The UPennalizers RoboCup Standard Platform League Team Description Paper 2016

Yongbo Qian, Alex Baucom, Qiao Han, Austin Small, David Buckman, Zheng Tian 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 developed on Aldebaran Nao robots by the UPennalizers - University of Pennsylvania Robot Soccer Team. The team has been involved in RoboCup for over a decade to foster the research in the areas of computer vision, machine learning, motion control and artificial intelligence. The current software system integrates perception, locomotion and behavior modules together to enable robots play soccer autonomously. This work mainly describes the improvements made across all modules for the RoboCup 2016 Standard Platform League (SPL) competition in Leipzig, Germany.

Keywords: RoboCup, SPL, 2016, 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 University of Pennsylvania. In 1999, two years after the first international RoboCup, this robot soccer team was formed and began stepping up to the challenges put forth by the competition. While the league was still utilizing four-legged Sony Aibos, the UPennalizers made the quarterfinal rounds every year through 2006 before taking a brief two-year hiatus in 2007. The team reformed and returned in 2009 to begin competing in the Standard Platform League with Aldebaran Nao robots, taking on bipedal motion alongside improved vision techniques and advanced behavior control. In 2016, with most of old the team members graduated, a relatively new team was formed to take on more challenges in recognizing white balls and goal posts, designing novel team strategies as well as improving the localization and locomotion systems.

Coached by Dr.Daniel Lee, the 2016 team consists of Yongbo Qian (team leader), Alex Baucom, Qiao Han, Austin Small, David Buckman, Zheng Tian, Andrew Wang, Quinn Wu and Cameron Deering. Most of the team members are pictured in Figure 1. The team made it to the quarter-finals this year in RoboCup and was placed 6th in the drop-in competition.

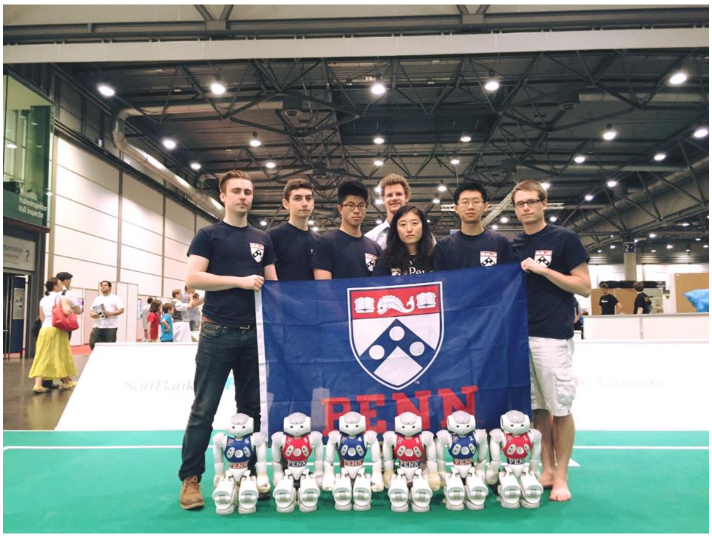

Fig. 1. The UPennalizers team after the RoboCup 2016. From left to right: Austin Small, David Buckman, Yongbo Qian, Stephen Mcgill, Qiao Han, Zheng Tian and Alex Baucom

2 System Overview

The current software system of the UPennalizers integrates perception, locomotion and behavior modules together to enable robots play soccer autonomously. The perception module detects landmarks on the soccer field and utilizes them to localize the robot. The locomotion engine allows omni-directional motions 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 team coordination. Detailed description of our software architecture can be found in the previous team reports [1] [2].

This year, in order to tackle the significant challenges posed 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 developments in the preparation of RoboCup 2016. The publicly released code with limited behavior can be accessed via [5].

3 Vision

One of the biggest rule changes in 2016 is that the major color of the ball has been changed to white. Now, along with white goalposts, supporting structures and net, as well as white field lines, robots and massive environmental noise, this color feature became extremely unreliable for previous detection approaches. In order to handle this change, improved algorithms for object detection were developed.

3.1 Adaptive Field Detection

As green becomes the only unique color cone we can leverage, wisely using it to detect static field becomes an important first step in perception, because it could provide contextual information for other objects. An approach for adaptive color classification that uses histogram analysis from parts of previous images was implemented to identify the field color. This method is robust under inconsistent lighting conditions which could potential utilized in outdoor games in the future. The field color is then used to find potential field boundary points. As shown in Figure 2, these points are input for a convex hull and a line fitting to find the field boundary. The detailed implementation of this method can be found in [3].

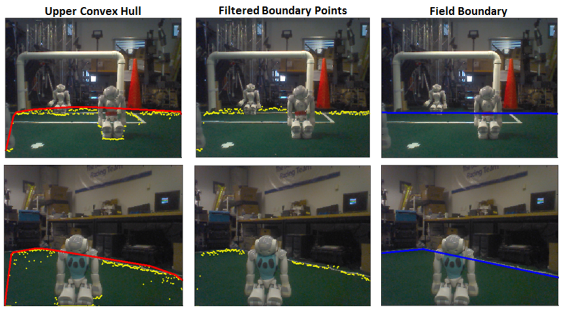

Fig. 2. Field boundary detection follows the sequence of building upper convex hull (left), filtering raw boundary points (middle) and line fitting using RANSAC (right).

3.2 White Ball Detection

Our current white ball detection algorithm is still based on the color segmentation technique. We introduced a new color class red to be trained on the black patches on the ball using the manually labeled color-table. Our ball detection is then designed to search through white blobs, and run checks on each of those blobs to see if they have properties similar to that of what the white blob of a real ball would have. The algorithm has strict checks for the size - the ball should be close to the desired projection size in different areas of the image; fill rate - the bounding box of the ball candidates should be mostly filled by white and red pixels; black ratio - there should be a certain amount of red pixels on the white ball, and green check - the ball should be surrounded by sufficient green pixels in all four directions. A successful ball detection example using colortable is shown in Figure 3 (top).

There are some edge cases that cannot pass these checks. For example, if the ball is on a line, the white blob of ball maybe hiding within the blob of a line. So our ball detection algorithm also searches between two separate lines to find if there are some characteristics that match the real pattern of the ball.

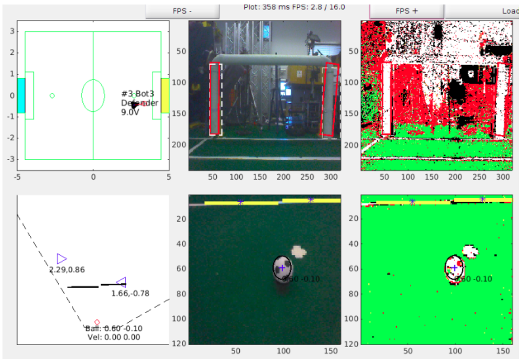

Fig. 3. Goalpost detection (top) and new ball detection (bottom)

3.3 White Goalpost Detection

The improvements were made on further utilizing geometry features and exploiting field context. For geometry features, strict checks on size, orientation, aspect ratio of the bounding box were performed to exclude some false positive goalposts from random white noise. For field context, since the goalposts only vertically grow onwards the field, we checked if the goalposts are perpendicular and mostly above the field boundary line. This helps to distinguish goalposts from white field lines as well as robots' arms. Figure 3 (bottom) also shows the successful detection on white goalposts.

3.4 Field Features Detection

Field features such as spot (penalty cross), center circle, corners, and long field lines are useful landmarks for robot's self-localization. The detection for those field features has been slightly modified. Building up on our initial line detection algorithm, we first check if the line segments could form a circle. Since the circle always appears with the center line, the intersection between the normal of each line segments and the center line is calculated. If the standard deviation of those intersection points is close to zero, those line segments can be labeled as center circle, and the average of intersection points can be consider as the center of circle. Figure 4 (top) plots out the center of circle calculated by those line segments on the center circle .

Before performing corner detection, the labeled circle lines are removed before performing the corner detection. If two field lines projected to the field intersect under a nearly perpendicular angle and the end points of both lines are close to the intersection, then those two lines are classified into a corner, which is shown in Figure 4 (bottom).

Finally, if the line cannot be classified as either circle lines or corner lines, we then find the best, longest line as the field line to pass into the localization module.

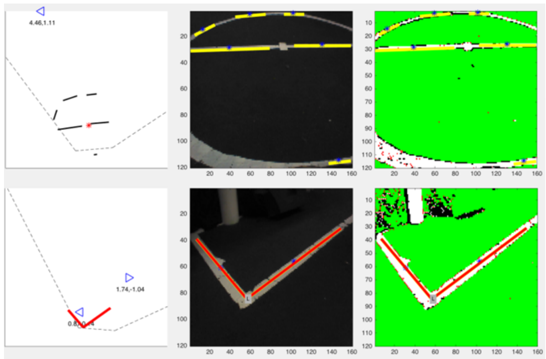

Fig. 4. The top left image marks the center of the circle based on line distribution; bottom images show a corner has been detected

4 Localization

Previously, our localization system did not use the yaw axis gyroscope of the Nao, since not all of our robots were V5. When the robot fell down, it would reset the orientation of its localization particle filter and use visual features to determine the new orientation. Due to the symmetry of the field, this would often cause the robot to become 'flipped' and score own its own net.

We have since updated all of our robots to V5 so all of them can track their yaw using the built in gyroscope. Since this yaw value is subject to drift over time, the localization system has been updated to estimate the yaw error within the particle filter. When new visual features are detected, the yaw error is adjusted slightly to correct for any drift that was introduced. This allows the robot to track its orientation very accurately, even after a fall.

In addition, considerable effort was put into ensuring that the initial position of the robot is always known with as much certainty as possible. When setup before a match, all robots now have a set position on the field to initialize to. Manually placed robots have the particles of their filter initialized only along the penalty box and just outside the center circle, depending on the kickoff team. Robots returning from penalty keep track of their orientation during the penalty phase to determine which side of the field they are entering on and initialize their particles appropriately.

Since the detection of white goal posts was not as reliable as before, the weights of corner, circle and field lines were increased in the measurement model. The definitions of corner and circle were modified to become objects with orientation, so that they can also correct both position and orientation of the robot. The position and angle of corner are defined to be the global coordinates of the vertex and the angle of the bisector, while the center circle has a position of the center point and an orientation of the estimated center line.

All of these measures ensure that the robot rarely has the opportunity to become 'flipped' and that it is more confident in its localization than in previous years. This has enabled much better team positioning in the ready and set phases and has allowed teammates to trust the location of each other much more.

5 Team Coordination

In order to facilitate better team behavior, a few changes were made to the team communication this year. Teammates now check all possible sightings of the ball and collectively decide which ball is the most likely candidate. If multiple robots see the same ball in the same location, the weighting for that ball goes up dramatically. This means that even if a player hasn't seen the ball, but a teammate has and everybody agrees that is the most likely candidate, the player who hasn't seen the ball can still base their actions on the current location of the ball.

This 'team ball' can then be used to make more meaningful dynamic role changes. Since each player knows where the 'team ball' is and where its teammates are, the player can evaluate the position of every other player relative to the ball. This means that the player can consider how its ETA to the ball compares to its teammates and make decisions based on that, instead of on only its own estimation of where the ball is. This allows team play to transition smoother as the ball moves around and lessens rapid role switching when false balls are detected. Furthermore, instead of keeping searching for the ball, the behavior of the defenders can be much less active in order to save the processing power.

6 Locomotion

The locomotion is controlled by a dynamic walk module, which at its core models the Naos center of mass (COM) trajectory using a linear inverted pendulum model (LIPM). Although this approach was used in our walking engine in previous years, the old system was not modular, making it difficult to incorporate state-of-the-art techniques. As a result, we chose to redevelop our locomotion system.

Our behavioral system controls path planning, and sends the locomotion system a series of accompanying footsteps. For each footstep, the locomotion system generates 3D trajectories for the Naos COM, as well as for both feet. With these trajectories, the Naos joint angles can be determined with inverse kinematics (IK) and sent to Naos motor buffers for execution. Although we havent incorporated closed-loop feedback yet, the modular nature of our system will accommodate these intended improvements.

In our approach, the COM trajectory for the first single support phase was first computed. Because the robot is initially stationary, an alternate approach was used to compute the COM trajectory leading into this first LIPM phase. A bezier curve was used to model this initial trajectory, linearly increasing the COM speed from 0 to the initial speed when entering the first LIPM phase. In order to determine the motion for subsequent single-support phases, position and velocity matching (the same approach applied with the bezier curve), are used to solve for the pendulum parameters of those subsequent phases. This approach completely avoids the use of a double-support phase. While computational complexity increases in comparison to more traditional approaches, a smoother motion results, greatly improving the stability of the walk engine.

7 Sound

We integrated pitch detection into behavior in order to react to the whistle at the beginning of each game. We used the AL sound library from Aldebaran's Nao and were able to read pcm buffer values from the hardware during the game. Since the sound command that the robot needs to detect is fairly simple, we used basic auto-correlation method to analyze the sound waves and determine the period and thus frequency of the sound readings. We evaluated both the frequency and volume to detect correct whistling. That is, the frequency has to be consistent over a set period of time and consistent with the expected frequency and the volume of the sound has to be louder than certain threshold. Because the whistle pitch can change slightly depending on the different hardware, we added adjustable parameters to the configuration so that these values can be calibrated easily before the game.

In terms of behavior, sound detection runs on a separate parallel thread and checks only when the robot is receiving set signal from the Game Controller. Once a whistle sound is heard, the robot goes into play state and this overwrites the set signal that it keeps receiving from Game Controller until the game is officially in play state.

8 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 and additional evolutionary rules changes. Since the end of RoboCup 2016, we started our development to step up to the new challenges posed in RoboCup. Some of our research focuses are as follows:

Lighting Invariant Vision:

- Use Linear Discriminant Analysis (LDA) on 2D chromaticity feature space to find out the optimal separability between different color classes, with an online self-supervised learning scheme to correct color classification overtime. The detailed algorithm is described in [4].
- Utilize some additional features like edge and geometry so that the robots can play without reliable color features under the natural lighting conditions.

Locomotion:

- Fully integrate with omni-directional closed-loop feedback controller to improve the robot's stability.

Team Behavior:

- Use artificial potential field functions and simulation self-play to learn better positioning strategies [4].
- Design a more extensive communication protocol to provide advanced techniques for multi-robot coordination

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.

9 Conclusion

The RoboCup SPL is a tremendously exciting community to be a part of. The international competition, aggressive technology development and compelling motivations fostered an environment that brought out a great level of effort from all team members who were involved. Although our software system outlined in this paper met the challenges in RoboCup 2016, there is still much left to do. Future implementations must use more than color cues, as finely trained color classifiers cannot be expected to work in the context of a real outdoor soccer field. Additionally, walking strategies must include models of more realistic field materials in order to remain stable. Team behavior must also meet higher standard for the team to remain competitive.

10 Acknowledgment

The 2016 UPennalizers team gratefully acknowledges the strong and consistent support from the GRASP Laboratory and the School of Engineering and Applied Science at University of Pennsylvania. We also wish to thank the past team members' contribution to the code-base and the organization.

References

1. 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)
2. 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)
3. Qian, Y., Lee, D.: Adaptive Field Detection and Localization in Robot Soccer. RoboCup Symposium 2016 (2016)
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. https://github.com/UPenn-RoboCup/UPennalizers