B-Human

Team Description for RoboCup 2011

Thomas Röfer[1], Tim Laue[1], Judith Müller[1], Alexander Fabisch[2], Katharina Gillmann[2], Colin Graf[2], Alexander Härtl[2], Arne Humann[2], Felix Wenk[2]

[1] Deutsches Forschungszentrum für Künstliche Intelligenz, Sichere Kognitive Systeme, 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 Universität Bremen and the German Research Center for Artificial Intelligence (DFKI). The team consists of numerous students as well as three researchers. The latter 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 already 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 getup 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 its start, the team B-Human has won every tournament it participated in. In 2009 and 2010, we won the RoboCup as well as the RoboCup German Open. This year, we already won the RoboCup German Open in Magdeburg, and we hope to be able to repeat our success of last years by also winning the RoboCup in Istanbul.

This team description paper provides a brief overview of our relevant publications since RoboCup 2010 (cf. Sect. 2) and of current work that is about to become used during the next competition (cf. Sect. 3).

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

Students. Alexander Fabisch, Arne Humann, Benjamin Markowsky, Carsten Könemann, Daniel Honsel, Emil Huseynli, Felix Wenk, Fynn Feldpausch, Martin Ring, Ole Jan Lars Riemann, Philipp Kastner, Thomas Liebschwager, Tobias Kastner.

Senior Students. Alexander Härtl, Colin Graf, Katharina Gillmann, Thijs Jeffry de Haas.

Staff. Judith Müller, Thomas Röfer (team leader), Tim Laue.

Fig. 1. The team B-Human at the RoboCup German Open 2011

2 Publications since RoboCup 2010

We are convinced that code releases are an important part of sharing scientific works with others. After RoboCup 2010, we therefore released our code – together with a comprehensive documentation [1] – to the public on our website `http://www.b-human.de/en/publications/`. We hope this act motivates other teams to release their code, too, or to use our code as a basis.

In order to achieve a reasonable level of play in the RoboCup Standard Platform League, a number of basic abilities are necessary. Foremost, this is a fast and robust gait. We use an approach that is based on the 3-Dimensional Linear Inverted Pendulum Mode [2], but that also uses sensor feedback from Nao's IMU for active balancing. While the walk used so far only employs the torso's tilt and roll angles for stabilization [3], an advanced version of the gait uses the difference between the expected torso pose and the estimated actual torso pose [4].

Other important abilities for playing soccer are obstacle avoidance and passing. These capabilities rely on information about other robots. In [5], we presented a vision-based approach for robot recognition in the RoboCup Standard Platform League as well as an algorithm to track the recognized robots. Both approaches were developed considering the limited computing resources of the Nao to allow an application in actual games.

3 Current Projects

In addition to the previously described, already published work, some new projects are still under development and expected to become finished until RoboCup 2011 or they are already finished. This includes the use of both cameras, which required a new semi-automatic camera calibration method, a field coverage model that enables faster ball searching, an assistance for color calibration, and a global ball model.

3.1 Using Both Cameras

The Nao robot is equipped with two video cameras, both mounted in the head of the robot. The first camera is mounted in the middle of the robots forehead, the second camera is installed about 4 cm below the first one and tilted by 40 ° with respect to the upper camera. Since the vertical opening angle of the cameras is only 34.8 °, stereo vision is impossible, and to make things worse, the cameras of the Nao can not deliver images simultaneously. This is why we previously used only one camera and disabled the other one completely.

Disabling one camera has a few drawbacks though. Using the upper camera only is prohibitive, because that would require the robot to bend down every time it wants to look at its own feet. While the consequences of using only the lower camera are not that severe, the field of view is narrowed down a lot by the shoulder pads when the robot wants to look at targets that are left or right to the robot.

In order to work around these problems, both cameras are used alternately. When it is requested to point the camera to a new target, it is dynamically determined whether switching to the other camera is useful, i. e. most importantly whether an otherwise obstructed target would become visible. If that is the case, the other camera is activated and appropriate angles for head joints are calculated.

In addition to expanding the field of view, dynamically switching cameras is used to avoid large head tilt acceleration while still being able to look at targets which are close to or far away from the robot in quick succession.

3.2 Camera Calibration

The process of manually calibrating the robot-specific correction parameters for a camera is a very time consuming task, since the parameter space is quite large (8 resp. 11 parameters for calibrating the lower resp. both cameras). It is not always intuitive which parameters have to be adapted if a camera is miscalibrated. Especially during competitions, when robots are often repaired on-site and therefore require recalibration, this is an annoying necessity.

In order to overcome this problem, we developed a semi-automatic calibration module (cf. [1] Sect. 4.1.1.1) last year. This module was typically used to get a good initial parameter set that was then adjusted manually. This year, we developed an improved version of the camera calibration module that differs in the following points:

– Instead of marking defined points on the field, the user can mark arbitrary points on field lines. This is especially useful for the operation during competitions, because it is also possible to calibrate the camera if parts of the field lines are covered.

- Since we use both cameras this year, the calibration module is able to calibrate the parameters of the lower as well as the upper camera. Therefore, the user must simply mark additional reference points in the image of the upper camera.
- In order to optimize the parameters, the Gauss-Newton algorithm is used[1] instead of hill climbing. Since this algorithm is especially designed for non-linear least squares problems like this, the time to converge is drastically reduced to typically 5–10 iterations. This has the additional advantage that the probability to converge is increased.
- During the calibration procedure, the robot stands on a defined spot on the field. Since the user is typically unable to place the robot exactly on that spot, and a small variance of the robot pose from its desired pose results in a large systematical error, additional correction parameters for the robot pose are introduced and optimized simultaneously.
- The error function takes the distance of a point to the next line in image coordinates instead of field coordinates into account. This is a more accurate error approximation, because the parameters and the error are in angular space.

With these improvements the module typically produces a parameter set that requires only little manual adjustments, if any.

3.3 Field Coverage Model

In many situations during the previous competitions and test games, our robots had problems to retrieve the ball, especially when they had no indication for the position of the ball. In general, it is a good idea to find the ball by inspecting parts of the field that were not regarded by any robot for a long time.

Therefore, we developed a field coverage model that indicates how much time has passed since the last time a robot saw a specific part of the field. This model divides the field into cells (see Fig. 2 (b)). Each cell has a counter that is initially set to zero. The counter will be set to a maximum value, if a cell center is within the visible area. The value is decreased in each execution cycle. This information provides the base for generating camera targets to efficiently search for the ball as displayed in Fig. 2 (d).

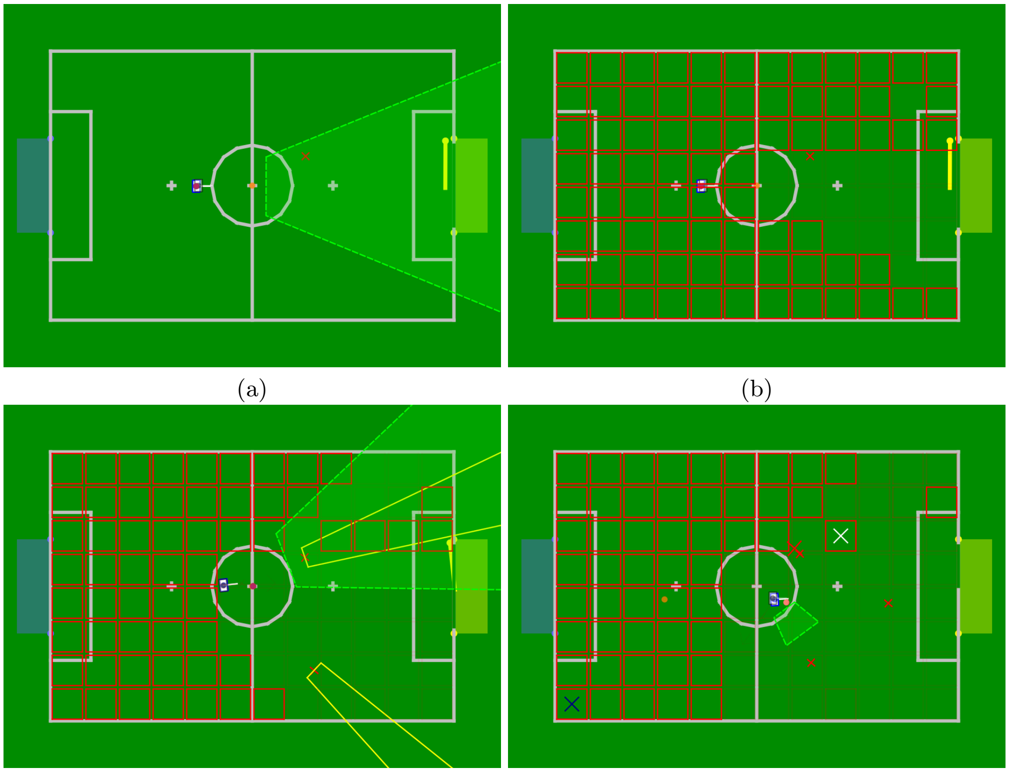
If a robot lost track of the ball, there is probably still no need to scan the pitch using the targets generated from the field coverage model. During normal game play, three hypotheses on the current ball model should be communicated by the other teammates to generate the global ball model (cf. Sect. 3.5). So even if the robot's own ball model can not be used, the robot is not entirely clueless. The situation starts to get delicate if there's no robot with valid ball model that can be shared among the team.

In the latter case, the assumption is that the ball is most likely on that part of the field, that was not regarded by any robot for the longest time – if it is on the field at all. Therefore, the four field coverage models are communicated among the robots as well, such that the number of places to look for the ball can be further reduced.

3.4 Color Table Assistance

Still a lot of image processing routines rely on colors and thus require a color calibration, in most cases realized as manually created color tables. Due to possible lighting changes,

[1] Actually, the Jacobian used in each iteration is approximated numerically



(a) (b) (c) (d)

Fig. 2. The robot's model of the world. a) The visible area projected on the field is shown by the green quadrangle. b) Uncovered cells are red. c) Other robots (depicted as yellow quadrangles) cast shadows over the visible area. d) The model generates the best camera targets to search for the ball. The white cross marks the best target that is reachable without turning. The blue cross marks the overall best target.

these color tables have to be created or at least corrected before every game, but often there is just little time for this and so errors might creep in. Obviously, it would be great to have a vision that does not rely on color tables at all, but for now, we simplified the process of creating and modifying them to reduce the expenditure of time and the error rate.

In the past, we created color tables completely by hand. When assigning a class to a color the same class was assigned to its surrounding colors in color space. So there was always a trade-off between precision and effort.

In contrast, using a nearest neighbor algorithm requires just a few colors to be set manually, but the results are quite precise. An example is shown in Figure 3. We can use such a time-consuming algorithm, because the calculation is done offline, before the game. To run the algorithm in acceptable time on a common laptop, we use a kd-tree to find the neighbors.

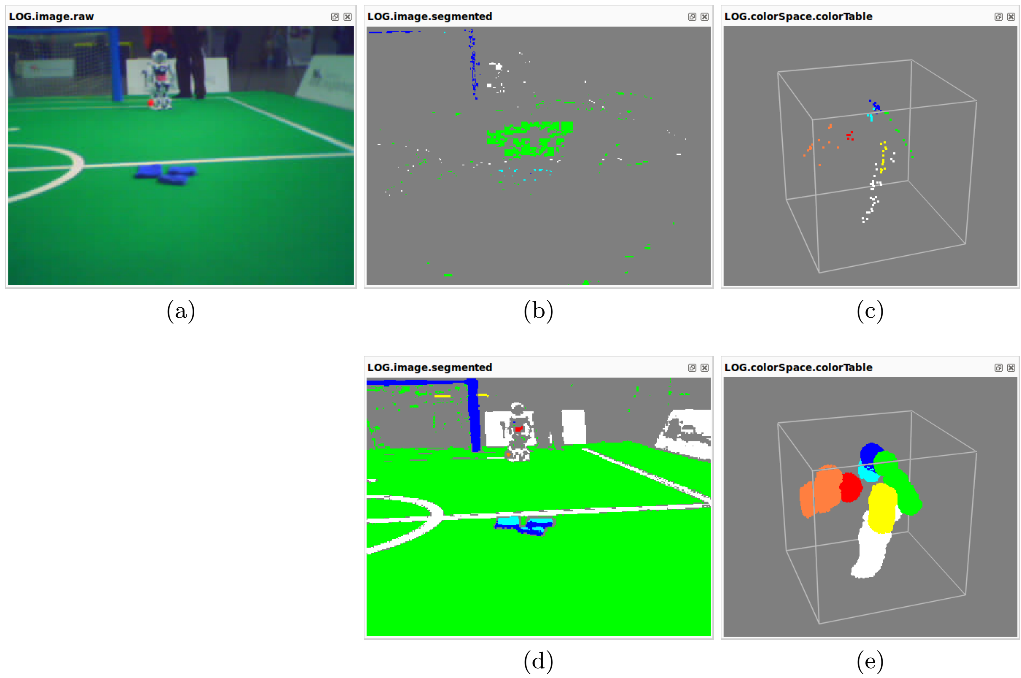


Fig. 3. The segmented images with the corresponding color table. a) The raw image seen by the Nao. b/c) The image segmented using the manual configuration. d/e) The segmented image after the nearest neighbor algorithm estimated the color classes.

3.5 Global Ball Model

Unlike some other domains, such as the Small Size League, the robots in the SPL do not have a common and consistent model of the world, but each of them has an individual world model, estimated on the base of its own limited perception. This induced us to implement a global ball model that lets all robots have an assumption of the current ball position, even if it was not seen by the robot itself. Additionally, this assumption is consistent among the team of robots (aside from delays in the team communication), this is useful for tasks such as the behavior role selection.

The calculation is done locally by each robot, but takes the ball models of all other teammates into account. This means that the robot first collects the last valid ball model of each teammate, which is in general the last received, except for the case that the teammate is not able to play, for instance because it is penalized or fallen down. In this case, the last valid ball model is used. The only situation in which a teammate's ball model is not used at all is if the ball was seen outside the field, which is considered as a false perception. After the collection of the ball models, they are combined in a weighted sum calculation to get the global ball model. There are four factors that are considered in the calculation of the weighted sum:

- The approximated validity of the self-localization: the higher the validity, the higher the weight.
- The time since the ball was last seen: the higher the time, the less the weight.

– The time since the ball *should* have been seen, i. e. the time since the ball was not seen although it should have appeared in the robot's camera image: the higher the time, the less the weight.
– The approximated deviation of the ball based on the bearing: the higher the deviation, the less the weight.

Based on these factors, a common ball model, containing an approximated position and velocity, is calculated.

Among other things, the global ball model is currently used to make individual robots hesitate to start searching for the ball, if they currently do not see it but their teammates agree about the ball position.

4 Conclusions

We are continuously trying to improve the performance of our team through numerous small and bigger developments in our code base. This paper only names a few of them, mostly targeting the problem of tracking the ball. In addition, we synchronize the gaze control between all robots of the team to keep track of the ball, we did first steps towards actively passing between robots, and we have further improved the reaction speed of our robots without sacrificing precision. We also worked a lot on the software infrastructure of our system, since we strongly believe that a complex software system such as ours has to progress as a whole. For instance, our simulator was restructured and has a new core, the configuration files use a new format, we switched from Subversion to git for version control, we upgraded our build system, and we now fully support MacOS X as development platform.

References

1. 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/file_download/33/bhuman10_coderelease.pdf`.
2. Kajita, S., Kanehiro, F., Kaneko, K., Fujiwara, K., Yokoi, K., Hirukawa, H.: A realtime pattern generator for biped walking. In: Proceedings of the 2002 IEEE International Conference on Robotics and Automation (ICRA 2002), Washington, D.C., USA (2002) 31–37
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. Graf, C., Röfer, T.: A center of mass observing 3D-LIPM gait for the RoboCup Standard Platform League humanoid. In Röfer, T., Mayer, N., Savage, J., Saranlı, U., eds.: RoboCup 2011: Robot Soccer World Cup XV. Lecture Notes in Artificial Intelligence, Springer (2011) to appear, also available in these pre-proceedings.
5. Fabisch, A., Laue, T., Röfer, T.: Robot recognition and modeling in the robocup standard platform league. 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)