Cerberus'14 Team Report

H. Levent Akın
Okan Aşık
Binnur Görer
Ahmet Erdem
Bahar İrfan

Artificial Intelligence Laboratory
Department of Computer Engineering
Boğaziçi University
34342 Bebek, İstanbul, Turkey
{akin, okan.asik, binnur.gorer, ahmet.erdem1, bahar.irfan}@boun.edu.tr

January 2015

Contents

Chapter 1

Introduction

The **Cerberus** team made its debut in RoboCup 2001 competition. This was the first international team participating in the league as a result of the joint research effort of Boğaziçi University (BU), Istanbul, Turkey and Technical University Sofia, Plovdiv branch (TUSP), Plovdiv, Bulgaria. The team competed in RoboCup 2001-2012 except the year 2004. Since 2005, Boğaziçi University is maintaining the team. In 2005, despite the fact that it was the only team competing with ERS-210s (not ERS210As), Cerberus won the first place in the technical challenges.

From the very beginning, Cerberus has chosen to develop all components of the software to form the basis for a more general robotics research rather than to be used for soccer only. Through the years, the members of the team have done many PhD and MS Thesis studies related to SPL and published more than 40 papers in journals and international conferences, including the RoboCup Symposia [1].

The organization of the rest of the paper is as follows. The software architecture is described in Chapter 2. In Chapter 3, the details of the vision module are provided. Self localization method is described in Chapter 4. The locomotion module and gait optimization methods used are explained in Chapter 5. Finally, various approaches we use for the planning module are described in Chapter 6.

[1] The full list of Cerberus publications are available here: http://robot.cmpe.boun.edu.tr/ cerberus/wiki/doku.php/publications

Chapter 2

Software Architecture

The overall architecture, shown in Figure 2.1, is in fact an instance of a classical *sense-plan-act* paradigm. The process starts with reading sensory information including images from the camera, readings of the inertial sensors, and current positions of the body joints. In the following sections, we explain the current state of our methods, but detailed information can be found in [1].

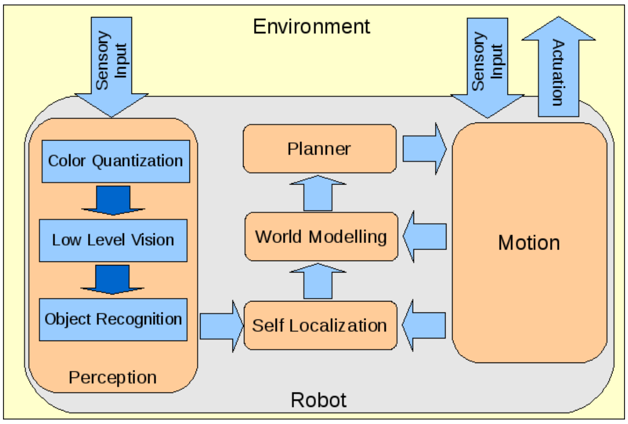

Figure 2.1: The overall control architecture of the Cerberus robot soccer team.

In order to use the API of NaoQi, we have adapted the shared memory communication infrastructure and walk engine of the B-Human's source code [2].

Chapter 3

Vision

3.1 Image Processing and Perception

The purpose of the perception module is to process the raw image and extract available object features from it.

3.2 Color Quantization

We utilize a Generalized Regression Neural Network (GRNN) [3] for mapping the real color space to the pseudo-color space composed of a smaller set of pseudo-colors, namely, white, green, yellow, blue, robot-blue, orange, red, and "ignore". In order to obtain the outputs of the trained GRNN in a time-efficient manner, a look up table is constructed for all possible inputs.

3.3 Scanline Based Perception Framework

Since the cameras of the Nao robots provide higher resolution images and its processor is slow, it becomes infeasible to process each pixel to find the objects of interests in the image due to computational efficiency and real-time constraints. Therefore, scan lines are used to process the image in a sparse manner, hence speeding up the entire process.

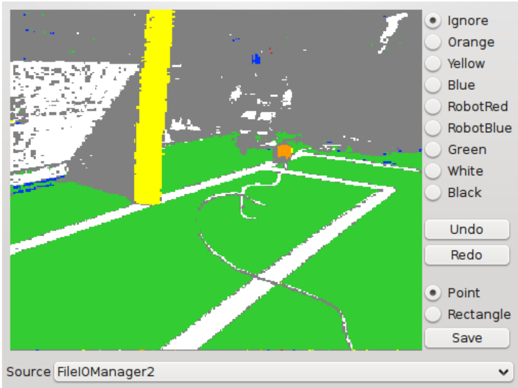

Figure 3.1: A classified image constructed with a trained GRNN.

After processing scan lines, we construct regions for the objects of interest such as goal bar, field lines, robots, and the ball. Objects are detected by processing previously constructed regions. After detection, objects are projected to frame of the robot.

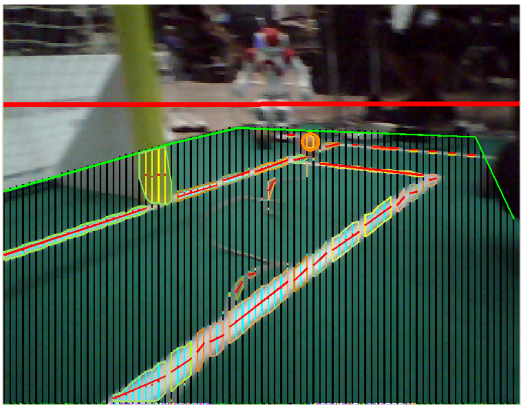

Figure 3.2: Result of processing the image using scan lines.

3.4 World Modeling and Short Term Observation Memory

The planning module requires perceptual information with less noise and in a more complete manner. The world modeling module should reduce sensor noise and complete the missing state information by predicting the state. This is a state prediction problem and we use the most common approach in the literature, the Kalman Filter [4], for solving this problem.

For any object, the observation is $z = \{d, \theta\}$ where d and θ are distance and bearing, respectively, to the robot origin. For the stationary objects, the state is $m = \{d, \theta\}$ and the state evolution model is $m^1_{k+1} = I \times m_k$ and $z_k = I \times m_k$ where k is time and I is the unit matrix.

For the dynamic objects, the observation is the same but the state is represented as $m = \{d, \theta, d_d, d_\theta\}$ where d_d is the change in distance in one time step and d_θ is the change in bearing likewise.

In the update steps, odometry readings are used. The odometry reading is $u = \{d_x, d_y, d_\theta\}$ where d_x and d_y are displacements in egocentric coordinate frame and d_θ is the change in orientation. When an odometry reading is received, all the state vectors of known objects are geometrically re-calculated and the associated uncertainty is increased.

The most obvious effect of using a Kalman Filter is that the disadvantage of having a limited field of view is reduced. As the robot pans its head, it can be aware of distinct landmarks which are not in the same field of view at the same time.

Chapter 4

Self Localization

Cerberus employs vision based Monte Carlo Localization (MCL). In the MCL algorithm, the belief state is represented by a particle set and each element represents a possible pose of the robot. We use MCL with a set of practical extensions (X-MCL) which is detailed in [5]. Until recently, we used the output of the world modeling module as input to the localization module. Namely the filtered landmarks are used as observations for the localization module. However, we now filter unidentified observations.

This approach is inspired from FastSLAM [6] algorithm and Multi-Hypothesis tracking [7]. In FastSLAM, each particle has its own world model (i.e. map). In Multi-Hypothesis tracking, there are multiple Gaussians where each relies on a different data association sequence and their numbers are limited by pruning.

To overcome the unified goal bar color problem, with the assumption that initial position of the robot is known, the model described above works with minimal change. For the kidnapping situations, we plan to develop a binary hypothesis based approach. The methodology is as follows. After kidnapping (or falling), the robot makes an assumption about side of the first observed goal bar, and perform localization normally. After that it simultaneously tries to validate this hypothesis based on robot observations, and incoming messages from teammates.

Chapter 5

Motion

For bipedal locomotion, we use two different walking engines. The first engine is developed in the lab [8, 9], we also use the omni-directional walking developed by Aldebaran Robotics [10].

Our biped walking is defined in terms of parameters some of which are mentioned above. Since balance is not guaranteed in the model and it is impossible to optimize the model with the maximum speed analytically, we apply an optimization algorithm, *Evolutionary Strategies*, to fine-tune the walking motion after determining a feasible parameter set by hand. More details about this walking engine can be found in [8].

Because our walking engine was slower than most of the teams, we started to develop another walking engine which is based-on the walking engine of the B-Human [11].

In addition to walking engines, special actions are generated as static motion like; *get up front*, *get up back*, *kick to the left*, *right*, *forward and back*, *block to the left and right*.

Chapter 6

Planner

The soccer domain is a continuous environment, but the robots operate in discrete time steps. At each time step, the environment, and the robots' own states change. The planner keeps track of those changes, and decides the new actions. The main aim of the planner is to sufficiently model the environment and update its state. Additionally, the planner should provide control inputs according to this model. Previously, we developed a market based planner and a Dec-POMDP based planner. Currently, we use a finite state machine based planner as explained in Section 6.1.

6.1 Finite State Controller based Planner

The Finite State Controller (FSC) based planner makes use of the formal model of the problem. At every planner step, the robot is in a particular state and we want our robot to take the best action in that state. FSC is based on the conventional Hierarchical Finite State Machine model, however, we changed some aspects to use it in high-level robot planning. There are states which correspond to the environment states. Transitions take place according to the current environment observations. There are also actions which will be taken when the robot is at a particular state. The robot can execute many actions in a particular state and these actions may override each other according to their priority. The most powerful part of this planner architecture is that once we code particular transitions or actions, they can be reused in different behaviors. We have developed a GUI tool called *FSC Designer* for this purpose [12]. *FSC Designer* enables easy development of finite state controller based behaviors by using already developed *Transition* and *Action* constructs as seen in Figure 6.1.

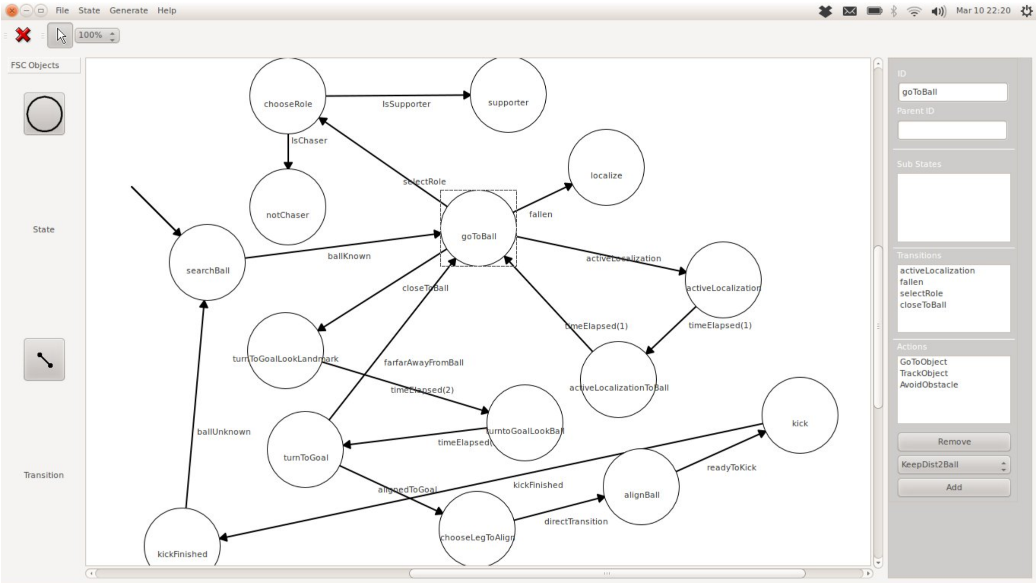

Figure 6.1: Snapshot of *FSC Designer*

6.2 Goalie Planner

The goalie has its own finite state machine controller to execute its behavior. General structure of the goalie is composed of searching ball, tracking ball, blocking ball if it is close enough and to localize itself during all of these behaviors. The localization problem to return to the center of the goal is solved partially by using corners, lines of the penalty area and odometry information. However, line and corner search and alignment processes take too much time when considering the speed of the game. The robot always has to check the ball of the position while localizing itself. As a result, we have to consider more visual sensory input to increase localization accuracy and decrease process time which has been planned as our direction in improving the performance of the goalie behavior.

Chapter 7

Conclusion and Future Works

In conclusion, we aim to compete in RoboCup SPL 2014 to test our methods in a challenging competition environment. Currently, we try to solve multi-agent planning, and kidnapping problems which are the top challenges of SPL. We perform research on the application of *Dec-POMDP* methods for robot soccer [13]. For solving the kidnapping problem, and in general the localization problem, we work on methods which effectively merge world models, and extraction and usage of dynamic landmark.

Acknowledgments

This work is supported by **Boğaziçi University Research Fund** through project **13A01P3**.

References

[1] H. L. Akın, E. Özkucur, B. Gökçe, and O. Aşık. Cerberus 2012 Team Description Paper. Technical report, Boğaziçi University, Turkey, 2012.

[2] Thomas Röfer, Tim Laue, Judith Müller, Armin Burchardt, Erik Damrose, Alexander Fabisch, Fynn Feldpausch, Katharina Gillmann, Colin Graf, Thijs Jeffry de Haas, Alexander Härtl, Daniel Honsel, Philipp Kastner, Tobias Kastner, Benjamin Markowsky, Michael Mester, Jonas Peter, Ole Jan Lars Riemann, Martin Ring, Wiebke Sauerland, André Schreck, Ingo Sieverdingbeck, Felix Wenk, and Jan-Hendrik Worch. B-human team report and code release 2010, 2010. Only available online: `http://www.b-human.de/file_download/33/bhuman10_coderelease.pdf`.

[3] Ethem Alpaydın. *Machine Learning*. MIT Press, 2004.

[4] Greg Welch and Gary Bishop. An introduction to the kalman filter. Technical report, University of North Carolina at Chapel Hill, Chapel Hill, NC, Chapel Hill, NC, USA, 1995.

[5] K. Kaplan, B. Çelik, T. Meriçli, Ç. Mericli, and H. L. Akın. Practical extensions to vision-based monte carlo localization methods for robot soccer domain. *RoboCup 2005: Robot Soccer World Cup IX, LNCS*, 4020:420–427, 2006.

[6] M. Montemerlo. FastSLAM: A factored solution to the simultaneous localization and mapping problem with unknown data association. In *CMU Robotics Institute*, 2003.

[7] S. Thrun, W. Burgard, and D. Fox. *Probabilistic Robotics*. MIT Press, Cambridge, MA, 2005.

[8] Bariş Gökçe and H Levent Akin. Parameter optimization of a signal-based omni-directional biped locomotion using evolutionary strategies. In

RoboCup 2010: Robot Soccer World Cup XIV, pages 362–373. Springer, 2011.

[9] B. Gökçe and H. L. Akın. Parameter optimization of a signal-based biped locomotion approach using evolutionary strategies. In *Proceedings of the Twelfth International Conference on Climbing and Walking Robots and the Support Technologies for Mobile Machines (CLAWAR 2009), Istanbul, Turkey*, pages 9–11, 2009.

[10] Aldebaran-Nao. http://www.aldebaran-robotics.com/eng/nao.php.

[11] Colin Graf and Thomas Röfer. A closed-loop 3d-lipm gait for the robocup standard platform league humanoid. In Enrico Pagello, Changjiu Zhou, Sven Behnke, Emanuele Menegatti, Thomas Röfer, and Peter Stone, editors, *Proceedings of the Fifth Workshop on Humanoid Soccer Robots in conjunction with the 2010 IEEE-RAS International Conference on Humanoid Robots*, Nashville, TN, USA, 2010.

[12] O. Aşık and H. L. Akın. FSC Designer: A Visual FSM Design Tool for Robot Control. In *3rd Computer Science Student Workshop (CSW'12)*, 2012.

[13] Okan Aşık and H Levent Akın. Solving multi-agent decision problems modeled as dec-pomdp: A robot soccer case study. In *RoboCup 2012: Robot Soccer World Cup XVI*, pages 130–140. Springer, 2013.