Does JoiTech Messi dream of RoboCup Goal?

Yuji Oshima*, Dai Hirose*, Syohei Toyoyama*, Keisuke Kawano*, Shibo Qin*,
Tomoya Suzuki†, Kazumasa Shibata†, Takashi Takuma‡ and Minoru Asada*
* Dept. of Adaptive Machine Systems, Graduate School of Engineering, Osaka University, Osaka, Japan
† Dept. of Mechanical Engineering, Faculty of Engineering, Osaka Institute of Technology, Osaka, Japan
‡ Dept. of Mechanical Engineering, Osaka Institute of Technology, Osaka, Japan
Email: robocup@er.ams.eng.osaka-u.ac.jp

***Abstract*—We focus on two critical features in big size robot soccer. First, efficient training is one of the important issues for RoboCup teams to adapt their robots on site. Real robot's training takes time and may damage the hardware while simulated one may miss the important factors to be improved. Second, to predict behavior of opponent robots is necessary for big size robots because they cannot move quickly so far. To overcome the issues we developed a versatile simulator and a smart strategy to improve accuracy of shoot and block. The former consists of multiple channels of inputs such as a real live video image stream, a real but already recorded one, and virtual one. The two kinds of outputs are real robot motor commands and virtual ones. We can have any combination of input/output to reduce the amount of training (less damage to real robot hardware) while improving the robot behavior. The latter, a strategy, consists of opponent recognition and adaptive action planning. The opponent (goalie or shooter) is detected by simple background subtraction, and its behavior is predicted based on the velocity in the image. This enables a shooter to find the space to shoot a ball, and for a goalie to block the goal from the opponent player's shooting. We successfully applied the system to the RoboCup 2013 humanoid league adult size, and got the championship as well as the best humanoid award ("Louis Vuitton Cup").**

I. Introduction

Team JoiTech participated in the soccer humanoid adult size league at RoboCup 2013. The team comprised students from Osaka University and Osaka Institute of Technology. We have participated in the RoboCup Japan Open and the RoboCup world competition each year since 2010. Our team was originally derived from RoboCup team JEAP, who participated in the humanoid league kid size competitions since 2006. Our team name, JoiTech, is an acronym for "**J**EAP and **O**saka **I**nstitute of **Tech**nology," but it also means "joint team with Osaka Inst. of Technology" and "enjoy technology." This was the third humanoid adult size world RoboCup soccer competition. The adult size league has two critical differences compared to the other leagues (kid size and teen size).

First, large and heavy robots cannot run for long periods because of the high load on their motors. It is also expensive to prepare spare robots. Thus, it is preferable to minimize the need for testing of large robots, which limits the development of code for these robots.

Second, adult size robots are required to strike the ball and block the goal. The rules of the adult size game differ from those of the other size leagues, because it is based on penalty kicks from human soccer. The adult size game involves an offensive player and a defending goalkeeper, and each side is limited to five penalty attempts. Shooting or blocking failures are not options in the championship. In recent years, the behavior of goalkeepers has become more important because many robots in the adult size league can score goals successfully. However, current adult size robots cannot move rapidly or drop down to block the goal. Thus, we developed a practical strategy for the goalkeeper given the constraints on the robot's motions, which most teams had not considered.

TABLE I. Tichno-RN hardware specifications

Tichno-RN		
Height (mm)	1500	
Weight (kg)	25	
DOF	22	
Actuators	VS-SV410, VS-SV1150, VS-SV3310	
Camera Type	iBUFFALO BSW20KM11BK	
Controller	Main Controller	Sub Controller
CPU	Panasonic Let's Note Intel Corei5 2.6GHz	VS-RC003HV ARM7TDMI LPC2148
ROM	223 GB (SSD)	512 KB
RAM	8 GB	40 MB
OS	Windows 7	None

We developed systems to solve the two critical problems encountered in the adult size league. This paper is organized as follows. We provide the hardware specifications in section II. In section III, we present an overview of the software and an object recognition system based on other software systems. We also describe the development environment used to solve the first problem, as well as the strategies used by the striker and the goalkeeper to address the second problem. The final section presents the results achieved by our team in the competitions and we summarize how our strategies performed during the games.

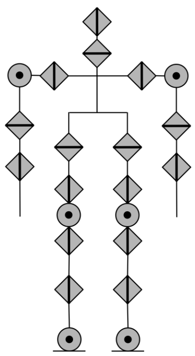
II. Robot Hardware

A. Hardware

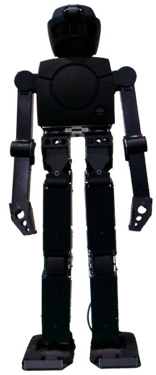
In this section, we explain the hardware specifications for Tichno-RN, the mechanical structure of which was developed by Vstone Co., Ltd [1]. A front view and schematic overview are shown in Fig. 1. The detailed specifications are given in Table I.

Tichno-RN has 22 degrees of freedom (DoFs) as shown in Fig. 1(a). The legs and arms each have six DoFs and four DoFs, respectively. The structure and powerful electric motors generate a strong torque of about $32\mathrm{N}\cdot\mathrm{m}$, which allows our robot to squat down and turn rapidly.

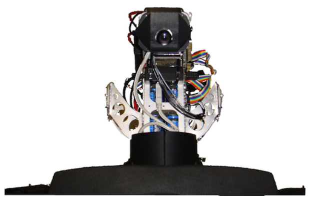
Each actuator has a micro-controller with sensors that detect the angular position of joints, temperature, and speed,



(a) DOF configuration



(b) The front view



(c) The camera

Fig. 1. Tichno-RN: (a) schematic overview of the actuators and the configuration of the body, (b) front view of the whole body, (c) camera mounted on the robot.

and which transmits the sensor information to a sub-controller. A camera is used as the eye of the robot and it has a wide angle view of 120 degrees. The camera position is adjusted to keep its own toes and the top of the goalposts within the visual field. Tichno-RN can squat down, stand up, hold a ball, and throw it, which are required to make a throw-in. Only the JoiTech robot succeeded in performing a throw-in during the technical challenge.

B. Control System

Tichno-RN has two controllers: a main controller and a sub-controller. The main controller has an advanced processor that allows object recognition and action decisions. This system uses a commercially available notebook with sufficient capacity for image processing. The main controller allowed us to develop a system without any special micro-controller programming skills. The sub-controller perceives information obtained from gyro and speed sensors in the actuators and it controls all of the actuators. The sub-controller stores the motor routines such as shooting and throw-in. The sequence of motor commands that comprise the motor routines are sent to the motors after the sub-controller receives an order from the main controller, and it subsequently receives sensor information related to the speed and angle in real time.

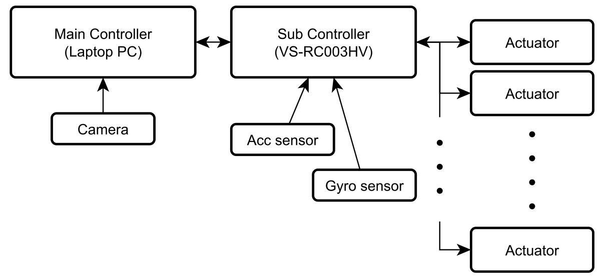


Fig. 2. Control system architecture: the main controller has an advanced processor for object recognition and action decisions. The sub-controller receives accelerator, gyro, and motor sensor information. The sub-controller sends a sequence of motor commands for one of the motor routines to the motors after it receives an order from the main controller.

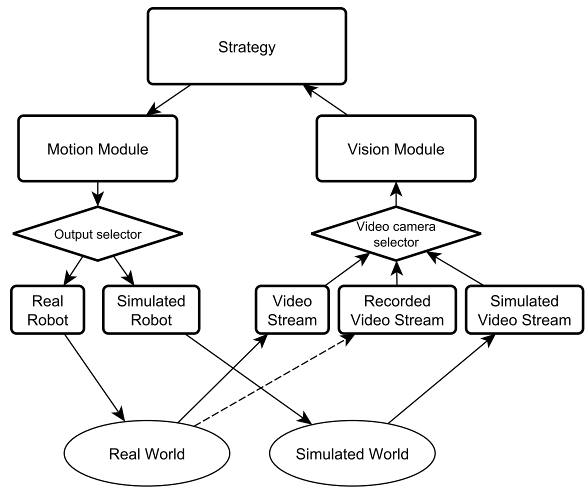


Fig. 3. System Architecture: The video stream and recorded video stream are produced from data captured by the real robot camera. The simulated video stream is produced by a simulator. The output selector selects a real or simulated robot. One output and one input can be selected. For example, it is possible to select recorded real world video data as the input and the simulated robot as the output.

III. Software

Fig. 3 provides an overview of the software used by the main controller. This software was implemented using C++ and we utilized OpenCV [2] as the image processing library. The operating system was Ubuntu 12.04 LTS 64bit.

The main program comprises three units. A strategy unit decides the robot's behavior based on sensory information. The motion module converts a behavior or motor routine into motor commands and initiates the motion. We define motions as motor routines such as walking or kicking. The vision module is an image processing unit. We also produced a simulation to facilitate efficient debugging.

The processing flow was as follows.

1) Recognize environmental information using the vision module.
2) Decide an action using the strategy.
3) Execute the motions using the motion module.

In the following subsections, we describe the image processing procedure, the strategies used by the attacker and the goalkeeper, a method for generating motions, and the test environment.

A. Method for Generating Motions

We use RobovieMaker2 to create the motions, which is a program developed by Vstone Co. Ltd. Fig. 4 shows the development environment of RobovieMaker2. Each value on

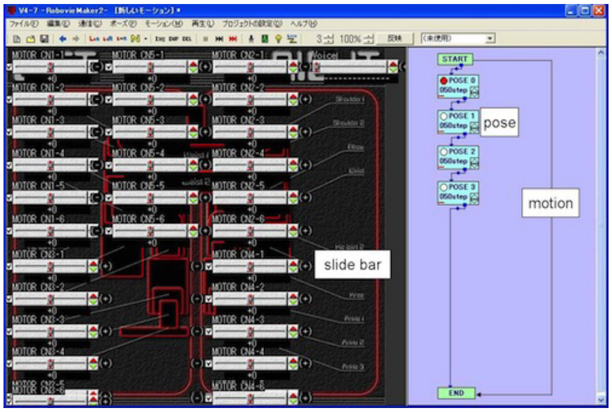


Fig. 4. RobovieMaker2: *left*, each slider corresponds to the joint angles of motors; *right*, flowchart showing an example of a motion.

the slider bars corresponds to the joint angles of the motors. The robot's poses or postures are generated using these slider bars. To create motions, we link a position to the next position and specify the time between them. Fig. 4 shows a motion flowchart on the right. RobovieMaker2 simplifies the process of producing motions. We produced 11 motions, including kicking and throw-ins. The walking motion we employed was developed by Vstone Co. Ltd.

B. Image processing

The vision module is used for image processing and it detects the positions of key objects such as the ball and the goal. The vision module recognizes the field, the ball, lines, and obstacles.

An example result of the visual processing is shown in Fig. 5. The vision module extracts the biggest green region and considers it, filtered from noise, as the field area. The vision module assumes that objects such as the ball, goal, lines, and obstacles are in contact with the field area. The ball, lines, and obstacles are recognized based on the colors and shapes of the objects.

Goal recognition is important for shooting. Due to height of the robot The robot's camera captures a large area, including space outside the field area, which can lead to a false detection of objects. Therefore, we developed a strict goal recognition strategy. We specified the goal configuration as two vertical lines and a horizontal on the field area. However, this strict recognition method sometimes missed the goal because of image blurring when the robot was moving. To address this problem, we separated goal recognition into two phases: detection and tracking. First, the robot detects the goal lines using the Hough transform [3] and recognizes the goal as lines that meet the field. This recognition method prevents the false detection of objects outside the field. Second, the robot tracks the goal using a particle filter [4] [5] based only on color until the goal is outside the camera image. This method allowed the robot to detect and track the goal in a stable manner, even while it was moving.

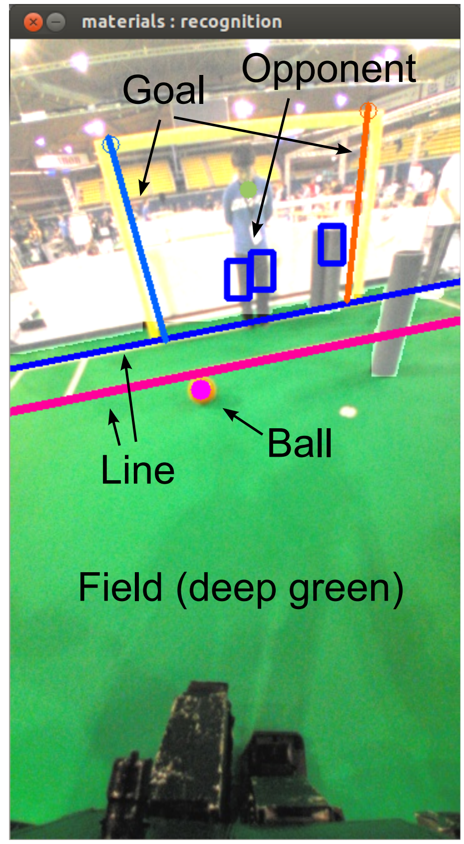


Fig. 5. Example of the results obtained with visual processing.

C. Test Environment

Any motions made by an adult size robot put a load on its motors and it is difficult to prepare spare robots or motors. Thus, it is desirable to minimize the number of trials when testing the software using a real robot.

To address this problem, we developed two test environments: a simulator and video recording/replay.

Combinations of these environments provided flexible and convenient systems for debugging the software. Next, we describe the simulator, the video recording and replay system, and combinations of these tools, which we refer to as the "versatile robot simulator."

1) Simulation: We created a simulation environment for software testing that does not require a real robot. Fig. 6 shows a snapshot captured by the simulator. This simulator was implemented using Open Dynamics Engine [6] and it reproduced the minimum set of game elements, i.e., the field, robots, lines, black obstacles, and a ball. We obtained the parameters for the simulated robot (e.g., moving speed, turning speed, and kicking strength) based on the movements of our real robot. We added noise to the parameters to test the robustness of the programs.

Our simulator aimed to test the software used to generate the game strategies (and visual recognition), rather than improving the robot motions. Consequently, our environment did not simulate walking or other motions in detail.

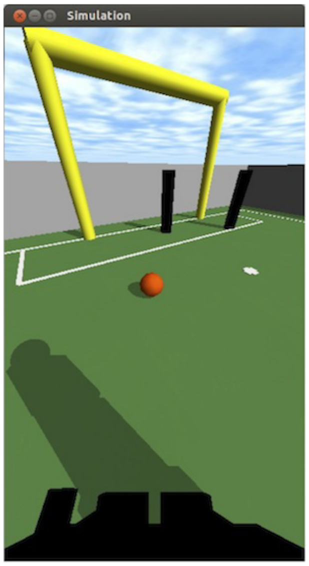


Fig. 6. Snapshot of the simulation. The black pole in front of the goal represents the opponent goalkeeper.

2) Vision Recording and Replay: Our simulator was a convenient tool for software testing, but it lacked visual realism. Therefore, we created a visual testing system to record the camera data captured by a real robot during testing and we streamed the recorded data to the vision module. This system allowed us to test the vision software repeatedly using the real robot's camera data without operating the real robot. This system was more advantageous than real robot tests when debugging some errors in specific situations (e.g., the robot lost the goal in a particular position) because the tests utilized exactly the same data and were repeatable.

This system was used during the development phase and in the competition. For example, we analyzed the color information in the field over time and considered the recorded video data. The real competition movies were useful for analyzing the strategies used by opponents and making adjustments before the next game.

3) Versatile Robot Simulator: We could select the camera on the real robot, the simulated camera, or recorded video data as the input for the vision module, and the real robot or the simulated robot as the output for the motion module. Each combination (3×2) produced a different test environment for a specific purpose. The combinations are described below.

- Simulated camera + simulated robot:
  The strategies used by the robot could be tested in a completely simulated world. The real robot was never required. However, it was necessary to make the test similar to a real environment, because the visual information in the simulator was very different from the real visual information. This test environment could be run in parallel with the software.

- Recorded video data + simulated robot:
  This combination allowed testing using previous real visual environments without damaging the robot. This was useful for fixing reproducible bugs because the recorded video could be used repeatedly. This test environment could be run in parallel with the software.

- Real camera + simulated robot:
  This was useful for detecting bugs because we could change the camera image by moving the video camera while watching the behavior of the robot in the simulator.

- Simulated camera + real robot:
  This is able to test in visual situation with real robot even impossible situation in real world. These environments were not valid for testing for competition.

- Recorded video data + real robot:
  It tests almost same parts of system as Recorded video data + simulated robot. These environments were not valid for testing.

- Real camera + real robot:
  This was the most basic test environment, which allowed testing in exactly the same environment as during the competition. We performed some tests in this environment, but they could damage the robot.

D. Strategies Used by the Attacker and the Goalkeeper

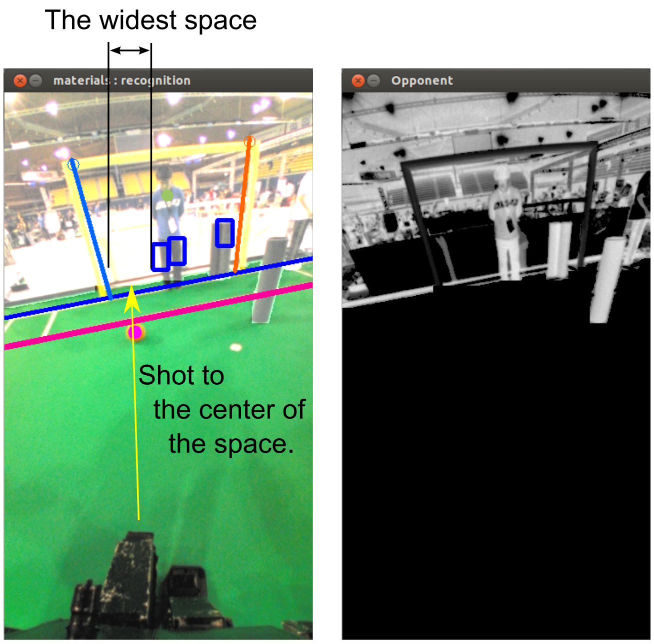
There are only limited opportunities for shooting in the adult size league. The likelihood of successful shooting and blocking are increased if the robot can dribble a ball and shoot it at the goal. Thus, we developed fundamental strategies for our robot, as follows.

1) Opponent Detection and Deciding the Shooting Direction: Recently, robots have been required to shoot accurately at the goal while avoiding the goalkeeper. In the 2013 competition, the robots also had to avoid a black obstacle placed in front of the goal. Next, we describe the strategy used by our robot, which allowed the robot to shoot at the goal accurately while avoiding the black obstacle and the goalkeeper.

1) Detect the location of the black obstacle using object recognition and the opponent robot by background subtraction (Fig. 7). Then the robot detects the positions of the goalposts by goal recognition.
2) Calculate the widest distances between the goalposts, the opponent robot, and the black obstacle.
3) Turn towards the center of the widest space and kicking the ball.

Using this method, the robot could kick the ball in the space with the highest likelihood of scoring a goal. Background subtraction allowed our robot to avoid the opponent robot, regardless of its color.

2) Goalkeeping strategy: Previously, the strategy used by most adult size robot goalkeepers was standing upright in the goal area without moving. However, the blocking success has become important because of the increased number of successful shots in the championship. Thus, we developed the following strategy (Fig. 8).



(a) Deciding the shooting direction. (b) Image showing background subtraction, where the goalkeeper is highlighted.

Fig. 7. Goalkeeper recognition.

1) Detect the opponent striker using background subtraction, and detect the approaching of the opponent striker by the distance between the striker and the ball.
2) Predict the direction of the ball that the opponent striker will kick by positional relationship between the striker and the ball. For example, if the opponent striker is located left of the ball, the opponent striker more likely to kick to right.
3) Move to the predicted direction.

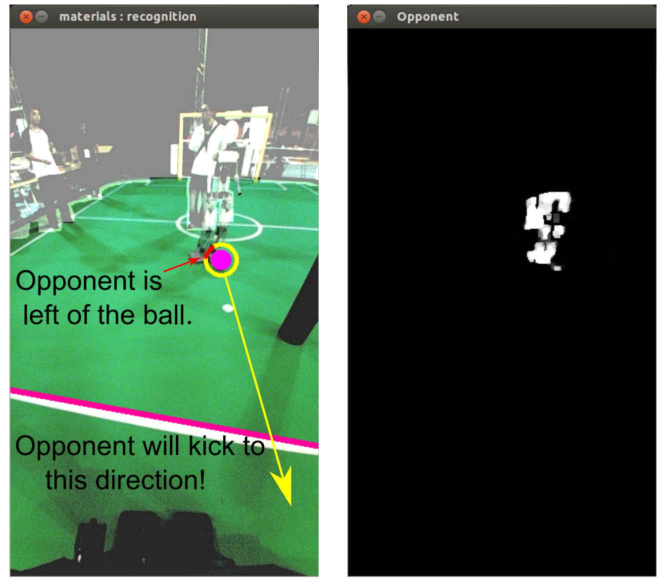
Using this method, the goalkeeper was expected to make a rapid movement to make a block because the time between the prediction and approach to the ball was short. However, the walking speed of our robot was not sufficient to successfully make a block, depending on the speed of the opponent robot. To address this problem, we developed another goalkeeping strategy to decide the direction the goalkeeper moved in, which depended on the current position of the opponent striker and that of the black obstacle at the start of the game.

IV. Results

Our results in the competitions are summarized in Table II [7].

The ratio of successful shots was only 24% because our vision system was not adjusted adequately in the round robin tournament. In the first trial of round robin 4, our robot successfully blocked the shot of the opponent HeuroEvolution AD. However, our robot failed to block any shots in the remaining trials, because the opponent team changed their attack strategy.

In the semifinal and the final, the ratio of successful shots improved to 80% (Fig. 9), because the robot adjustments were



(a) Detection of the opponent robot and the shooting direction. (b) Image with background subtraction, where the opponent is highlighted.

Fig. 8. Opponent recognition.

completed. Our robot made a successful block in the semifinal and the final (Fig. 10). Our ratio of successful blocks was only 27%, but HeuroEvolution AD's ratio was 0%. HeuroEvolution AD's ratio of successful shots throughout all of the competition was 80%, which highlights the importance of successful blocks in our championship victory.

V. Conclusion

This study explained the technical strategy used by JoiTech to win the RoboCup 2013 humanoid league adult size championship. We had to solve specific problems to win the RoboCup soccer humanoid adult size championship. The first was a limitation of the number of times that testing could be performed, which was mainly due to hardware constraints. The second was that the number of shots was limited by the rules, which required successful shooting and blocking. We developed a versatile simulator to solve the first problem, which comprised six different input and output combinations. The virtual world with a recorded data stream allowed us to conduct testing in near-real environment without damaging the robot. This drastically reduced the number of tests on required the real robot. With respect to the second problem, predicting the direction in which the robot and the opponent robot kicked the ball improved the accuracy of shooting and blocking. These two improved the likelihood of successful

TABLE II. Results in the competitions

Match	Opponent	Result
Round Robin 1	Tsinghua Hephaestus	1-0
Round Robin 2	ROBIT	2-0
Round Robin 3	EDROM Adult Size	1-0
Round Robin 4	HeuroEvolution AD	1-4
Round Robin 5	Tech United Eindhoven	1-0
Semifinal	Tsinghua Hephaestus	4-1
Final	HeuroEvolution AD	4-3

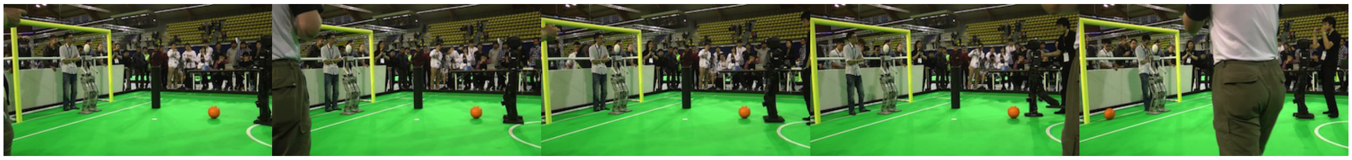


Fig. 9. Shooting during the competition: the robot turned to the left because the opponent and the black obstacle were to the right of the goal.

Fig. 10. Blocking during the competition: the robot moved to the left when the opponent was close to the right side of the ball.

shooting and blocking until the final match. Thus, we won the championship and the best humanoid award (”Louis Vuitton Cup”).

The versatile robot simulator was useful for testing at the strategy level, but it could not be used for testing at the motion level (e.g., for throw-ins). It is difficult to simulate real robot motions because of the complex computations required, such as modeling contact with objects. In most cases, the robot's motions were not created by a simulator, but instead were refined by trial and error using real robots. Kawai et al. [8] proposed a method that can reduce the number of trials when adjusting robot motions. We could produce a system with a lower computational load to refine the robot motions by adding this method.

Recently, the quality of the adult size competition with simple rules has improved drastically. In the next step, the adult size robots will be required to be more dynamic and to use a flexible strategy like the other size league, so the rules will be closer to actual human soccer. Thus, it will be necessary to develop a more flexible and general system to ensure successful shooting and blocking in the future.

ACKNOWLEDGEMENT

The authors gratefully acknowledge the contributions of the JoiTech team members. This work was supported by a Grant-in-Aid for Specially Promoted Research Number 24000012.

REFERENCES

[1] Y. Oshima, D. Hirose, J. Park, Y. Okuyama, R. Sumita, D. Akutsu, T. Suzuki, K. Shibata, T. Takuma, K. Tanaka, H. Mori, and M. Asada, “JoiTech Team Description,” in *RoboCup 2013 Symposium papers and team description papers CD-ROM*, 2013.

[2] G. Bradski, “The OpenCV Library,” *Dr. Dobb's Journal of Software Tools*, 2000.

[3] P. a. Hough, “Machine Analysis Of Bubble Chamber Pictures,” *Conf.Proc.*, vol. C590914, pp. 554–558, 1959.

[4] G. Kitagawa, “Monte Carlo Filter and Smoother for Non-Gaussian Nonlinear State Space Models,” *Journal of Computational and Graphical Statistics*, vol. 5, no. 1, pp. 1–25, 1996. [Online]. Available: http://dx.doi.org/10.2307/1390750

[5] N. Gordon and D. Salmond, “Bayesian state estimation for tracking and guidance using the bootstrap filter,” *Journal of Guidance Control and Dynamics*, vol. 18, pp. 1434–1443, 1995.

[6] R. Smith, “Open dynamics engine,” 2008, http://www.ode.org/. [Online]. Available: http://www.ode.org/

[7] “Robocup 2013 eindhoven,” retrieved August 30, 2013 from http://www.robocup2013.org/.

[8] Y. Kawai, J. Park, T. Horii, Y. Oshima, K. Tanaka, H. Mori, Y. Nagai, T. Takuma, and M. Asada, “Throwing Skill Optimization through Synchronization and Desynchronization of Degree of Freedom,” *RoboCup 2012: Robot Soccer World Cup*, vol. XVI, pp. 178–189, 2013.