JollyPochie in the Four Legged Robot League

Ayumi Shinohara, Shigeru Takano, Satoru Miyamoto, Makoto Toyomasu

Department of Informatics, Kyushu University 33, Fukuoka 812-8581, Japan
{ ayumi, takano, s-miya, m-toyo } @i.kyushu-u.ac.jp
Phone number:+81-92-642-2697

May 9, 2003

Abstract

This paper presents our approach to the Sony Four Legged Robot League of Robocup 2003. The components of our system mainly consist of vision, localization, behaviors and walking module. We are also developing software tools that support interactive adjustment of walking motions, fixed motions, and color recognition.

1 Team Development

1.1 Team members

Professors
Ayumi Shinohara, Masayuki Takeda, Yoshihiro Okada, Shigeru Takano, Akira Ishino
Graduate Students
Makoto Toyomasu, Satoru Miyamoto, Satoshi Tsuruta, Takashi Funamoto,
Yoshikazu Numaguchi, Naoki Noguchi
Undergraduate Students
Yuichiro Matsubayashi, Jun Inoue

1.2 Team introduction

Our team "JollyPochie [dʒóli·pótʃi:]" consists of faculty staff and students of department of informatics, Kyushu University [1]. Our research interests mainly include machine learning, machine discovery, data mining, image processing, string processing, software architecture, visualization, and so on. To be honest, none of us had background knowledge on robotics so far. Some of us were inspired by Robocup2002 held in Fukuoka, in which our department locates, and hoped to challenge to a new research direction.

Basically, we are constructing every modules and tools from scratch by ourselves, both for educational purpose and for satisfaction of our curiosity. We have just started programming in March this year, so that the modules and tools are not mature at the present time. We hope to catch up with the world competitive level by participating the four legged league in this year 2003.

The rest of this paper briefly describes the current status of our development. We notice that it will be changed drastically since we keep improving the modules and tools rapidly.

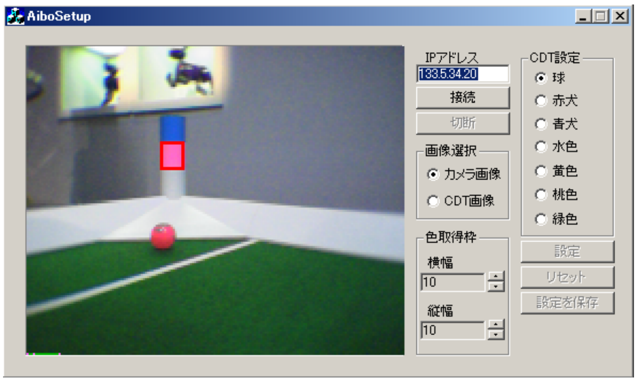

Figure 1: Screen Shot of Color Classification Tool

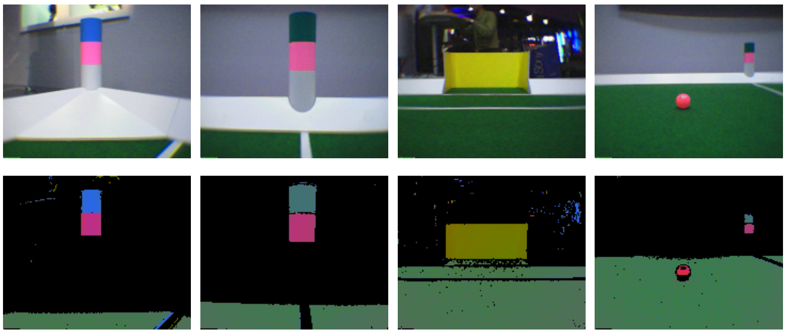

Figure 2: Example of Classified Images

2 Vision

Generally an autonomous robot does not have very fast CPU inside him/her, we thought that it is costly and possibly obstructs real-time operation to process large or high color images. Thus we adopted CDTs (Color Detection Table) which can handle up to eight colors at frame rate for identify objects on court, because the robot has the color detection facility in hardware. By using the CDTs of seven colors, we can detect the field objects such as goal, marker, enemy or friend robot and ball. Concretely, the seven colors in the court are orange for the ball, blue and yellow for the goals, pink, green, blue, yellow for the landmarks (blue and yellow are the same colors as goals), dark red and dark blue for the robots. The exact range of each color must be determined to classify these seven colors. Our color classification tool (Fig. 1) is very simple, but useful for adjusting the CDTs. Figure 2 illustrates the example of color classified images obtained by our developed tool.

Additionally, during the ball tracking, the distance from the robot to the ball is computed by using the infrared sensor. This information is useful to determine the next action of the robot.

3 Localization

At the moment, the localization task does not work as we expected. In order to determine the next action of a robot, it is necessary to know the location of the robot itself and which goal to attack or defend. Each robot is programmed to protect its own defense area. However since our localization module was not good enough, we could not give full play to its effect. It is extremely important for a goalkeeper robot to keep the defense area. Therefore, regarding the goalkeeper, we introduced only goal as the exact landmark and could obtain good localization information.

4 Behaviors

We have developed a quite powerful shooting action "PowerBomb Shoot", named after a similar action in professional wresting. After lifting up the boll by front legs, the robot flings it against the floor and simultaneously hits it by the front face. It is so powerful that even the goalkeeper can aim to shoot to the opponent goal directly if it has a chance. We carefully designed that the motion completes shorter than three seconds.

We have two attackers which also served as defenders. The basic behavior of attackers is, (1) find the ball, (2) approach the ball if it could find it, (3) turn around the ball toward the opponent goal, without touching the ball in order to prevent holding, and (4) perform the PowerBomb shoot. At every position, it try to shoot whenever it is possible. We locate each attacker as "Left Wing" and "Right Wing" area.

The defender would do clear kick instead of PowerBomb Shoot based on the situation. The goalkeeper does as follows. (1) try to find the ball, (2) traverse near to the ball, (3) clear the ball if it is close to goal.

5 Motion

It is important for the robot to search the ball, goals, and polls even in the process of moving forward, backward, or turning. It requires that the head can be moved independently from the action of legs. On the other hand, fixed motions such as kicking, shooting, and getting up require cooperative action between the head and legs. By this reason, we classified basic motions into two categories. One is for fixed motions which requires cooperative head movement, and the other is for walking motions which does not require it. Based on this idea, we have developed two software tools which support exploiting new motions and walking styles respectively, through the graphical user interface.

The movement is controlled by two modules Head and Walker. The Head modules controls the tilt, pan and roll of the neck for searching landmarks, while the Walker modules controls both neck and legs. When performing a fixed action, the Head falls into the idle state and the Walker modules controls everything. By this classification, we can simply specify the behaviors without confliction. It contributed our rapid development, although a subtle control was rather difficult.

5.1 Walking motion development tool

The walking motion development tool help us to adjusting parameters which affect walking motions interactively (Fig. 3). The parameters consist of the positions at which each leg reaches the floor and leave from it, the height of the head position, the height of the hip position, phase difference α between the left and right legs, phase difference ϕ between the front left leg and the right rear leg, the power ratio β of one stroke, the shape of the stroke, and the period of one stroke. From these parameters, we compute all angles of the joints in four legs by inverse kinematics. We can transmit the parameters to the robot via TCP connection, and observe its behavior interactively.

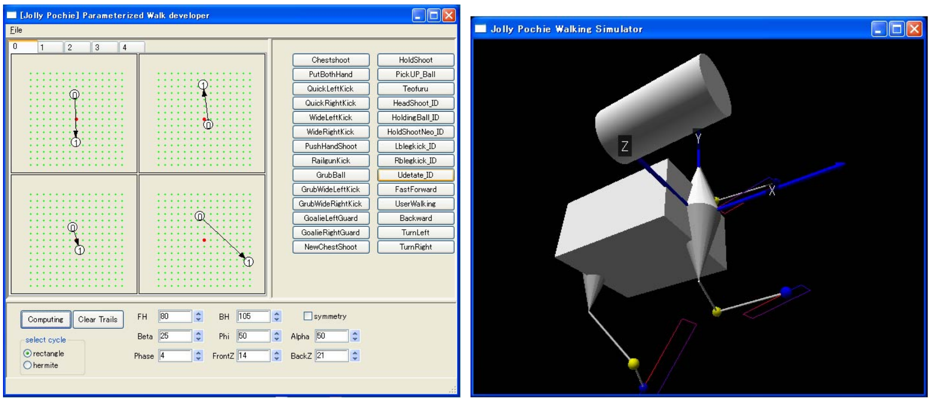

Figure 3: Walking motion developing tool

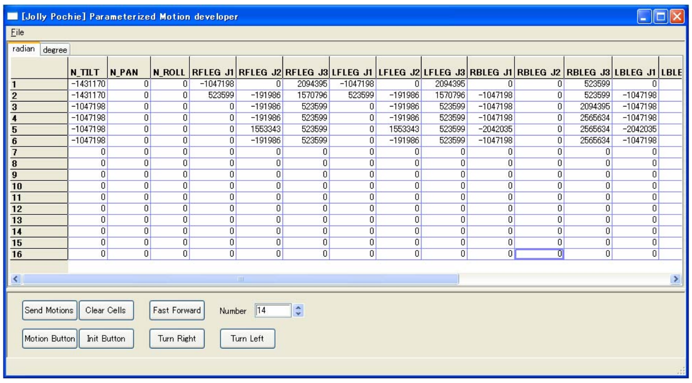

Figure 4: Fixed motion developing tool

5.2 Fixed motion developing tool

In our system, a fixed motion is specified by a sequence of 15 tuples of angles. Three angles are used for tilt, pan and role of the neck, the rests are for legs (three angles for each leg). We can edit every values both in degree and micro-radian by choosing one of two sheets (Fig. 4). We can transmit these values to the robot via TCP connection, and observe its behavior interactively. We developed the PowerBomb Shoot by using this tool.

6 Concluding remarks

We have explained the current state of our development very briefly. As other teams always do, we are continuing the development and improving the modules and tools day by day. Therefore it is quite natural that our programs will be changed drastically.

References

[1] JollyPochie Homepage <http://www.i.kyushu-u.ac.jp/JollyPochie/>