SEU-UniRobot Team Description Paper

Chuan Liu[1], Hui Shi[1], Zhenxin Liu[1] and Yingzi Tan[1]

[1] RoboCup Research Group, Southeast University, Nanjing, China
449091542@qq.com

Abstract. This paper presents the general hardware and software designs of the RoboCup Kid-size division team SEU-UniRobot for the RoboCup 2019 competition in Sydney, Australia. A series of improvements will be discussed with an emphasis on robust field line identification and more accurate self-localization. In the end we will describe some problems we are working on.

1 Introduction and Preceding Participations

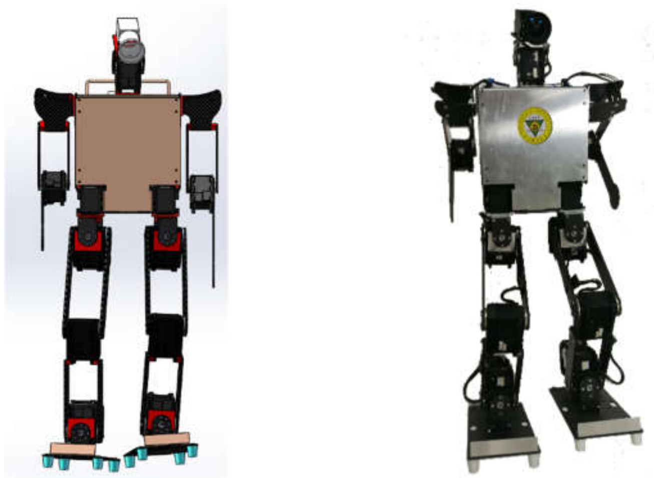
SEU-UniRobot is a robotic project whose team members are master degree candidate and undergraduate students at Southeast University. We have participated in the 2016 competition in Leipzig and the 2017 competition in Nagoya. Unfortunately, we missed the 2018 competition in Montreal because the competition date clashed with our final exam. Now it is our third time to participate in this competition. We hold a sincere belief that we can make more progress and spare no effort to contribute to this community.

2 Overview

The robots developed by SEU-UniRobot for RoboCup 2019 are fully autonomous humanoid robots, with the capability to play varied parts as a team in the competition. Fig.1 shows the robot. And the Table.1 just tells the general specification of our robots. More details will be introduced in following sections.

Table 1. Robot Specification

Weight	4.0 kg (Including Batteries)
Height	630 mm
Velocity (Forward)	0.25 m/s
CPU Board	Jetson TX2
OS	Ubuntu 16.04 64bit
Interface	RJ45 ×1, USB ×2, Push button ×2
Servo motor	MX-28R ×6, MX-64R ×8, MX-106R ×4
Battery	Infinity (14.8V 2600mAh), XingChen (12V 2000mAh)



(a) Robot model (b) Real robot

Fig. 1. Robot of SEU-UniRobot

3 Hardware System

3.1 Hardware framework

This year, we update our hardware of robot. Fig.2 shows the hardware framework. We use NVIDIA Jetson TX2 as the main controller instead of a mini-PC with a We microcontroller, and all the devices are connected to the main controller.

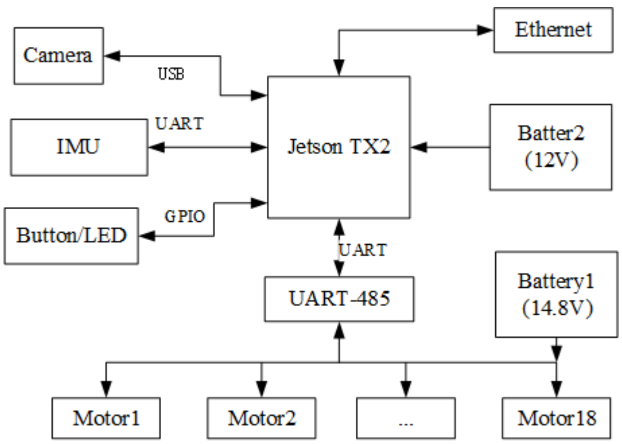


Fig. 2. Hardware framework

3.2 Sensors

Camera: MV-UBS131GC USB camera with surface exposure. It samples pictures with a resolution of 640x480 pixels with a frequency of 30 Hz.
IMU: JY-61P. It is a 6 DOF module with accelerometer and gyro, and it use TTL to communicates with the controller.
Joint positions: The robot uses joint positions feedback provided by the DYNAMIXEL motors in each joint.

4 Software System

4.1 Architecture

Fig.3 shows the architecture of our software system. It is mainly composed of following parts: perception, strategy, motion controller and actuator. The perception part gets information from sensors, teammates and game controller. Images are captured by a USB camera and processed in the computer board. Useful information would be sent to the World Model as the sharing parts among processes. Based on the information, the main controller will generate some command. The commands about the motion will be parsed in motion controller, according to the feedback data of joint and IMU, motion controller will generate some actions and transport them to the actuators, while other commands will be acted in actuators directly.

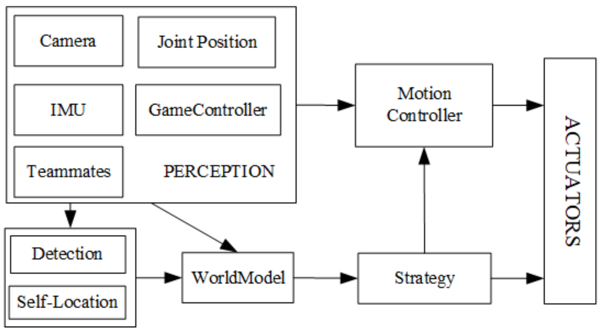


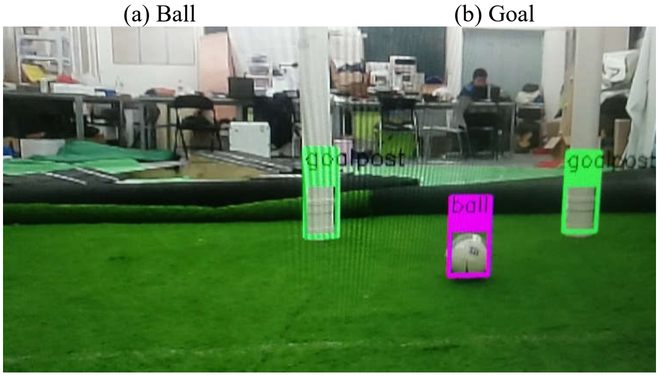
Fig. 3. Software architecture

4.2 Ball and goal detection

In the last years, we use Haar-like features and adaboost classifier to detect the ball. The goals are detected according to their color and shape. But this year, because of the powerful compute capability of Jetson TX2, a neural network named yolov3-tiny based on the darknet [1] was used to detect the ball and goal. It can run at the speed of 20 FPS. Fig.4 shows the results of this method.

(a) Ball

(b) Goal



(c) Ball and goal

Fig. 4. Ball and goal detection using yolov3-tiny

4.3 Field recognition

In this paper, the convex collocation-based field recognition algorithm proposed in the literature [2] is used. The algorithm in the literature is mainly applied to the green carpet field. Considering the difference between carpet and artificial turf, this paper uses an offline color table method to complete the green discrimination [3]. Using the color table to record the grass color features in both directions, it is better to solve the problem that the grass color is different. Fig. 5 shows the results of this method, and the area in the white line at the edge of field is the field.

Fig. 5. Field recognition

4.4 Localization

Our localization is based on the Monto Carlo method and a particle filter [4]. Given an initial state, current position is checked consistently by examining the landmarks got in the image processing part. The possible positions are shown as particles. Those particles have the information of both position and direction. As the robot moves, the particles spread, indicating that uncertainty increases. As the robot stops, particles quickly converge to get a relatively accurate position and pose. The effect has been shown in Fig.6.

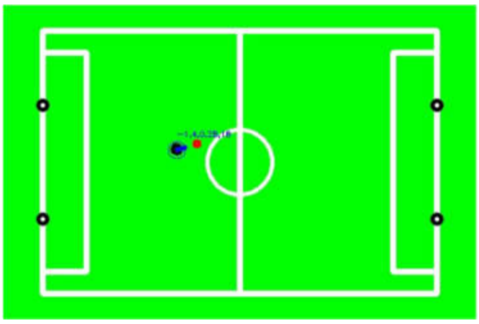


Fig. 6. Localization effect

4.5 Strategy

We use finite state machine shown in Fig .7 for strategy control. External state machine is responsible for robotic start-up, standing up, kicking the ball and localization. Internal state machine is responsible for searching the ball, posture adjustment and dribble.

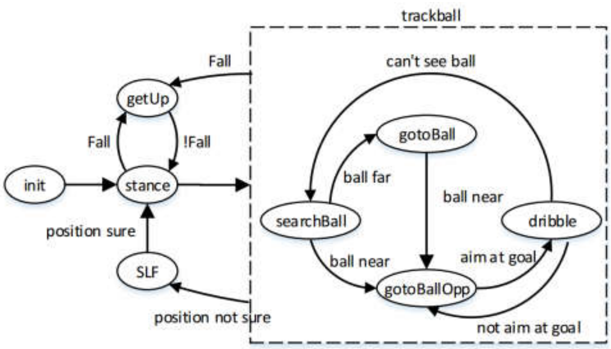


Fig. 7. Finite state machine

4.6 Motion Control

Trajectory tracking was used to produce omnidirectional gait for biped robots last year. But it's an open loop walk engine. This year, we close the loop on the basis of this gait. Firstly, we make a feedback based on the real joint angle form motors so that each step can reaches the target position as much as possible. Secondly, CoM balance is added to the robot's walk system, according to the exact position of each limb at every moment in the process of walking, which can be calculated from the true angle of the robot joint. What's more, five-time curve interpolation is introduced to the trajectory of walk engine in order to make the trajectory curve smoother. In total, the gait of our new robot is more stable and stronger.

For some complex actions such as kick and get up, we developed a software with Qt5 and OpenGL to generate the actions based on the interpolation in the key frames of an action. The UI of this software is shown in Fig .8.

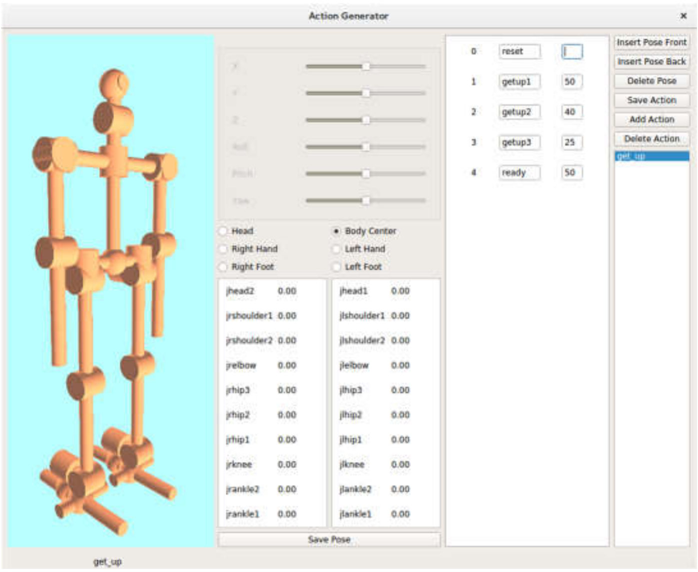


Fig. 8. Action generator

5 Conclusions

As shown in the previous section, our main improvements are platform, detection and localization. We use a new method to detection the ball and goal. And special attention should also be drawn to the field recognition. Due to the incoordination between motion plans issued from control system and actual action performed by robot, upper strategy and perception system suffers a lot, resulting in an unsatisfactory result. However, we are still busy preparing for the competition to be held in Sydney, Australia, we believe further improvements will be made.

References

1. Redmon, Joseph and Farhadi, Ali. YOLOv3: An Incremental Improvement. arXiv, 2018.
2. Qian, Yongbo, and D. D. Lee . Adaptive Field Detection and Localization in Robot Soccer. *Robot World Cup* Springer, Cham, 2016.
3. Thomas RÖfer. Region-Based Segmentation with Ambiguous Color Classes and 2-D Motion Compensation. *Robocup: Robot Soccer World Cup Xi, July, Atlanta, Ga, Usa* DBLP, 2008.
4. Nagi, Imre , W. Adiprawita , and K. Mutijarsa . Vision-based Monte Carlo localization for RoboCup Humanoid Kid-Size League. *International Conference on Control Automation Robotics & Vision* IEEE, 2015.