B-Human

Team Description for RoboCup 2013

Thomas Röfer[1], Tim Laue[1], Judith Müller[2],
Michel Bartsch[2], Arne Böckmann[2], Florian Maaß[2], Thomas Münder[2],
Marcel Steinbeck[2], Simon Taddiken[2], Alexis Tsogias[2], Felix Wenk[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 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 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 five RoboCup German Open competitions since 2009. In 2009, 2010, and 2011, we also won the RoboCup. This year, we hope to be able to replicate previous successes by winning the RoboCup again in Eindhoven.

This team description paper is organized as follows: Section 1.1 presents all current team members, followed by short descriptions of our publications since the last RoboCup in Sect. 1.2. New developments after RoboCup 2012 to make the vision system more tolerant to changing lighting conditions are the topic of Sect. 2. Our new self-localization approach is described in Sect. 3. Section 4 presents our re-implementation of ultrasound-based obstacle detection. B-Human's new signature move, i. e. taking an arm out of the way when necessary, is described in Sect. 5. Changes in our software infrastructure are outlined

Fig. 1: The team B-Human at the RoboCup German Open 2013 award ceremony.

in Sect. 6. We are also working on the league's new GameController (cf. Sect. 7). The paper concludes in Sect. 8.

1.1 Team Members

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

Students. Alexis Tsogias, Andreas Stolpmann, Arne Böckmann, Florian Maaß, Malte Jonas Batram, Marcel Steinbeck, Martin Böschen, Martin Kroker, Michel Bartsch, Robin Wieschendorf (not on the picture) Simon Taddiken, Thomas Münder.

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

1.2 Publications since RoboCup 2012

As in previous years, we released our code after RoboCup 2012 – but this time only accompanied by an update report [1] – to the public on our website `http://www.b-human.de/en/publications/`. Up to date, we know of 15 teams that based their RoboCup systems on one of our code releases (AUTMan Nao Team, Austrian Kangaroos, BURST, Edinferno, NimbRo SPL, NTU Robot PAL, SPQR) or used at least parts of it (Austin Villa, Cerberus, MRL SPL, Nao Devils, Northern Bites, RoboCanes, RoboEireann, UChile). In fact, it seems as if the majority of teams participating in RoboCup 2013 uses B-Human's walking engine.

At the RoboCup Symposium we will present a method to generate and execute kick motions [2]. The method comprises calculating the trajectory of the kick foot online and moving the rest of the robot's body such that it is dynamically balanced. The latter is achieved by moving the Zero-Moment Point of the NAO, which is determined using inverse dynamics.

We will also present a new object recognition system, in which the ball, the goals, and the field lines 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 [3]. It still runs in real time (< 6 ms per image) even on a NAO V3.2.

At this year's RoboCup Symposium's open source track, a topic will be presented that has not been covered in detail by previous publications or code releases: The software architecture underlying the B-Human system [4]. The publication points out design choices and strengths of our current architecture, in particular in comparison to the currently popular Robot Operating System (ROS) [5].

For many years, our institute also had a team that participated in the RoboCup Small Size League: *B-Smart*. Former members of this team contributed to that league's standard vision system *SSL-Vision* [6]. In the paper by Zickler *et al.* [7], the history and recent applications and developments of this system – such as the application as a source for ground truth data in the Standard Platform League [8] – are described.

An ongoing project outside the current RoboCup domain, but still in the context of sport robotics, is the development of the entertainment robot *Piggy* [9] that is able to play a simple ball game with humans. This robot is stationary and does not play in a team but has some capabilities that are beyond the current RoboCup state of the art: On the one hand, its vision system is almost independent of (changing) lighting conditions and colors and thus capable of perceiving and tracking balls even outdoors. On the other hand, Piggy is able to perform dynamic and safe human-robot interaction.

2 Vision

2.1 Color Classification

Due to more varying lighting conditions at the German Open 2013, we implemented a more robust and easier to setup color classification. The main idea is based on using the HSV (hue, saturation, value) colorspace, because in that space, mainly the V channel will vary significantly if the lighting conditions change. Therefore, color classes are defined as ranges of the channels H and S, while there are only only implicit minimum thresholds for the V channel. Since the conversion from the YCbCr colorspace, in which the images are delivered by NAO's cameras, to the HSV model is rather expensive, additional thresholds for the channels Cb and Cr limit the number of pixels that have actually to be converted to the HSV colorspace. All colors are defined separately, which allows

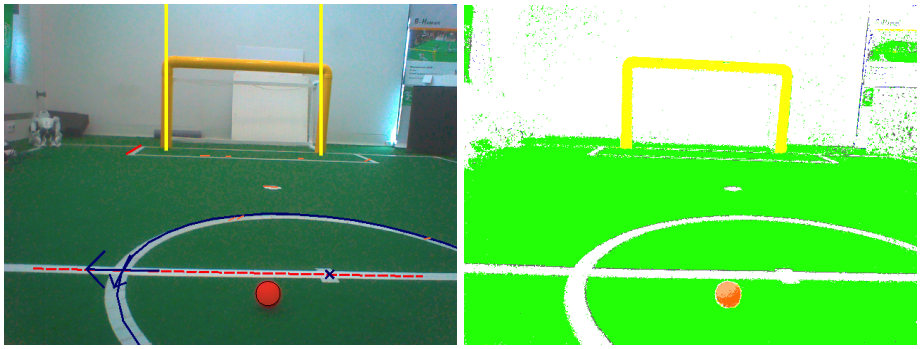


Fig. 2: An image (overlaid by the perceptions) and its color-classification

for defining ambiguous color classifications that are only resolved by the context in which they are used. For instance, in Fig. 2, a part of the ball is classified both as orange and white.

One of the main problems with this approach is the classification of the color white. Based on the description above a simple maximum threshold for the channel S should be sufficient to classify a pixel as white. However, due to the noise level of the camera, white pixels – in particular at the border between green and white – are slightly more saturated, making it very hard to define a threshold. Therefore we decided to use an alternative approach to classify white. For our vision system to work properly, white only needs to be distinguished from green and orange. Therefore we first check whether a pixel is not green in the HSV colorspace. If it is, the pixel is transformed into the RGB color space and validated against minimum blue and red thresholds.

2.2 Goal Detection

For getting a more persistent detection of goals, the corresponding perceptor was reimplemented. A scan-based approach now detects yellow regions intersecting with the horizon based on the new color classification. These potential goal posts are then evaluated for their match to optimal goal post models (cf. Fig. 3). This process checks agains a maximum distance and a minimal height evaluation as well for being on the own field by using the calculated field border (cf. Sect. 2.4). If these criteria do not excluding the posts, a constant width, the ratio between width and height, and the expected width and height for the measured distance are checked for further validation. In addition, the relation between different potential posts has influence on their ratings. In the end, the two best-rated potential posts above a minimum rating threshold are accepted as actual goal post percepts.

Fig. 3: Validation of potential goal posts

2.3 Obstacle Detection

This year's rule changes introduced a bigger field and an additional robot. The bigger field allows for more and different tactics such as passing or man-to-man marking. However to successfully deploy such tactics the robots need a good model of the surrounding environment. Therefore we have developed a new visual obstacle detection method for this year's RoboCup.

First a SSE-based implementation of a Sobel filter is used to generate an edge image of the Y channel. Afterwards the algorithm of Otsu [10] is used to calculate the optimal threshold between edges and non-edges in this image. Each edge pixel above the threshold belongs either to a field line, an obstacle, or the background. On the SPL field, each field line is completely surrounded by the green carpet. Since edges only occur at the sides of field lines we can assume that edges that do not separate a green from a non green area are obstacles or belong to the background. Since robots cannot leave the field, the background can be treated as an obstacle as well. Therefore every edge pixel above the threshold that does not separate a green from a non-green region is classified as belonging to a potential obstacle, i. e. an obstacle spot.

For each of these obstacle spots, a height is calculated by searching the image upwards and downwards for the beginning of a green region. If the height is above a certain threshold, an obstacle spot is accepted. In a post-processing step, obstacle spots that might result from arms of other robots are excluded, because the distance computation assumes that obstacles touch the ground, which is not the case for arms.

The remaining obstacle spots are clustered according to the distance between them and entered into a radial model of the environment. The model is similar to the one described by Hoffmann *et al.* [11].

2.4 Field Boundary Detection

The new rules state that fields may be located close to one another with no barrier between them. This required the implementation of a new field boundary

detector, since the old one scanned the image from top to bottom. This resulted in wrong boundaries located on other fields. The new approach uses both cameras and scans upwards beginning in the image from the lower camera, continuing in the image from the upper camera. This way it is possible to detect the boundary of the own filed more accurately. The field border is shown in both lower images of Fig. 5.

3 Self-Localization

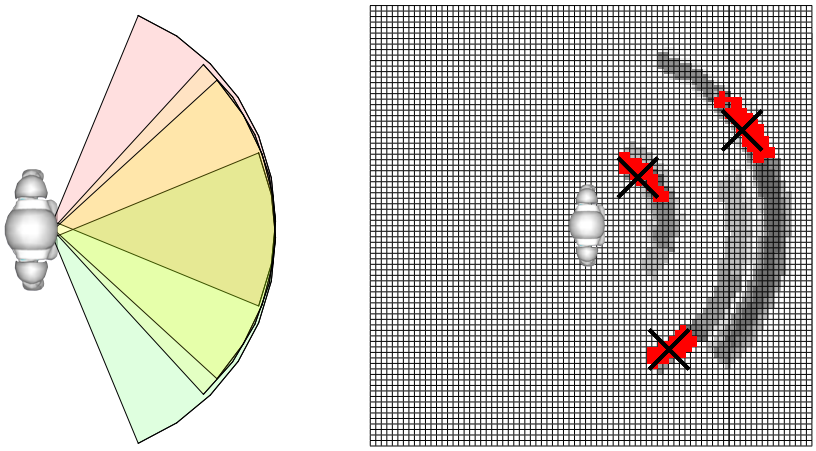
A robust and precise self-localization has always been an important requirement for successfully participating in the Standard Platform League. B-Human has always based its self-localization solutions on probabilistic approaches [12] as this paradigm has been proven to provide robust and precise results in a variety of robot state estimation tasks.

Since many years, B-Human's self-localization is realized by a particle filter implementation [13, 14] as this approach enables a smooth resetting of robot pose hypotheses. However, in recent years, different additional components have been added to complement the particle filter: For achieving a higher precision, an Unscented Kalman Filter [15] locally refined the particle filter's estimate, a validation module compared recent landmark observations to the estimated robot pose, and, finally, a side disambiguator enabled the self-localization to deal with two yellow goals [16].

Despite the very good performance in previous years, we are currently aiming for a new, closed solution that integrates all components in a methodically sound way and avoids redundancies, e. g. regarding data association and measurement updates. Hence, the current solution – that has already been used successfully during the RoboCup German Open 2013 – has been designed as a particle filter with each particle carrying an Unscented Kalman Filter. While the UKF instances perform the actual state estimation (including data association and the consequent hypothesis validation), the particle filter carries out the hypotheses management and the sensor resetting. Currently, only side disambiguation (based on game states and a team-wide ball model) still remains a separate component. However, current work in progress is to include this side assignment in the main state estimation process by adding it to each particle's state.

4 Ultrasound Obstacle Detection

Since we were not very satisfied with the ultrasound-based obstacle detection system implemented for RoboCup 2012 [1], we reimplemented it for this year. The NAO is equipped with two ultrasound transmitters (one on each side) and two receivers (again one on both sides), which results in four possible combinations of transmitter and receiver used for a measurement. We model the areas covered by these different measurement types as cones with an opening angle of 90° and an origin at the position between the transmitter and the receiver used (cf. Fig. 4a). Since NAO's torso is upright in all situations in which we



(a) (b)

Fig. 4: Ultrasound obstacle detection. a) Overlapping measuring cones. b) Obstacle grid that shows that sometimes even three robots can be distinguished. Clustered cells are shown in red, resulting obstacle positions are depicted as crosses.

rely on ultrasound measurements, the whole modeling process is done in 2-D. We also limit the distances measured to 1.2 m, because experiments indicated that larger measurements sometimes result from the floor. The environment of the robot is modeled as a 2.4 m $\times$ 2.4 m grid of cells with a size of 30 mm $\times$ 30 mm each. The robot is always centered in the grid, i. e. the contents of the grid are shifted when the robot moves. Instead of rotating the grid, the rotation of the robot relative to the grid is maintained. The measuring cones are not only very wide, they also largely overlap. To exploit the information that, e. g., one sensor sees a certain obstacle, but another sensor with a partially overlapping measuring area does not, the cells in the grid are ring buffers that store the last 16 measurements that concerned each particular cell, i. e. whether a cell was measured as free or as occupied. With this approach, the state of of a cell always reflects recent measurements. Experiments have shown that the best results are achieved when cells are considered as part of an obstacle if at least 10 of the last 16 measurements have measured them as occupied. All cells above the threshold are clustered. For each cluster, an estimated obstacle position is calculated as the average position of all cells weighted by the number of times each cell was measured as occupied.

Since the ultrasound sensors have a minimum distance that they can measure, we distinguish between two kinds of measurements: close measurements and normal measurements. Close measurements are entered into the grid by strengthening all obstacles in the measuring cone that are closer than the measured distance, i. e. cells that are already above the obstacle threshold are considered to have been measured again as occupied. For normal measurements, the

area up to the measured distance is entered into the grid as free. For both kinds of measurements, an arc with a thickness of 100 mm is marked as occupied in the distance measured. If the sensor received additional echoes, the area up to the next measurement is again assumed to be free. This sometimes allows narrowing down the position of an obstacle on one side, even if a second, closer obstacle is detected on the other side as well – or even on the same side (cf. Fig. 4b). Since the regular sensor updates only change cells that are currently in the measuring area of a sensor, an empty measurement is added to all cells of the grid every two seconds. Thereby, the robot forgets old obstacles. However, they stay long enough in the grid to allow the robot to surround them even when it cannot measure them anymore, because the sensors point away from them.

5 Arm Motions

In [1] we introduced arm contact detection to recognize when a robot's arm collides with an obstacle. In this year's system we integrated dynamic arm motions on top of that feature. If a robot detects an obstacle with an arm, it may decide to move that arm out of the way to be able to pass the obstacle without much interference. In addition, arm motions can be triggered by the behavior control, allowing the robot to move its arm aside even before actually touching an obstacle.

An arm motion is defined by a set of states, which consist of target angles for the elbow joint and the shoulder joint of each arm. Upon executing an arm motion, the motion engine interpolates intermediate angles between two states to provide a smooth motion. When reaching the last state, the arm remains there until the engine gets a new request or a certain time runs out. While the arm motion engine is active, its output overrides the default arm motions, which are normally generated during walking.

A development version of this feature has already been used during RoboCup 2012 and it was improved for the German Open 2013. It since proved to be very valuable in many game situations, because the arms of our robots are out of the way if they need to be, but they still function as obstacles to the opponent team in defensive situations.

6 Infrastructural Changes

This year we have spent a lot of time on improving our infrastructure. Among the more noteworthy features are a custom kernel, a new behavior description language, nearly complete *C++11* support due to the use of *LLVM clang* as compiler and cross compiler, a streamlined installation procedure, logging of camera images at full frame rate during the game, and some extensions to the module framework.

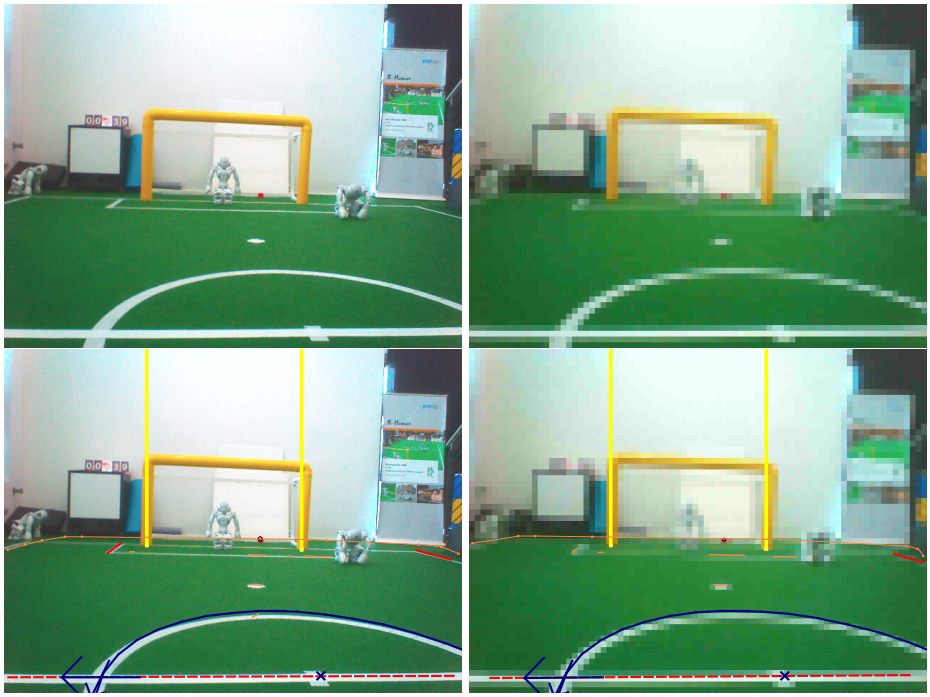


Fig. 5: Images taken by the upper camera of a NAO (original resolution on the left, corresponding thumbnails on the right).

6.1 Operating System

We have reconfigured the official Aldebaran kernel to include several new drivers and we enabled hyper-threading.

The Wi-Fi driver has been replaced by the rt5370sta driver to improve the link quality. In order to enable hyper-threading, the LAN driver had to be replaced by the Realtek r8169 driver, because the original driver crashed. In addition, we improved the official camera driver. It now supports the manual setting of white balance, hue, and fade-to-black.

The modified kernel can be downloaded from our GitHub repository.[1]

6.2 Logging

Evaluating the actions of robots during the games is an important but difficult task. For this purpose logging has been used and improved over the past years. We could log the percepts during games, or save logs including full resolution images with a robot connected to a PC. However, both methods have great disadvantages. The former offers high frame rates due to a very low computational

[1] `https://github.com/bhuman`

load and can be used on all robots during games, but it lacks the possibility of comparing the percepts with the actual images. This makes the evaluation difficult. In the second method, it is not possible to store the logs on the robots, due to full resolution images, which makes a connection to an external PC necessary. Sending the images to a computer takes a lot of time, so that the frame rates drop significantly, and it would not be allowed during actual games.

These problems have now been solved by computing thumbnails of the images and storing them in the log files. The thumbnails are computed by shrinking the images, so that each resulting pixel is the average of an 8×8 block in the original image. The resulting images have a resolution of 80×60 (upper image) and 40×30 (lower image) pixels. Although the resolution is highly reduced, it is still possible so recognize important parts on the field, as one can see in Fig.5.

6.3 C-Based Behavior

Last year, we already replaced the Extensible Agent Behavior Specification Language (XABSL) [17] by an extension of C++ [1]. As XABSL, it followed the paradigm of hierarchical state machines. It avoided most of overhead of coupling two different programming languages together, but it still required a pre-compiler to translate the behavior language into plain C++. This year, we replaced the pre-compiler by a set of C++ macros that still support a very similar syntax, but do not need a separate compiler run anymore. In contrast to the previous approach, the new language supports options with formal parameters in regular C++ notation, for which even default values can be specified. This results in easy to read behavior descriptions. Since the behaviors are actually modeled in C++, IDEs fully support the language. In addition, a new view allows to visualize the option graph directly in our simulator SimRobot.

7 GameController

At last year's RoboCup, we presented a new GameController for the Standard Platform League (cf. Fig. 6), which has become the new official game controller. It was successfully used at the RoboCup German Open as well as the RoboCup U.S. Open. For this year, we are adding support for the Humanoid League. We will also develop a NAOqi module that can be used to handle the communication with the GameController on the NAO, and it will implement the official button and LED interface. We also decided to reimplement the GameStateVisualizer, which is used to display the score to the audience, to keep the software integrated in a single code base.

8 Conclusions

B-Human has developed several major features since RoboCup 2012. The new robust vision approach reduces the need to calibrate colors before each game.

Fig. 6: Screenshot of the new GameController during a game

The new self-locator integrates functionality previously distributed over separate modules in a single, streamlined implementation. The obstacle detection using both vision and ultrasound eases realizing advanced behaviors such as passing or man-to-man marking, as well as actively controlling the arms to avoid obstacles. Our custom kernel greatly improves the network connectivity, delivers camera images as they were supposed to be, and improves the overall system performance by nearly 30%. The new logging capabilities considerably increased the overall debugging experience leading to higher code quality. Besides our code releases, the new GameController is another contribution of B-Human to the development of the Standard Platform League.

References

1. Röfer, T., Laue, T., Müller, J., Bartsch, M., Batram, M.J., Böckmann, A., Lehmann, N., Maaß, F., Münder, T., Steinbeck, M., Stolpmann, A., Taddiken, S., Wieschendorf, R., Zitzmann, D.: B-Human team report and code release 2012 (2012) Only available online: `http://www.b-human.de/wp-content/uploads/2012/11/CodeRelease2012.pdf`.
2. Wenk, F., Röfer, T.: Online generated kick motions for the NAO balanced using inverse dynamics. In: RoboCup 2013: Robot Soccer World Cup XVII. Lecture Notes in Artificial Intelligence, Springer (2014) to appear.
3. 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, Springer (2014) to appear.

4. Röfer, T., Laue, T.: On B-Human's code releases in the Standard Platform League - software architecture and impact. In: RoboCup 2013: Robot Soccer World Cup XVII. Lecture Notes in Artificial Intelligence, Springer (2014) to appear.
5. Quigley, M., Conley, K., Gerkey, B.P., Faust, J., Foote, T., Leibs, J., Wheeler, R., Ng, A.Y.: ROS: an open-source robot operating system. In: Proceedings of the Open-Source Software workshop of the International Conference on Robotics and Automation (ICRA), Kobe, Japan (2009)
6. Zickler, S., Laue, T., Birbach, O., Wongphati, M., Veloso, M.: SSL-vision: The shared vision system for the RoboCup Small Size League. In Baltes, J., Lagoudakis, M.G., Naruse, T., Shiry, S., eds.: RoboCup 2009: Robot Soccer World Cup XIII. Volume 5949 of Lecture Notes in Artificial Intelligence., Springer (2010) 425–436
7. Zickler, S., Laue, T., Gurzoni Jr, J., Birbach, O., Biswas, J., Veloso, M.: Five years of SSL-Vision - impact and development. In: RoboCup 2013: Robot Soccer World Cup XVII. Lecture Notes in Artificial Intelligence, Springer (2014) to appear.
8. Burchardt, A., Laue, T., Röfer, T.: Optimizing particle filter parameters for self-localization. In del Solar, J.R., Chown, E., Ploeger, P.G., eds.: RoboCup 2010: Robot Soccer World Cup XIV. Volume 6556 of Lecture Notes in Artificial Intelligence., Springer (2011) 145–156
9. Laue, T., Birbach, O., Hammer, T., Frese, U.: An entertainment robot for playing interactive ball games. In: RoboCup 2013: Robot Soccer World Cup XVII. Lecture Notes in Artificial Intelligence, Springer (2014) to appear.
10. Otsu, N.: A threshold selection method from gray-level histograms. IEEE Transactions on Systems, Man and Cybernetics **9** (1979) 62–66
11. Hoffmann, J., Jüngel, M., Lötzsch, M.: A vision based system for goal-directed obstacle avoidance used in the RC'03 obstacle avoidance challenge. In: In 8th International Workshop on RoboCup 2004 (Robot World Cup Soccer Games and Conferences), Lecture Notes in Artificial Intelligence, Springer (2004) 418–425
12. Thrun, S., Burgard, W., Fox, D.: Probabilistic Robotics. MIT Press, Cambridge (2005)
13. Fox, D., Burgard, W., Dellaert, F., Thrun, S.: Monte-Carlo localization: Efficient position estimation for mobile robots. In: Proceedings of the Sixteenth National Conference on Artificial Intelligence, Orlando, FL, USA (1999) 343 – 349
14. Laue, T., Röfer, T.: Particle filter-based state estimation in a competitive and uncertain environment. In: Proceedings of the 6th International Workshop on Embedded Systems, Vaasa, Finland (2007)
15. Julier, S.J., Uhlmann, J.K.: A new extension of the kalman filter to nonlinear systems. In: Proceedings of AeroSense: The 11th International Symposium on Aerospace/Defense Sensing, Simulation and Controls, Orlando, FL, USA (1997) 182–193
16. Röfer, T., Laue, T., Müller, J., Graf, C., Böckmann, A., Münder, T.: B-Human Team Description for RoboCup 2012. In Chen, X., Stone, P., Sucar, L.E., der Zant, T.V., eds.: RoboCup 2012: Robot Soccer World Cup XV Preproceedings, RoboCup Federation (2012)
17. Loetzsch, M., Risler, M., Jüngel, M.: XABSL - A Pragmatic Approach to Behavior Engineering . In: Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2006), Beijing, China (2006) 5124–5129