JEAP Team Description

Ryuzou Nakata[*1], Keita Ogawa[*1], Hiroki Komatsu[*1],
Junji Suzuki[*1], Akiko Nakatani[*1], Naoto Nishijima[*1], Toshiyuki Homma[*1],
Masaki Ogino[*1], and Minoru Asada[*1,2]

[*1] Adaptive Machine Systems, Graduate School of Engineering
[*2] JST ERATO Asada Project Synergistic Intelligence
Osaka University, Osaka, Japan
robocup@er.ams.eng.osaka-u.ac.jp

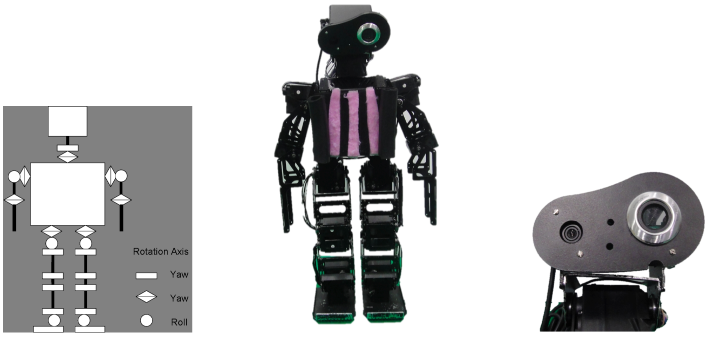
Abstract. This article describes the current status of the humanoid soccer team JEAP which utilizes the commercially available humanoid platform Vision 4G with customized parts. A modular software architecture consists of three modules: vision module, motion module and behavior module. And now, we are developing self-localization system.

1 Introduction

Team JEAP originated from team *Senchans*[2] that participated in competitions of the humanoid league since 2002, the first year of the humanoid league. In 2006, the team changed its name to JEAP[3] acronym for JST ERATO Asada Project [web: www.jeap.org], a research project supported by the Japanese Science and Technology Agency (JST). A main goal of this project is to understand the cognitive development process of humans based on synthetic approaches with humanoids. The project adopted Vision 4G as a research platform, and the RoboCup humanoid league is a good tested to enchance the capabilities of congnitive function. One of the research issues is advanced dynamic biped walking. In this article, we describe hardware specifications in section 2. The software design is given in section 3. The self-location system is given in section 4 .

2 Robot Hardware

Since the RoboCup 2007, we utilize the VisiON 4G robots. They are fully autonomous robot, manufactured by Vstone Co.,Ltd.. The motors of the VisiON 4G robots are developed in order to generate the stronger torque stably. The covers of the motors are made from aluminum and help the dissipation of the motor heat. Therefore, they can stay in action during a game without breaking down. We use Logicool Webcam Pro 9000 camera to capture high resolution images. Their front view and schematic overview are shown in Fig. 1. A VisiON 4G robot has 22 degrees of freedom and pan-tilt cameras. Its detailed specification is given in Table 1.



(a) DOF configuration (b) Appearance (c) Camera

Fig. 1. The VisiON 4G robot: On the left side is a schematic overview of the actuators and their attitude in relation to the bodies. The center and the right side figures are the front view of the whole body and the head part of the robot, respectively.

Table 1. VisiON 4G hardware specifications

VisiON 4G		
Height (mm)	465	
Weight (kg)	3.2	
DOF	22	
Actuators	VStone Servo	
Camera Type	Logicool Webcam Pro 9000	
Controller	Main Controller	Sub controller
CPU	PNM-SG3	VS-RC003 ARM
ROM	16GB (Flash HDD)	512 KB
RAM	512 MB	40 MB
OS	Linux	None

3 Modular software environment

The software of the robot consists of three modules: which are a vision module, a motion module and a behavior module. Fig.2 shows the overall system of the software. These three modules execute processing in parallel during a game.

3.1 Vision Module

We developed a library for the image processing as a vision module. Since the luminance and pixel values of colors sometimes change because of the shadow and lighting condition, it is necessary for robot to know the range of color variation. For this purpose,

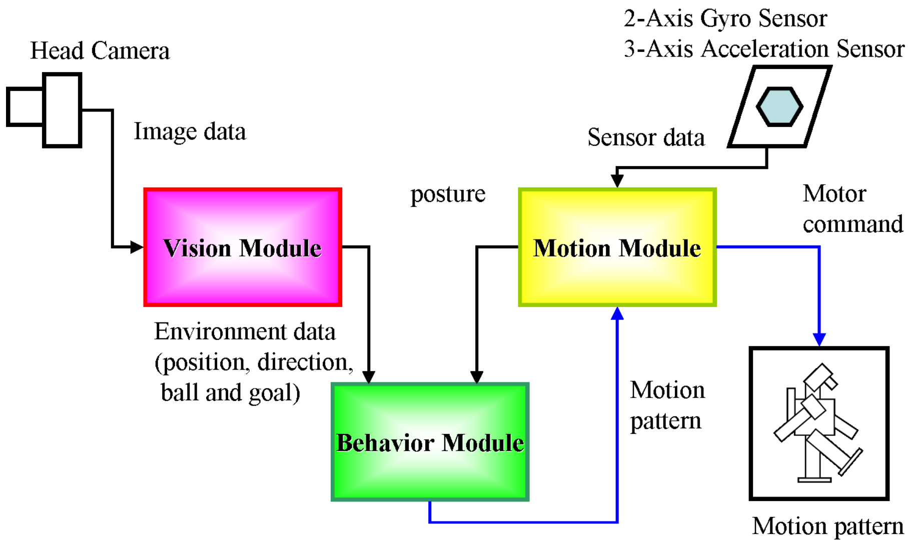
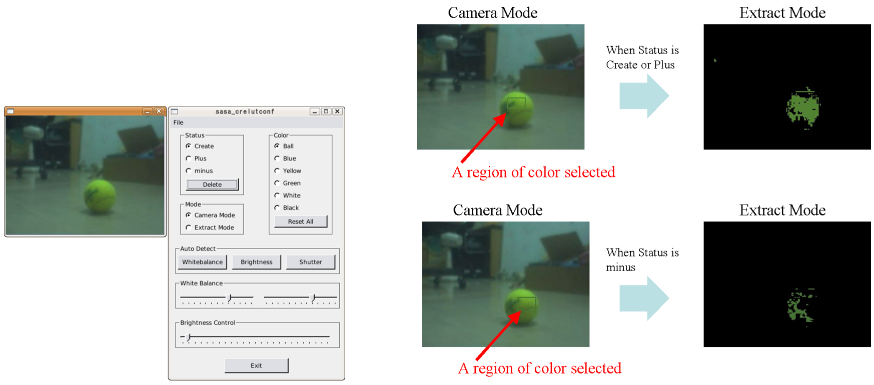


Fig. 2. Overall system of the software: There are three modules. A vision module detects positions and directions of the ball and goals from image data. A motion module receives the data from gyro and accelerator sensors and estimats the posture of the robot. Further, it gives motor commands to the robot as a function of the pattern of motion from behavior module. Behavior module detects the appropriate pattern of motion based on information from vision and motion modules.

we implemented a GUI application for setting color variation in vision module(see Fig. 3(a)). By using this system, we can make a color table by adding or subtracting the range of the values of pixels before the game(see Fig. 3(b)). The robot uses it during the game and detects the area of objects robustly.

3.2 Motion Module

The motion module analyzes the sensor data and gives the motor commands to the robot. VisiON 4G has the 2-axis gyro and 3-axis acceleration sensors. The posture of the robot, which is whether robot is down or not and if so which direction the robot is down, is detected by threshold processing of these sensor data. The motion module also receives the pattern of motion from behavior module, and transforms it into motor commands. In order to create these patterns of motion (e.g. kicking, standing up and so on), we use RoboviMaker, which is software developed by Vstone to handle small CPU board VS-RC003 on the robot(see Fig. 4). By using RobovieMaker, we can specify several postures by setting joint angles of the robot with slider bar, a motion is made by sequencing these postures.



(a) The interface

(b) One example of making color table process

Fig. 3. GUI application to set up the parameters of the image processing: In order to avoid spending a number of time to set up before the game.

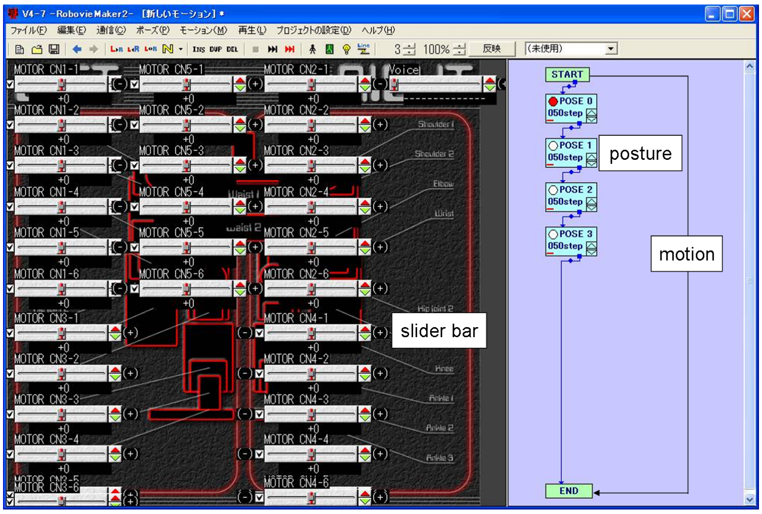


Fig. 4. GUI application to create patterns of motion: In order to create and test a lot of patterns of motion, we use RobovieMaker.

3.3 Behavior Module

The behavior module determines the appropriate pattern of motion, depending on the situation. We assign a role(goalkeeper, offence or defence) to one of three robots. The behavior module sequentially receives the information of the field(ball, goals and its own posture) from the vision and motion modules, and selects the appropriate motion from several motion patterns, such as kicking, approaching, standing up and so on.

4 Self-location

We are developing self-localization system for the robots by using field lines based on the Particle filter algorithm[1]. This filter is usually used to estimate Bayesian models and the sequential analogue of Markove chain Monte Carlo methods. In the system, robot's location are estimated in the following procedure:

1. Intersections of radially arranged lines in image and field lines on the field are detected(see Fig. 5).
2. We set a robot in global locations of the field, and make a lookup table of these intersections before the game.
3. Robot's location in the field is estimated by calculating the likelihood in the Particle filter algorithm from the lookup tabele.

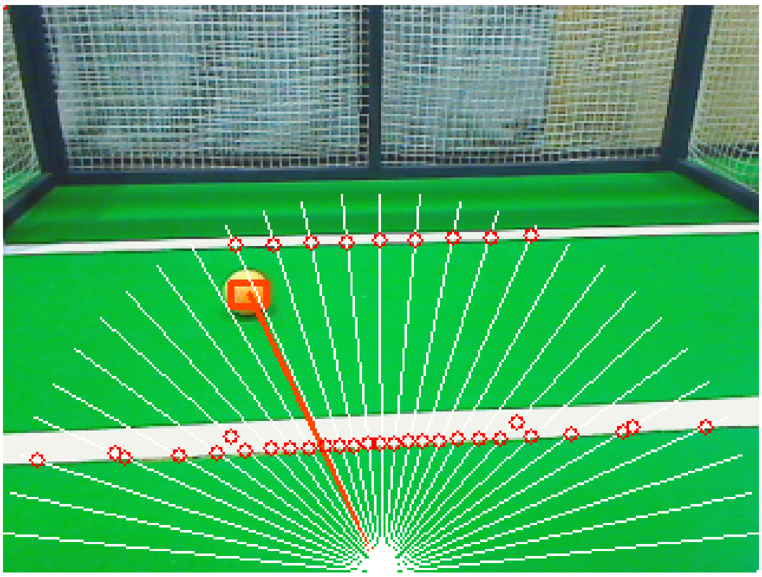


Fig. 5. Detect field line

References

1. D. Fox, S. Thrun, F. Dellaert, and W. Burgard. Particle filters for mobile robot localization. In A. Doucet, N. de Freitas, and N. Gordon, editors, *Sequential Monte Carlo Methods in Practice*. Springer Verlag, New York, 2000.
2. Norbert Michael Mayer, Joschka Boedecker, Masaki Ogino, Masaaki Kikuchi, Rodrigo da Silva Guerra, and Minoru Asada. Osaka University Team Senchans 2005. In *RoboCup 2005 Symposium papers and team description papers CD-ROM*, 2005.
3. Norbert Michael Mayer, Masaki Ogino, Rodrigo da Silva Guerra, Joschka Boedecker, Sawa Fuke, Hiroyuki Toyama, Ayako Watanabe, Kazuhiro Masui, and Minoru Asada. JEAP Team Description. In *RoboCup 2006 Symposium papers and team description papers CD-ROM*, 2006.