

Bembelbots Frankfurt

ROBOCUP SPL TEAM AT GOETHE UNIVERSITY FRANKFURT

Team Description for RoboCup 2013

Dipl.-Inf. Markus Meissner¹, Dr. Holger Friedrich², Dipl.-Inf. Andreas Fürtig¹, Dipl.-Inf. Tobias Weis², Jens-Michael Siegl¹, Christian Becker¹, Vincent Michalski¹, Gerhard Ruscher¹, B.Sc. Andreas Kehlenbach¹, Dipl.-Math. Anna Reckers¹, Konrad Zacharias¹, Eric Hanssen¹, and Alexander Heun¹

¹Goethe University, Frankfurt, Germany

²Frankfurt Institute for Advanced Studies, Frankfurt, Germany

teamleader@bembelbots.de

1 Team Bembelbots

The Joint Robotics Lab (JRL) is a joint project of three working groups of the Institut of Computer Science at Goethe University Frankfurt with the common goal to strengthen project oriented learning.

Early 2009 the JRL offered a working group, called RoboCup-AG. Within this group up to ten students started developing a robot behaviour using the simulator Webots¹. Still in the same year this initial team founded the RoboCup team *Bembelbots Frankfurt*. The whole team was very motivated already from the beginning, met several times a week and everybody spent a lot of free time to perform strongly. This was necessary as right from the start, we decided to build our own framework from scratch instead of using already existent frameworks.

Due to fast satisfying results we participated in the RoboCup German Open, already in 2010. Since that time, we took part every year, gaining a lot of experience. We also obtained new practical and theoretical knowledge through visiting open RoboCup workshops in Berlin, through playing friendly matches and exchanging knowledge with other teams. Thereby we made continuous progress improving our framework. We implemented a robust vision, a satisfying self localisation, a dynamic behaviour, an own walk and much more. As a result of our efforts we got the chance to participate in the RoboCup World-Championship in 2012 and enjoyed the experience of our first international competition very much.

During the years the team members changed but overall it grows continuously.

¹<http://www.cyberbotics.com>



Figure 1: Team Bembelbots at the RoboCup GermanOpen 2013

Team Structure: Team leader of Bembelbots is Markus Meissner, a PhD student at the Electronic Design Methodology group at the Institute of Computer Science at the Goethe University Frankfurt. He has a good overview of the team members' activities and is very avid, especially in the development of behaviour and motion. Dr. Holger Friedrich, former head of the team, has left Goethe University, but still spends a lot of his free time for the team, especially in the areas vision, framework and software development methodology. Andreas Fürtig just finished his studies and wrote his diploma thesis on colour classification [2]. He is now a dedicated PhD student at the Joint-Robotics-Lab, dealing with organisational issues and coordinates the work with the new students and research projects..

We have several senior students with strong contributions to the RoboCup. Tobias Weis has finished his diploma thesis on self localisation [3] and is still working continuously on improving the localisation,

Christian Becker also finished the work on his thesis on advanced line detection and scan-matching algorithms [4]. Jens Siegl contributes technical knowledge and maintains the robots since the beginning of the project. Andreas Kehlenbach provided the infrastructure for XABSL and works on enhancing the walk-engine. Christian Becker and Alexander Heun develop behaviours and Gerhard Ruscher concentrates on motions. Anna Reckers is the contact for mathematical problems. In addition we have several students who joined the RoboCup team in the last years. Konrad Zacharias, Eric Hanssen and four additional students recently joined the team and are still searching for their field of interest.

2 Research Interests and Planned Activities

Focusing on computer vision development in the past years, the team Bembelbots made fast progress towards a reliable image based self localisation and therefore was also able to enhance the robots behaviour concerning game strategies.

In the last years the their selves from this framework: faster walks and an overall improved agility are promising enhancements that legitimate the development of a self-made motion engine. Hence we started to disengage from the NaoQi and developed our own walk and are taking further steps to develop a completely self-contained motion engine. Gerhard Ruscher is currently working on a complete new motion engine interacting with the DCM directly in his diploma thesis.

3 Summary of Past Relevant Work

3.1 Software Framework

We developed our own software framework from scratch. It is written in C++ and consists of several modules. We compile against the NaoQi middleware, but have minimized the dependencies and wrapped library dependent calls. In addition to NaoQi we use external libraries OpenCV, libzip, Boost, and the XABSL engine. For the future we are planning to amplify the split-off of the NaoQi from our framework. The following paragraphs give an overview of the key aspects we focused on during the last three years.

²<http://code.google.com/p/googletest/>

3.1.1 Development Methodology

We have a strong focus on automated testing. Our base software framework is compiled and tested every time new code is checked into our source code management system. The unit test framework we use is googletest², and we check the coverage of our tests with a profiling tool gcov from the GNU Compiler Collection suite. Continuous Integration with Jenkins builds the whole framework on different systems everytime the code is changed and ensures a stable framework at any given time.

3.1.2 Locomotion

Walking Engine: Our first implementation for an own walk was based on [7]. As the paper proposed, we implemented a *closed-loop* walk.

However the stability of that walk was too bad for real walking. Our robots were able to dribble, but not to walk. Further tests and literature research guided us to an *open-loop* walk engine.

According to that and with the helpful assistance of Team Nao-HTWK of Leipzig we successfully developed a full parametrizable walk implementation during the last year. A joystick control has been developed to comfortably set the parameters.

Kick: We implemented three kinds of kicks: front, back and side kick. The Nao chooses the type of kick due to the implemented behaviour. In addition, each kick is parametrizable, where the robot himself computes the right parameter dependant on the ball position.

3.1.3 Cognition

Camera Like most teams, for a long time we used only one camera. Regarding the increasing fieldsize in 2013, it became more necessary to have a far-reaching view. Seeing the goal at the end of the field is as important as seeing the ball directly in front of the robots' feet. That is the reason why we decided to use both cameras and adapted the existing algorithms to use this new potential.

Classification The cognition process is mainly split into two parts. In a first step, *colour classification* is applied to the high resolution image. Classification is based on a Gaussian model describing the statistics of the colour values. A fast implementation using Lookup Tables guarantees a processing rate in real time [1]. After this step, the environment is separated into 6 different colour classes,

which can easily be processed in order to find important elements on the field. The classification step is very *robust to changes in illumination*, which typically appear in competitions or show events, and thus there is no need to calibrate the Lookup Table by hand before each game.

To deal with the new field dimensions, we decided to use both cameras of the Nao. Both images are not processed in parallel, instead we toggle the cameras in each step to save computational power. Due to a high framerate of nearly 30 processed images per second, this means we classify 15 images of each camera per second.

To minimize cache-misses, which lead to a slower processing rate, the whole Lookup Table was also compressed [2]. A result image of the classification step can be seen in Fig. 2.

Detecting Objects Based on the fact that the colour-classification provides very good results, the subsequent object recognition modules work on binary images of each colour class of interest. Mainly histograms on this binary images are used to find objects of interest, like goals and balls. This method comes with the benefit of being very robust on a small number of misclassified pixel values.

Besides simple geometry checks on the shape of the objects to be recognized, a feedback loop has been implemented relating the distance of an object to its dimension. After calibration of the robots, we achieved good results for distance measurements based on triangulation.

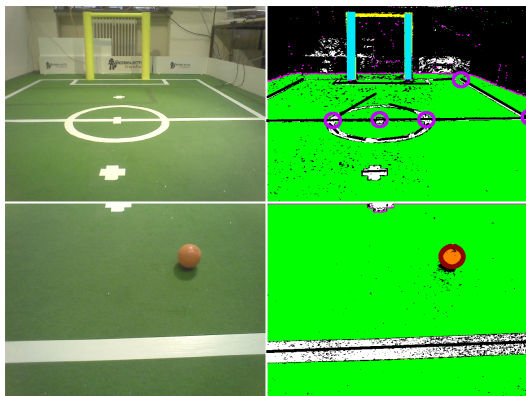


Figure 2: Result of the Dual Vision Module. *Left column:* Original Image (converted to RGB), *right column:* Classification of the input image with found vision results: A ball, yellow goalpoles and several corners, which then are processed by the localisation module.

Detecting Lines, Corners, and the Centre Circle

The colour classified images are scanned with horizontal and vertical scanlines to find green-white-green transitions. Currently we are developing a method, that not only takes colour into account

but also makes use of flanks in the illumination channel of the *YUV*-image. The scanline distance is adjusted based on the distance of image points projected onto the ground-plane, such that the distance between scanned points on the ground is always less than 5cm.

Detected line points will be connected to lines, and in a second step lines that match in space and orientation will be joined. These lines are then projected onto the ground-plane of the robot-coordinate-system. In this space, known geometrical relations between field lines are applied, easing the detection of the centre circle and the corners.

3.1.4 Localisation

During the past year our main focus of attention was the improvement of our localisation methods.

The detection of lines, corners, the centre circle and poles (as explained above) are used for computing all likely positions of the robot in the world.

The more features detected in one picture or the more expressive the detected features are, the less positions are likely (a detected goal equals two likely positions on the field, where a single L-Corner yields eight possible positions). Every possible position calculated like this is a measurement for the Kalman filter. However, the benefit of different coloured goals is no longer existent and the feasibility of following more than just one track is a crucial constraint.

The additional symmetries introduced by two goals of the same colour implicate an increasing variety of robot positions for a single measurement. The increasing number of field players further decreases the possibility of detecting strong landmarks, so Vincent Michalski and Tobias Weis implemented a Multi Hypothesis Kalman Filter and introduced two new measurements for breaking the symmetry. The use of infrared emission by the goalkeeper aids the localisation of field players by providing a strong hint about their orientation and about their field half (IR range is approx. 3m). Fused ball positions (especially with the goalkeeper) also provide useful information about the field half, and, in the case of a successfully fused team-ball, about their positions.

For cases where little information is available for localisation (e.g. only one line), we are developing a Scan-Matching approach based on [10] to provide some kind of local position estimation.

The detected field lines and other features will be matched to the known environment, given the position estimate. To minimize the lookup of which detected feature should be matched to which of the

world we use a similar approach like [11]. This will allow position correction even with very little information. Fig. 3 shows an example.

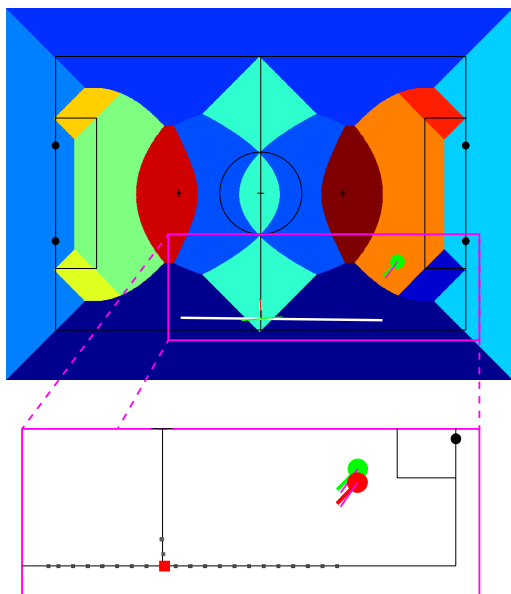


Figure 3: Top: The soccerfield underlayed with a voronoi map. Each colour corresponds to a fieldline. This map is used for lookup. White the detected fieldlines, projected to the ground at the believed robot position (green). Bottom: Magnified view after the Scan-Matching process. The grey dots represent the former detected lines. It is easy to see that the overlay is much better. Red is the corrected position on the field.

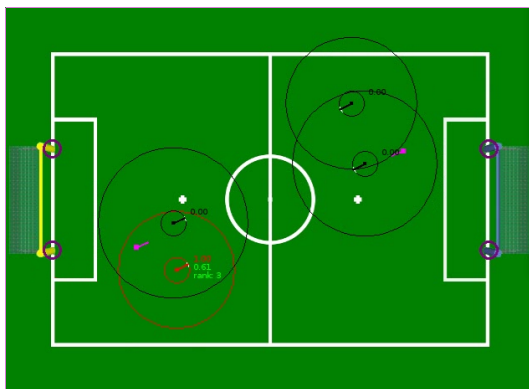


Figure 4: Live visualisation of the hypotheses of the kalman filter with according covariances. The direct result of the current measurement (goal) is shown in pink, the most likely hypothesis in red.

3.1.5 Behaviour

We integrated the XABSL engine into our framework and implemented a wrapper class, to parse arguments from the framework to the behaviour and vice versa.

3.1.6 Scaleable field size

Based on the announcement to enlarge the field in 2013 we will no longer be able to construct a full sized field in one of the institutes rooms. Therefore we restructured all modules and external tools to utilize a centralized variable field definition. This also allowed the construction of a smaller, permanent field in one of our lab rooms.

3.1.7 Isolation of our software from NaoQi

Until this year we built a big library including our software and related NaoQi functionality. At the moment we are pushing the separation of our software and the communication with NaoQi. Therefore our software will run separatly and will communicate with NaoQi via several shared memories. A lot of code can be reused from the old framework, but a lot of testing will still be necessary. Because of that, we hope to get this running until the RoboCup in Eindhoven, including a complete new debug framework and tools.

3.2 Tools

Several tools have been developed in the last years, allowing us to debug and analyse the behaviour of our Naos. Those tools are written in Python, mainly for portability purposes. They work either with Linux, Windows and partly also on mobile phones (Android). *NaoDebug* is the main application to debug the robot, for example change camera parameters or receive live images. The current state of the XABSL FSM can be displayed graphically, as well as the current state of the world model of the robots and their estimated positions in the world. This speeds up the whole development process, as errors are detected very early and easily. Our old remote tool to control the robot for demonstration purposes was merged with *NaoDebug*. An old version of the remote (*NaoRemote*) is still available on the team home page [8].

References

1. Fürtig, Andreas; Friedrich, Holger; Mester, Rudolf (2010): Robust Pixel Classification for RoboCup, Farbworkshop Ilmenau
2. Fürtig, Andreas (2011): Farbklassifizierung für humanoide Roboter im RoboCup Umfeld, Diploma thesis at Goethe-University, Frankfurt
3. Weis, Tobias (2011): Selbstlokalisierung für humanoide Roboter im RoboCup Umfeld, Diploma thesis at Goethe-University, Frankfurt
4. Becker, Christian (2013): Linienbasierte Feature-detektion und Positionstracking durch Voronoi-

- Diagramme in einer symmetrischen Umgebung ,
Diploma thesis at Goethe-University, Frankfurt
5. Rohnfeld, Erik (2010): Reinforcement Learning zum Lernen eines Schusses auf dem Nao-Roboter, Bachelor thesis at Goethe-University, Frankfurt
 6. Jäger, Moritz (2010): Entwicklung eines automatisierten Verfahrens zur Optimierung der Laufbewegung des Nao Roboters, Bachelor thesis at Goethe-University, Frankfurt
 7. Behnke, Sven (2006): Online Trajectory Generation For Omnidirectional Biped Walking
 8. NaoRemote for Nao Robot (2012): www.bembelbots.de/web/robocup/downloads/
 9. Quinlan et al.:The 2005 NUbots team report. Technical Report, The University of Newcastle, 2006.
 10. Cox, Ingemar J. (1991): Blanche - An Experiment in Guidance and Navigation of an Autonomous Robot Vehicle
 11. Whelan, T. et al (2012): Line point registration: A technique for enhancing robot localization in a soccer environment