

The UPennalizers

RoboCup Standard Platform League

Team Description Paper 2017

Yongbo Qian, Xiang Deng, Alex Baucom and Daniel D. Lee

GRASP Lab, University of Pennsylvania, Philadelphia PA 19104, USA,
<https://www.grasp.upenn.edu/>

Abstract. This paper presents the current research and software system for humanoid soccer robots developed by the UPennalizers - the robot soccer team from University of Pennsylvania. The team has been involved in RoboCup for over a decade to foster the research in the areas of robotic vision, machine learning, motion control and artificial intelligence. The current software system integrates perception, locomotion and behavior modules together to enable a team of robots to play soccer autonomously. This work mainly describes the improvements made across all modules for the preparation of the RoboCup 2017 Standard Platform League (SPL) competition in Nagoya, Japan.

Keywords: RoboCup, SPL, 2017, University of Pennsylvania

1 Introduction

The UPennalizers is affiliated with the General Robotics, Automation, Sensing and Perception (GRASP) Laboratory and the School of Engineering and Applied Science at the University of Pennsylvania. In 1999, two years after the first international RoboCup, this robot soccer team was formed and began to step up to the challenges put forth by the competition. While the league was still utilizing four-legged Sony Aibos, the UPennalizers became the runner-up in 2003 and made to the quarter-final rounds every year through 2006. After taking a brief two-year hiatus in 2007, the team was reformed and returned in 2009 to begin competing in the Standard Platform League with humanoid robots NAO, taking on bipedal motion alongside improved vision techniques and advanced behavior planning. In RoboCup 2016, the team made to the quarter-finals again and was selected to the all-star team from the drop-in challenge. Starting from 2017, a number of new members joined the team, taking on more challenges in improving ball detection and lighting-invariant perception algorithm, implementing the feedback-based motion system and designing novel team strategies.

Coached by Dr. Daniel Lee, the 2017 team consists of Yongbo Qian (team leader), Alex Baucom, Xiang Deng, Daniel Pak, Abdullah Zaini, Jason Wang, Rubab Rehman and Julio Reynaga Galeas. The team members are pictured in Figure 1.

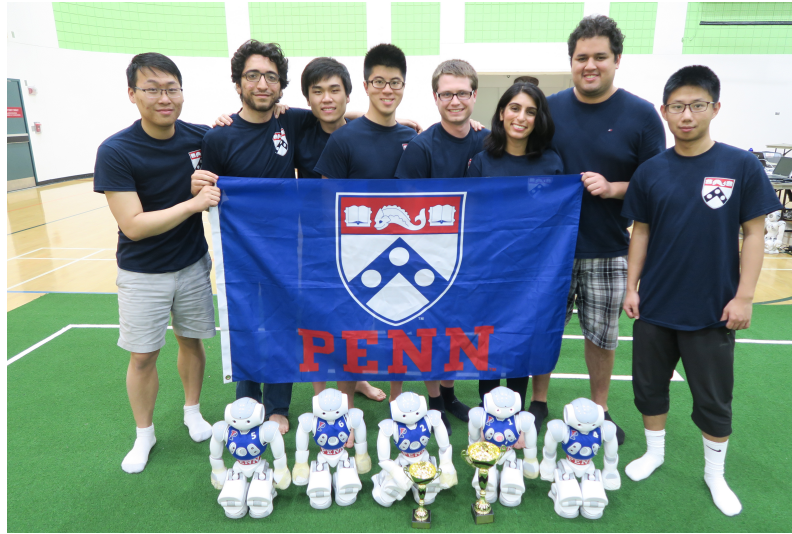


Fig. 1: The UPennalizers team after the RoboCup U.S. Open 2017. From left to right: Jason Wang, Abdullah Zaini, Daniel Pak, Yongbo Qian, Alex Baucom, Rubab Rehman, Julio Reynaga and Xiang Deng

2 System Overview

The current software system of the UPennalizers integrates perception, locomotion and behavior modules together in an efficient, scalable and cross-platform way to enable a team of robots to play autonomous soccer. The perception module handles object detection routines, recognizing the soccer ball and field features which are further utilized for localization. The locomotion engine allows omni-directional walk and uses sensory feedback to compensate for external disturbances. High-level behavior module uses finite state machines to define single robot's behavior as well as the team strategy. The overall structure of our software architecture is shown in Figure 2. Our software runs on the Linux operating system and is written in a combination of Lua and C/C++. The C/C++ routines implement the low level processing, handling interactions to motors and sensors systems in Naoqi, along with our kinematics and image processing libraries. The Lua portions interact with C/C++ device driver routines, operating the high-level vision and locomotion processes and behavior state machine. Those high-level routines are platform agnostic which could be easily plugged into different humanoid and simulation platforms [1]. We also use MATLAB for external monitoring and debugging. Detailed descriptions of our software system can be found in the previous team reports [2] [3].

This year, in order to tackle the significant challenges created by major rule changes, we have made modifications across all modules in order to improve the robustness of our robot system. The following sections will focus on our newest

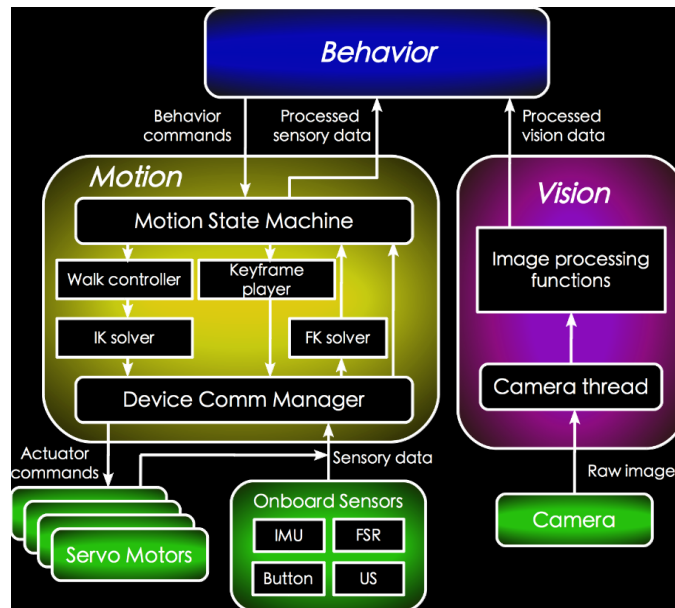


Fig. 2: Block Diagram of the Software Architecture

developments from the preparation of RoboCup 2017. The publicly released code in the 2016 version with limited behavior can be accessed via [6].

3 Perception

The perception module performs object detections on the soccer ball and field features (lines, corners and the center circle) at frame rate and projects them to the ground plane through projective transformation to calculate the relative distances to the robot. Although there are no substantial rule changes on those objects in 2017, we still continue working on advanced vision algorithms to improve our robot's ability to perceive the surroundings in real-time.

3.1 White Ball Detection Modification

The RoboCup 2016 introduced the new white and black soccer ball which posed some challenges to our current ball detection algorithm. Although the detection was reliable for most cases, it was still not robust enough to recognize the ball if it was overlapped with other white objects, such as lines and robot's feet. Currently, the detection was designed to search through white blobs in circular shape with reasonable size, and then run checks on each of those blobs to see if they have the pattern of the black patches similar to that of what the real ball would have. If the ball was close to white surroundings, the system was not able

to detect the white circular blob in the first place. In order to solve these edge cases, a reversed approach from inside-out was developed. For example, if the ball was not detected through the original approach, the algorithm would search on lines for the desired pattern of the black pixels, and grow a bounding box for white pixels outwards from the centroid to fit a circular shape. Figure 3 shows a successful ball detection using this approach when the ball is on a line.

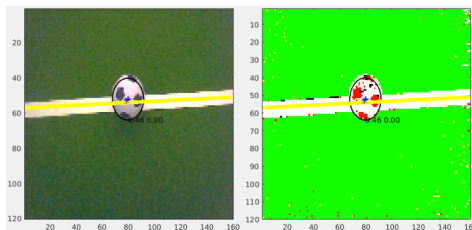


Fig. 3: Ball detection on a line

In addition, in order to further reduce the false positives, instead of having a single sanity check for ball's size for each top and bottom camera image, we evaluated ball's size with respect to its position in the image. This could filter out some incorrect candidates such as large ball close to the top of the image or small ball close to the bottom of the image.

3.2 Lighting Invariant Perception

Since the league is moving towards the natural lighting condition, we started to develop some techniques for lighting-invariant perception. In a preliminary work [4], we proposed an approach that projected the 3-dimensional color space of the image pixels to a lower dimensional representation which could preserve the separability of different color classes, while discounting the effect of illumination variation.



Fig. 4: Accurate green classification under natural light

Specifically, we first transformed the RGB color space using log chromaticity projection, and then performed discriminant analysis with kernels on the chromaticity feature space to classify different color classes. Figure 4 shows this approach is effective for field color detection under strong sunlight.

We also tested an online self-supervised learning approach on the field color detection. As depicted in Figure 5, when field occupancy became low because of lighting changes, samples from above robots horizon (non-green) and samples on the field with relatively high green probability (green) from previous training phase would be used as new training samples to re-perform our discriminative analysis. In this way, the projection direction for the dimensionality reduction was modulated over-time which could produce improved classification result adapting to the current lighting condition.

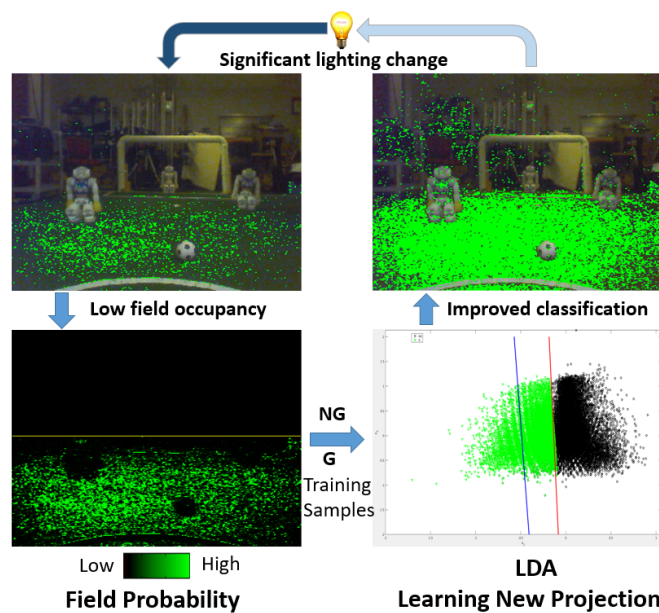


Fig. 5: Proposed self-supervised learning approach pipeline

3.3 Localization

In 2016, with proper integration of the new yaw axis gyroscope sensor value of Nao V5 and improved field features detection, our rao-blackwellized particle filter based localization system was accurate enough for most of the times during the game. However, the gyro value is subject to drift over time, especially when our robots fall over too many times. This caused the robot to be 90 degrees off in orientation since we mostly relied on line updates to correct the yaw error.

This year, by including more visual features in the measurement model, such as corners and field boundaries detected from the top camera, our robot was able to track its orientation very accurately, even after several falls or spinning with no need of using goal-post as a visual feature. This also allowed teammates to trust the location of each other much more, which helped our team coordination module described in the next section.

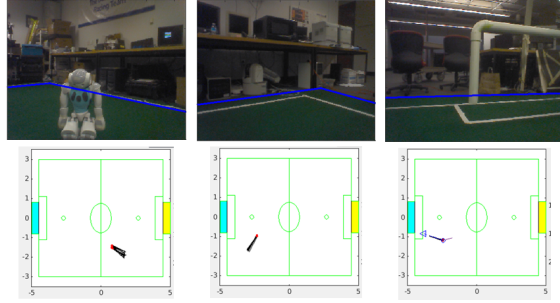


Fig. 6: Using field boundaries for localization. Details could be found in [5]

4 Team Coordination

4.1 Behavior

Since our ball detection was not robust enough when it is far away from the robot and overlaps with a line, we developed several team behaviors aim to have our robots search and track the ball as a team. A team search state was implemented and added to our behavior state machine. If all the robots have performed local search for a while yet still failed to detect the ball, they would be further spread out to cover more areas on the field by going towards a closest location within several manually defined ones where the ball is more likely to be placed after throw-in.

Another method we developed is creating a team ball score where each player would score the likelihood of every players observation of ball candidates. Once the team ball was determined, all the robots would turn to face this ball and behave accordingly based on roles. Through this way, the robots shared their ball knowledge within the team, and their local searches would be biased to this most agreed ball. We also used team ball information for effective role assignment - all role switching computations were carried out using every players ETA to the team ball, even if they observed a different ball candidate. This ensured the team consensus for the role switching and all roles would be fluidly rotated and filled.

4.2 Position Strategy

Robots intelligent positioning on the field is another important aspect of team coordination. Inspired by the recent success of Googles AlphaGo, we decided to use policy gradient method to learn a strategic positioning scheme via simulation self-play [4]. We created a MATLAB simulation (Figure 7 left) that could run a 10 minute game of 10 robots in about 20 seconds. The framework was designed to easily be able to change player behaviors and configurations in order to optimize parameterized policies automatically. We started by combining a set of simple heuristics to create an artificial potential field function (Figure 7 right) that the player would traverse using local gradient descent. This approach had the advantage of combining the path planning and the positional strategy as well as incorporating a set of clearly defined weights that could be tuned to produce very different behaviors. In order to keep the search space small, we also looked for approaches that could minimize the number of function evaluations required, namely the NelderMead algorithm and layered learning strategies. Training one role at a time has shown some promise of being computationally feasible.

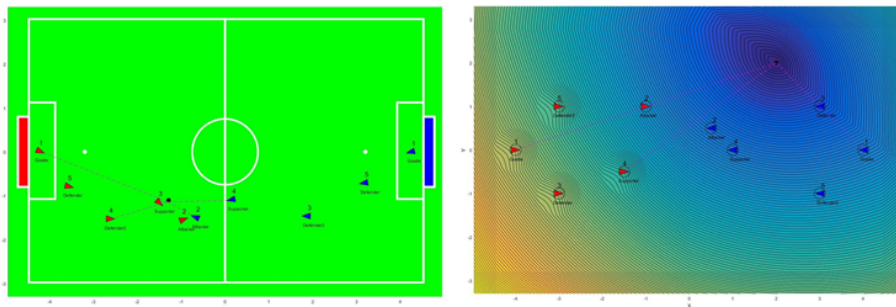


Fig. 7: The game simulator and potential field overlay for the red attacker

5 Locomotion

For the first time, a more realistic field covered with astroturf will be used in RoboCup 2017. This change has made locomotion a lot harder for small and lightweight robots. In order to adapt to the new terrain and further stabilized our robots during the game, some improvements have been made on our self-developed locomotion module.

5.1 Terrain Adaptation

We aimed to achieve a simple adaptation without directly measuring the external forces applied to the contact points or acquiring the dynamics model of the

new terrain. We formulated controls which adjusted the interactions between the foot and the terrain in two cases: stance phase and swing phase. More specifically, during the swing phase, we used heel/toe-off compensations to reduce the chances of colliding with ground in most cases; during the stance phase, we superimposed a stiffness control function which better compensated the intrinsic compliance of the artificial grass. On the other hand, we used a simple approach to measure the stability of the walking over the new terrain: the distribution pattern (mean and variance) of robot's center of mass (CoM) with respect to its foot steps. Our stability measurements allowed us to iteratively tune the parameters of terrain adaptation.

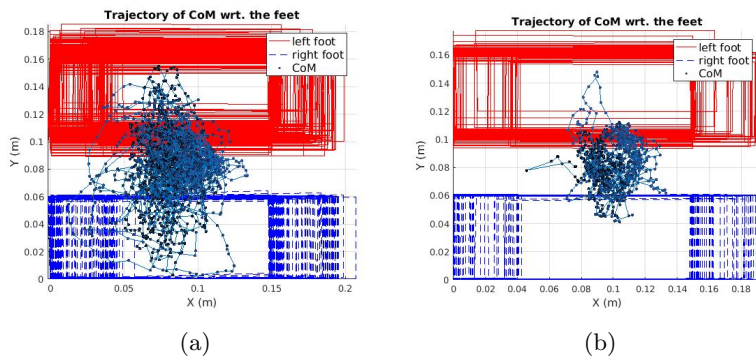


Fig. 8: Comparisons of center of mass distributions during 10 seconds walking over the artificial grass, before (left) and after (right) the locomotion's adaptation to the new terrain.

5.2 Walk Kick and Stabilization

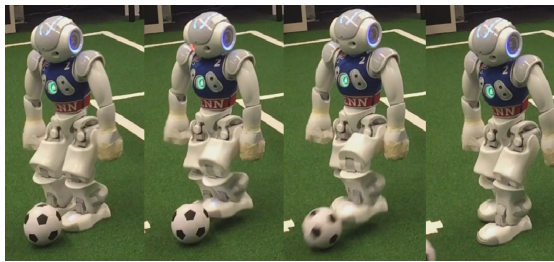


Fig. 9: Walk kick with one-step stabilization

This year we emphasized on developing a unified framework combining the kicking with regular walking cycle. This allowed our robot to kick the ball fast without the extra time to execute the kick motion. In addition, we developed a stabilization routine which allowed the robot to naturally perform one step or multiple steps stabilization. The stabilization routine was applicable to both kicking and walking. Based on IMU measurements, we formulated a feedback control loop that can dynamically determine the position and velocity of new foot steps which were used to better stabilize the torso.

6 Research Interest and Future Work

While improvements made throughout this year formed the backbone of our strategies to remain competitive in the new environment, it is clear that novel techniques will be needed in the near future for reliable play under additional evolutionary rules changes. Some of our short-term research focuses are as follows:

Perception:

- Utilize additional features like edge and geometry so that the robots can play without reliable color cues under the natural lighting conditions.
- Implement efficient deep learning methods for ball and robot detection
- Exploit vision information from frame by frame differences

Locomotion:

- Improve the efficiency of stiffness control
- Combine IMU and motion key frames for a dynamically stabilized getup module

Team Behavior:

- Optimize the team behavior to limit robots' actions strategically to compensate the overheating problem
- Develop adversarial decision making for both individual and team strategies based on the opponents' actions
- Design a more extensive communication protocol to support advanced techniques for multi-robot coordination in any-sized team

For long term research goal beyond the scope of competition, we are interested in developing intelligent systems that can adapt and learn from experience. We will focus on learning representations that enable robots to efficiently reason about real-time behaviors and planning in dynamic environments to guide the decision making and reduce uncertainty.

7 Conclusion

The RoboCup SPL is a tremendously exciting community to be a part of. The international competitions, aggressive technology development and compelling motivations fostered an environment that brought out a great level of effort from all team members who were involved this year. Although the improvements made to our software system outlined in this paper should be capable of meeting the requirements for RoboCup 2017, there is still much left to do to elevate our team to the higher level. In addition, beyond preparing for the annual RoboCup competitions, we hope to apply our soccer playing advancements in humanoids to further impact other domains.

8 Acknowledgment

The 2017 UPennalizers team gratefully acknowledges the strong and consistent support from the GRASP Laboratory and the School of Engineering and Applied Science at the University of Pennsylvania. We also wish to thank the past team members' significant contributions to the code-base and the organization.

References

1. McGill, S., Yi, S., Zhang, Y., Lee, D.: Extensions of a RoboCup Soccer Software Framework. RoboCup 2013: Robot World Cup XVII, Volume 8371 of the series Lecture Notes in Computer Science pp 608-615 (2013)
2. Qian, Y., Baucom, A., Han, Q., Small, A., Buckman, D., Lee, D.: The UPennalizers RoboCup Standard Platform League Team Description Paper 2016. RoboCup Symposium 2016 (2016)
3. Qian, Y., He, Y., Han, Q., Poudel, S., Small, A., Lee, K., Lee, D.: RoboCup 2015 Standard Platform League Team Description Paper. Technical report (2015)
4. Qian, Y., Baucom, A., Lee, D.: Perception and Strategy Learning in Robot Soccer. The 11th Workshop on Humanoid Soccer Robots, 16th IEEE-RAS International Conference on Humanoid Robots (2016)
5. Qian, Y., Lee, D.: Adaptive Field Detection and Localization in Robot Soccer. RoboCup Symposium 2016 (2016)
6. <https://github.com/UPenn-RoboCup/UPennalizers>