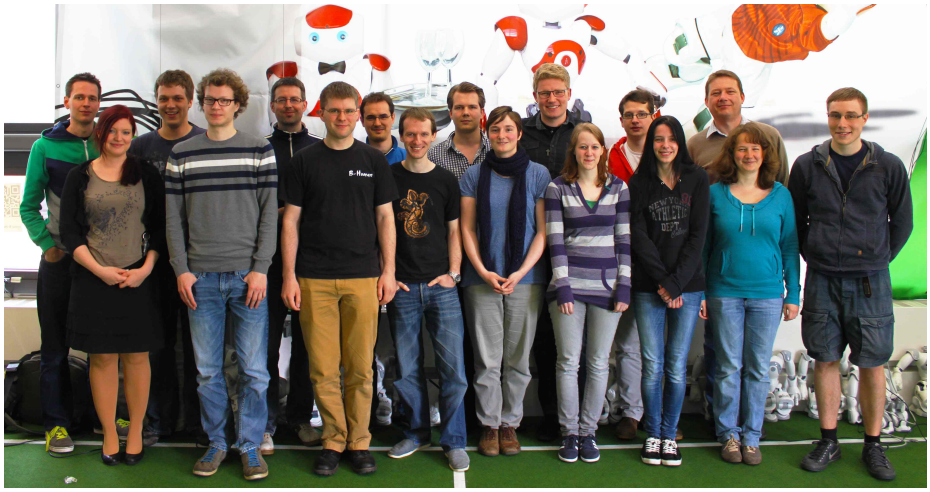

Fig. 1: All actual members of team B-Human 2014.

Team Members. B-Human currently consists of the following people who are shown in Fig. 1:

Team Leaders / Staff: Judith Müller, Tim Laue, Thomas Röfer.
Students: Michel Bartsch, Jonas Beenenga, Arne Böckmann, Dana Jenett, Vanessa Klose, Sebastian Koralewski, Florian Maaß, Elena Maier, Paul Meißner, Caren Siemer, Andreas Stolpmann, Alexis Tsogias, Jan-Bernd Vosteen, Robin Wieschendorf
Associated Researchers: Udo Frese, Dennis Schüthe, Felix Wenk

1.1 Publications since RoboCup 2013

As in previous years, we released our code after the RoboCup 2013 together with a detailed description [12] on our website (`http://www.b-human.de/en/publications`) and this time also on GitHub (`https://github.com/bhuman/BHuman2013`). Up to date, we know of 21 teams that based their RoboCup systems on one of our code releases (AUTMan Nao Team, Austrian Kangaroos, BURST, Camellia Dragons, Crude Scientists, Edinferno, JoiTech-SPL, NimbRo SPL, NTU Robot PAL, SPQR, Z-Knipsers) or used at least parts of it (Cerberus, MRL SPL, Nao Devils, Northern Bites, NUbots, RoboCanes, RoboEireann, TJArk, UChile, UT Austin Villa). In fact, it seems as if half of the teams participating in the main SPL soccer competition in RoboCup 2014 use B-Human's walking engine.

At the RoboCup Symposium 2014, we will present an automatic calibration for the NAO [7]. The method is based on the robot calibration [2] method of Birbach *et al.* [1] and involves an autonomous data collection phase, in which the robot moves a checkerboard that is mounted to the foot, in front of its

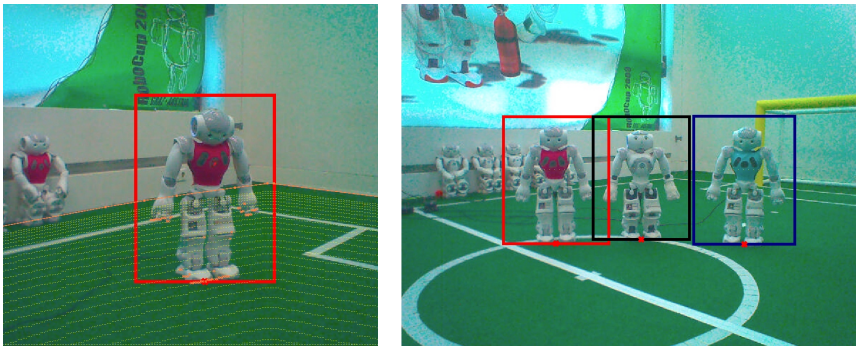

Fig. 2: On the left, it is scanned for a robot below the boundary of the field. On the right, jerseys are found on robots: red, unknown, blue.

lower camera. The calibration is formulated as a problem of non-linear least squares by minimizing the distances between the visually detected checkerboard vertices and their predicted image positions, using the NAOs forward kinematics and the pinhole-camera model. The problem is solved utilizing the algorithm of Levenberg [8] and Marquardt [9].

After RoboCup 2013, we published the invited champion paper [11] that describes some key aspects of our last year's success. These included a refined obstacle detection (via sonar as well as via image processing), a detection of the field's border, and a logging system that is able to save thumbnail images in real-time during normal games. In addition, the paper features descriptions of some human team members' workflows that are necessary to avoid annoying failures during competitions.

2 Vision

Automatic Camera Calibration. Calibrating the cameras is necessary since no two robots are alike, thus even small variations of the camera orientation can result in significant errors in the estimation of the positions of objects recognized by the vision system. We have been working on some methods to automate the calibration process. Since the manual calibration is a very time-consuming and tedious task, we developed a semi-automatic calibration module in the past. With this module the user has to mark arbitrary points on field lines, which are then used by a Gauss-Newton optimizer to estimate the extrinsic camera parameters. This year we additionally automated the process of collecting the points. They are obtained by the vision module that detects the white lines on the field.

Robot Detection. Our new robot detection module recognizes standing and fallen robots up to seven meters away in less than two milliseconds of computation time. This is an improvement compared to the module we used last year,

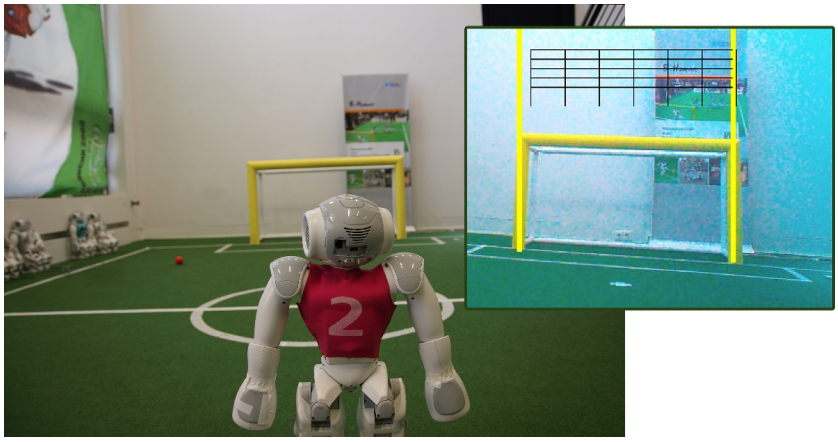

Fig. 3: Scanning area and blocks for goal side detection.

which had a lower range and a longer computation time. In addition, the new module scans for jerseys on the robots to get their team color.

Initially, the approach searches for robot parts within the field boundary, which is marked as orange line in the picture (cf. Fig. 2 left). From there on it scans down in vertical lines with growing space between the pixels looking at. The space and its growing rate are calculated from the distance of the corresponding points on the field. Every pixel scanned this way is marked yellow in the picture. If there are a couple of non-green pixels scanned in a vertical line, then the lowest of these pixels gets marked with a small orange cross. These are possible spots at the feet and hands of a robot, and they can be merged to a bounding box. After calculating the surrounding box, the method searches for a jersey within it and sets the color of the box to red or blue if successful (cf. Fig. 2 right).

Goal Side Detection. A big challenge in the Standard Platform League is the symmetric setup of the field, i. e., it is not trivial to know in which direction to play. We are currently investigating the possibility to use the background behind the two goals to distinguish between the two sides. To avoid seeing features that might continuously change, we use an area a bit further above the goals. We divide the area scanned into blocks (cf. Fig. 3) to compensate the displacement which is incurred when a robot scans the background of goals from different positions on field. For every block we calculate a histogram over every channel of the YCrCB color space and compare it with the reference data. By using these blocks, we are able to check whether there is a displacement or not. Because the view angle depends heavily on the robot's position on the field, we have to save some reference pictures from different positions on the field. Currently, we are working on a method to integrate more than one picture of each goal into the reference. The reference that matches the currently seen goal best indicates which field side we are currently observing.

3 Modeling

ObstacleModel. In early 2014, a bachelor thesis started with the aim to merge the obstacles recognized by ultra sound, computer vision, foot contact, and arm contact in a single representation instead of having multiple representations for each type of input data. For instance, so far, there is a representation *ObstacleWheel* [12] that is only based on the obstacles detected by the computer vision system. Moreover, the behavior mostly uses that representation, but relies on other input in some special situations.

The joint obstacle model creates an Extended Kalman Filter for each obstacle. All obstacles are held in a list. The position of an obstacle is relative to the position of the robot. Visually recognized obstacles (e. g. goal posts, robots) are transformed from the image coordinate system to field coordinates. Contacts with arms and feet are interpreted as an obstacle somewhere near the shoulder or foot in field coordinates. There is already an obstacle model based on ultrasound measurements that provides estimated positions on the field [12]. In the prediction step of the Extended Kalman Filter, the odometry offset since the last update is applied to all obstacle positions. The estimated velocity of an obstacle is also added to the position. The update step tries to find the best match for a given measurement and updates that Kalman sample. If there is no suitable sample to match, a new sample is created. In this step, we ensure that goal posts have a velocity of 0 mm/s in x and y directions. Due to noise in images and motion, not every measurement might create an obstacle if no best match was found. To prevent false positive obstacles in the model, there is a minimum number of measurements in a fixed duration required to generate an obstacle.

If an obstacle was not seen for a second, its "seen" counter is decreased. If a threshold is reached, the obstacle is deleted.

Free Part of Opponent Goal. Our field players try to kick in the direction of the opponent goal whenever it is possible. To determine, whether the path to the goal is free, and in which direction a kick would not be blocked by other robots, we determine the largest free part of the opponent goal. The previous implementation resulted from a time, when obstacle detection was not very reliable. Therefore, it was based on detecting the largest visible part of opponent's goal line. Since the obstacle detection was vastly improved recently (cf.. Sect. 2), and the goal line is hard to see on the now larger field, a new approach was implemented.

The new version uses the obstacle model presented above. The goal is divided into sections, which later indicate the free parts. All obstacles the robot perceives are projected in the direction of the goal, revealing which of them are actually blocking parts of the goal. Then the blocked parts are marked as "not free", all other parts of the goal, which are not blocked or hidden by an obstacle, are marked as "free" . After that the coherent "free" blocks are compared and the biggest "free" part is selected for the kick.

Fig. 4: On the left, a single obstacle is avoided. On the right, a new way right around the group is planned.

4 Behavior

Short Range Path Planning. Our behavior uses two different methods to create walk commands. The first one is an RRT planner [12] that plans trajectories across the field and thereby surrounds obstacles. The second one is a reactive component, that only avoids the closest obstacle and is used in the final phase of the approach to a target position.

A new method is currently in development to replace the second one. The path planning at close quarters should consider more factors in the planning process. The major difference is that the process looks at every robot and obstacle the robot perceives, instead of only the closest one. This allows the robot to plan its path around a whole group of obstacles. In order to achieve this, we group all obstacles (including robots, the ball, goalposts etc.) with the condition that an obstacle in a group stands close enough to another obstacle in this group. Then we search for the closest obstacle (and the corresponding cluster of obstacles) which stands in the direct way to the target. After we know the group to avoid, we simply calculate the convex hull of this group, the current position of the robot and the target position (cf. Fig. 4). The result is a path around a group and we can decide which way (left or right) we want to go around it.

Drop-in Player Competition. We mainly use the same behavior in the Drop-in Player Competition games as we do in the regular soccer competition. However, some of the information that is usually exchanged between B-Human players is not part of the standardized section of the SPL standard message, and therefore it is not available in Drop-in Player games. We cope with that fact by constructing the information we need and handle it separately. A B-Human player would, for instance, send its current role. For a Drop-in Player game, this is constructed locally from the intention and the position, which are given in the SPL standard message.

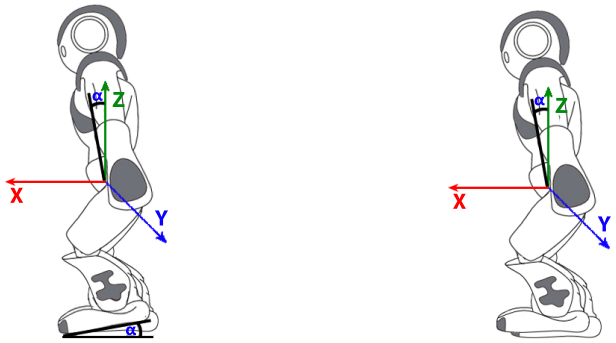

Fig. 5: On the left, the foot is rotated relative to the torso of the robot. On the right, the foot is rotated so that it is parallel to the ground.

As some information given by other players might be imprecise, wrong, or even missing, the reliabilities of all teammates are estimated during a game. This is done by continuously analyzing the consistency, completeness, and plausibility of the received data. In addition, it is checked, whether communicated ball perceptions are compatible to our own perceptions.

Depending on the information given by other players and the reliability of each of them, we decide whether we try to play the ball or take a supporting role for a different player.

5 Motion

Walking. Compared with 2013 [3], the walking was improved slightly with the aim of a smoother walk. The robot corrects the orientation of the lifted foot during the walk phase to make it parallel to the ground (cf. Fig. 5). The orientation of the torso is used to find out whether there is a deviation from the planed walk. The rotation around the Y axis gives the angle α that is used to correct the foot's pitch angle in small amounts.

Ball Stopping. In the Standard Platform League it is incontestable that many robots are able to perform very strong and accurate long distance goal kicks. As a result, it seems insufficient to only let the goalie actively stop the ball if there are usually also four other players on the field. Hence, we developed multiple methods that allow field players to stop the ball in different situations.

A crucial part for a ball stopping motion is, besides actually stoping the ball, to only perform it in the right moment. Otherwise, it would be a waste of time, which could be better used for a counterattack. For that reason we decided to never execute the motion based on insufficient data but only if the robot is really sure that the ball is very fast, the ball's movement direction is intersecting the

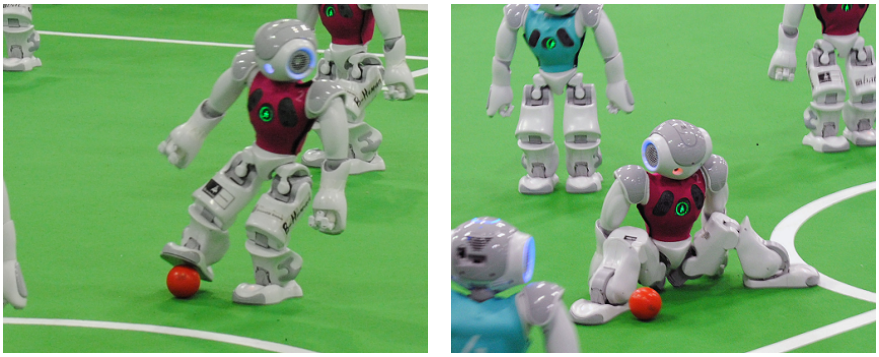

Fig. 6: On the left, a dynamically stopped ball from an initial standing position. On the right, the keyframe-based ball stopping motion from an initially fast walking robot.

area reachable by the stop motion, and it is very likely that a goal would be scored. Furthermore, the motion should be performed as quick as possible in order to execute it in time and to be ready for a counterattack. Consequently, every field player should be able to stop the ball even if it is walking very quickly. Since a robot usually needs some time to slow down from a very fast walk we developed two stop ball motions in order to react the best if starting from a standing or walking situation.

We think that the fastest and most accurate method to stop the ball is to set one foot in the right moment on top of the ball (cf. Fig. 6). As this is not that different from kicking a ball, we use our kick motion engine [10] to dynamically alter the motion trajectory of a prototypical stop ball motion similar to our long distance kick trajectories. In doing so, the robot is able to quickly react to the ball movement direction, and it can decide in advance where the foot has to be placed. Unfortunately, this motion only works really well if the robot is standing at the start. Otherwise, it takes too long for a walking robot to slow down with the result that the motion is not executed in time and therefore the ball cannot be stopped.

In case the robot is walking, we developed a key frame based ball stopping motion that is stable enough to interrupt the walk even if one foot is still in the air phase. Thereby the robot opens its legs to a split alike position (cf. Fig. 6) but leaves one sole of one foot on the ground, which ensures stable camera data. In doing so, we made sure that the motion is stable enough for the robot to track the ball during the movement. Thus, the movement can be aborted as soon as the ball movement direction or speed changes. In order to protect the robot best we took special care to reduce the motor stiffness in the right moments during the movement with the result that the robot lands fairly soft. Getting up afterwards is handled by our getup motion engine [12] with a modified starting point to match the split alike position from our normal getup routine.

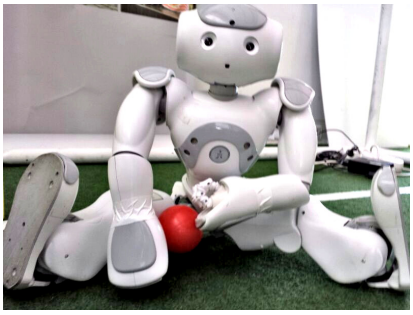
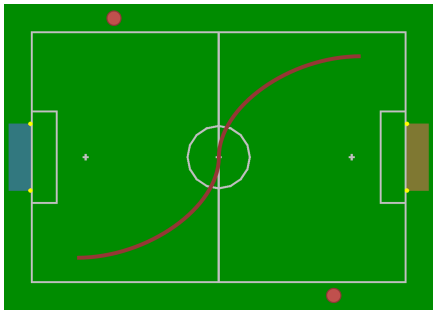

Fig. 7: On the left, a robot lifted the ball a little bit above the ground. On the right, the referee "lazy S" diagonal run and the assistant's positions are shown.

Ball Lifting. As the 2014 rules interpret it as a goal keeper's kick-out if it lifts the ball, we implemented a new motion that is able to pull the ball between the robots legs, very near to the body, so that the robot can role the ball up along a leg with one of its arms as shown in Fig. 7. The arm used to lift the ball is determined by the ball's position.

6 Technical Challenges

Open Challenge. In all RoboCup soccer leagues that use robots, refereeing is still performed by humans. However, some situations that referees have to decide upon are, at least partially, already detected by the players themselves. For instance B-Human's robots estimate where the ball has to reenter the field after it was kicked out and they also know when the ball is free after a kick-off. Therefore, it makes sense to think about which situations could be detected by robots and to start implementing robot referees.

The way a robot referee walks over the field is based on human referees in real soccer matches. In general, a game has a head referee and two assistants. Their positions and walking paths are displayed in Fig. 7. The two assistants are located on opposing field sides (red circles in the picture). The referee walks on a smooth diagonal curve, often called "lazy S", depending on the current ball position. Therefore, all relevant events are always taking place between it and one of the assistants, so all situations can be seen from two perspectives. Thus, it is less likely that none of them has seen a relevant situation.

Our Open Challenge contribution is the implementation of a robot head referee and its two assistants. The presentation involves these robots as well as field players of different teams. There will be no communication between the referee robots and the robots playing. The players provoke some situations that the referees should identify. Such situations could be an *Illegal Defender* or a *Fallen Robot*, but also a ball kicked out or a goal scored. The head referee robot will announce its decision and, when announcing a penalty, point at the position where the infraction happened.

Any Place Challenge. The aim of the Any Place Challenge is to make robot football more realistic by encouraging the use of less environmentally-dependent vision systems. Our current approach to object recognition is largely based on color segmentation and needs to be hand-tuned to the given lighting conditions. To overcome that limitation and to successfully take part in the technical challenge, we are adapting the works of Härtl *et al.* [5], in which the ball and the goals are found based on color similarities with a detection rate that is comparable to color-table-based object recognition under static lighting conditions, but that is substantially better under changing illumination.

Sound Recognition Challenge. In this challenge, robots have to recognize different signals based on a *Audio Frequency Shift Keying (AFSK)* as well as the sound of a whistle.

For recognizing the AFSK, we decided to demodulate the signal incoherent, i. e. without recovering the carrier [13]. We decided to check the frequency spectrum for the incoming signals, i. e. the AFSK signal when it is transmitted and most of the time noise. Therefore, we have to take different aspects into account, which are mainly the sampling frequency with respect to the highest signal frequency, the time domain resolution of the window to be transferred by *discrete Fourier transform (DFT)* into the frequency domain, and the window length of the DFT for the frequency resolution. Here we have a conflict between window length in time and DFT window length, as we get a higher resolution of the DFT spectrum with higher window length N, but this leads to a greater time window where the signal length is less than the window length, which is not useful.

The sampling frequency f_s has to be set to more than the doubled highest frequency in the spectrum (Shannon's theorem) $f_s > 2 \times f_{high}$. For our case, we have $f_s > 2 \times 320Hz \rightarrow f_s = 1kHz$. As the NAO microphones' lowest sampling frequency is 8 kHz, we downsample the signal to 1 kHz after we low pass filtered the 8 kHz signal to eliminate aliasing. As low pass filter we use a *Finite Impulse Response (FIR)* filter, which has a constant group delay [4] [6].

For the DFT length, we use approximately a third of the signal length, the rest which is then missing to get a good resolution in the frequency domain is padded with zeros. This so-called *zero padding* can also be used to get a good FFT (Fast Fourier Transform) length, which has to be a power of two. The window used is a sliding window, such that the last samples are also included in the next window frame. The advantage is given by a smoother frequency transition, e. g. from 200Hz to 320Hz [6].

Finally, we check the magnitude at the given signal frequencies compared to the overall average magnitude of a set of last spectra. This allows us to distinguish between noise and the signal itself.

The same process can be used for the whistle, as this has also a specified frequency. Moreover, we can check for harmonics in the higher frequencies. We only need to adapt the sampling frequency as the whistle's frequency is higher than 500 Hz (the highest detectable frequency at $f_s = 1kHz$).

7 Infrastructural Changes

B-Human Framework. After the 2013 code release [12], we completely switched to 64 bit with the desktop side of our software. In addition, we upgraded to Microsoft's Visual Studio 2013 on Windows, which supports the use of more C++11 features. The *STREAMABLE* macros introduced in the 2013 code release now use C++'s in-place initialization that allows streamable representations to have regular default constructors again. In addition, a similar syntax is now used to define the dependencies of our modules, i. e. their requirements and what they provide. This gives us more flexibility for the module base classes that are generated. Thereby, we were able to drop the central blackboard class without modifying existing modules and replace it with a hashtable-based blackboard that is not a bottleneck in terms of code dependencies anymore.

In addition, this allowed us to simplify our online logger. The latter was also improved in terms of performance. We also introduced downscaled grayscale images for online logging, which again reduces the logging bandwidth required by a factor of two. As an alternative it is now also possible to log full images rarely, e. g., once a second, to store in-game images that, e. g., could be used to improve the image processing system. At the RoboCup German Open 2014 we logged all games on all robots, except for the final, and used the data to improve our code for subsequent games.

The encapsulation of the behavior specification language CABSL that was introduced in 2013 was improved, which makes it a more general means to provide a state-machine to a class. For instance, the logger now also uses CABSL to manage its different processing states.

GameController. The GameController mainly developed by B-Human team members is used as the official referee application in the SPL since 2013 and in the Humanoid League since 2014. Due to the rule changes for the RoboCup 2014 it was required to implement these rule changes in the GameController. To adapt it to the latest rule changes, we added the official coach interface and support for a substitute player. In addition, the SPL drop-in competition was integrated as a mode with a different set of penalties and neither coach nor substitute. The GameController now also supports *all-time team numbers*, i. e. teams keep their number over the years and do not need to reconfigure it for each competition.

8 Conclusions

In this paper, we described a number of new approaches that we have implemented or we have been working on since RoboCup 2013. As in previous years, we did not re-develop major parts from scratch but aim towards incremental improvements of the overall system. This concerns all areas of research, i. e. vision, state estimation, behavior, and motion.

In addition to the "normal" soccer software, we work on approaches for all three technical challenges as well as for the new Drop-In Player Competition.

Besides our annual code releases, the GameController is another contribution of B-Human to the development of the Standard Platform League and the Humanoid Leagues.

References

1. Birbach, O., Bäuml, B., Frese, U.: Automatic and self-contained calibration of a multi-sensorial humanoid's upper body. In: Proceedings of the IEEE International Conference on Robotics and Automation (ICRA). pp. 3103–3108. St. Paul, MN, USA (2012)
2. Elatta, A.Y., Gen, L.P., Zhi, F.L., Daoyuan, Y., Fei, L.: An overview of robot calibration. IEEE Journal on Robotics and Automation 3, 74–78 (October 2004)
3. Graf, C., Röfer, T.: A closed-loop 3D-LIPM gait for the RoboCup Standard Platform League humanoid. In: Zhou, C., Pagello, E., Behnke, S., Menegatti, E., Röfer, T., Stone, P. (eds.) Proceedings of the Fourth Workshop on Humanoid Soccer Robots in conjunction with the 2010 IEEE-RAS International Conference on Humanoid Robots (2010)
4. von Grüningen, D.C.: Digitale Signalverarbeitung: mit einer Einführung in die kontinuierlichen Signale und Systeme. Fachbuchverlag Leipzig im Carl Hanser Verlag, München (2008)
5. Härtl, A., Visser, U., Röfer, T.: Robust and efficient object recognition for a humanoid soccer robot. In: RoboCup 2013: Robot Soccer World Cup XVII. Lecture Notes in Artificial Intelligence, vol. 8371. Springer (2014)
6. Kammeyer, K.D., Kroschel, K.: Digitale Signalverarbeitung: Filterung und Spektralanalyse mit MATLAB-Übungen. Vieweg + Teubner, Wiesbaden (2009)
7. Kastner, T., Röfer, T., Laue, T.: Automatic robot calibration for the NAO. In: RoboCup 2014: Robot Soccer World Cup XVIII. Lecture Notes in Artificial Intelligence, Springer (2015), to appear
8. Levenberg, K.: A method for the solution of certain problems in least squares. The Quarterly of Applied Mathematics 2, 164–168 (1944)
9. Marquardt, D.W.: An algorithm for least-squares estimation of nonlinear parameters. SIAM Journal on Applied Mathematics 11(2), 431–441 (1963)
10. Müller, J., Laue, T., Röfer, T.: Kicking a Ball – Modeling Complex Dynamic Motions for Humanoid Robots. In: del Solar, J.R., Chown, E., Ploeger, P.G. (eds.) RoboCup 2010: Robot Soccer World Cup XIV. Lecture Notes in Artificial Intelligence, vol. 6556, pp. 109–120. Springer (2011)
11. Röfer, T., Laue, T., Böckmann, A., Müller, J., Tsogias, A.: B-Human 2013: Ensuring stable game performance. In: RoboCup 2013: Robot Soccer World Cup XVII. Lecture Notes in Artificial Intelligence, vol. 8371, pp. 80–91. Springer (2014)
12. Röfer, T., Laue, T., Müller, J., Bartsch, M., Batram, M.J., Böckmann, A., Böschen, M., Kroker, M., Maaß, F., Münder, T., Steinbeck, M., Stolpmann, A., Taddiken, S., Tsogias, A., Wenk, F.: B-Human team report and code release 2013 (2013), only available online: `http://www.b-human.de/downloads/publications/2013/CodeRelease2013.pdf`
13. Werner, M.: Nachrichtentechnik: Eine Einführung für alle Studiengänge. Vieweg + Teubner, Wiesbaden (2010)