Jolly Pochie 2007 : Team Description Paper

Akira Ishino[1], Ayumi Shinohara[1], and Hayato Kobayashi[2]

[1] Graduate School of Information Science, Tohoku University
ishino@ecei.tohoku.ac.jp, ayumi@ecei.tohoku.ac.jp
[2] Department of Informatics, Kyushu University
h-koba@i.kyushu-u.ac.jp

1 Introduction

The team "Jolly Pochie [dʒóli·pótʃi:]" has participated in RoboCup Four-Legged League since 2003. In the first two years, the team consisted of the faculty staff and graduate/undergraduate students of Kyushu University, Department of Informatics. Since 2005, it has been a team with Tohoku University.

Faculty members
Ayumi Shinohara, Akira Ishino
Student members
Hayato Kobayashi, Tsugutoyo Osaki, Tetsuro Okuyama, Shuhei Yanagimachi, Masayoshi Sugimoto, Keisuke Oi, Wataru Matsubara, Takahito Sasaki, Tomoyuki Nakamura, Seiji Hirama, Gerardo Snyder, Naoyuki Terazono

Our research interests include machine learning, machine discovery, data mining, image processing, string processing, software architecture, visualization, and so on. We think RoboCup is a suitable benchmark problem for these domains.

Last year, we ended second place in the RoboCup Japan Open 2006. The first game in RoboCup 2006 successfully ended with a score of 3:1, however, we lost the following two games. Our achievements last year include second place in the Open challenge and the first day's winner of the exhibition race by legged robots.

This paper presents our current development status focusing mainly focused on the progress since last year. Section 2 describes our new fuzzy logic based decision engine used to implement soccer strategies. Section 3 describes the reconstruction of our image processing method. Section 4 introduces auxiliary landmarks which are used for the automatic adjustment of parameters. We outline our new learning method in section 5. Section 6 illustrates our development environment base on the web application. Finally, section 7 concludes this paper.

2 Decision Engine

Last year we wrote programs for strategy in Lua, the scripting language. This year, we utilize fuzzy logic in an inference engine for writing strategic programs. The fuzzy logic does not deal only with true and false but it also deals with the numerical values between 1 and 0, where 1 represents true and 0 false, respectively. Since the operands

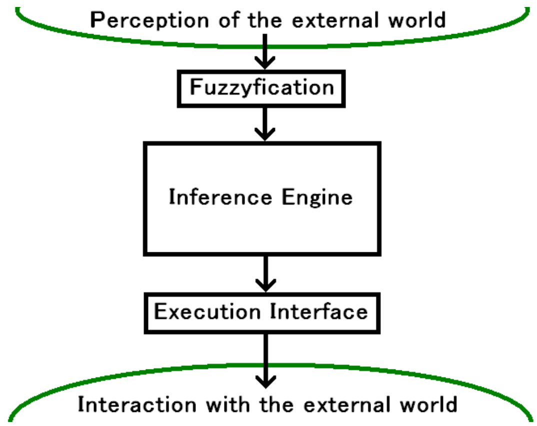

Fig. 1. The decision engine concept.

are numerical, they are more flexible to use and are appropriate for rule based system. The rule based system is a good way to maintain a stable AI and to implement new behaviors since they are processed independently. The whole system is a platform which can be used for all programs which need a decision process between numerical entries and a set of actions.

Robots are particularly appropriate to use a decision process because they have to choose a motion according to their environment. We have implemented the decision engine in agent players. The rules for the decision engine are quite simple and rapid to construct compared to our previous if-then-rule-based system. Moreover, each player can dynamically select the appropriate set of rules during a game depending on its state, role and the other players.

The decision engine is written in a C++ program with a bounded Lua interface. It provides a debugging tool, good processing into performance and the simplicity of a scripting language. The scripts are now separated four parts: parameters, rules, by actions and states. In this way, each developer can create and improve behaviors solely.

3 Image Processing

Last year, we reconstructed our image processing system. Unfortunately, it contained bugs when used in dark rooms, thus, we lost some games. This year, we fixed those bugs, and improved the system for more efficiency. The system refers to the techniques ‘soft color’ and ‘blob formation’ described in [1]

The average time for image processing was 11.28 ms last year. Our robots calculated connected components for colors, e.g. yellow, orange, cyan, and so on, separately. This year they calculate connected components for all colors at the same time, and the average time for image processing is now 4.79 ms. Last year, our robots spent most of the time for image processing but that is not the case this year.

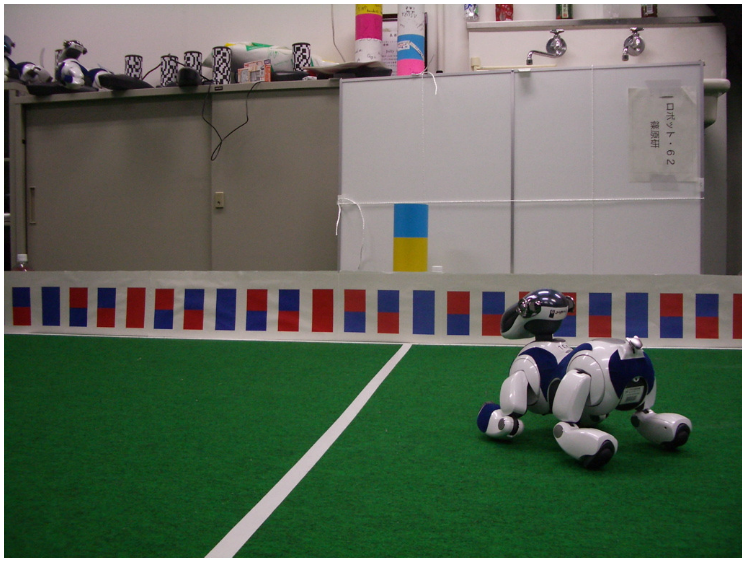

Fig. 2. The Auxiliary Landmarks

Because the number of poles decreases to two this year, the ability to detect white lines is more important. Even though we have already implemented white line detecting method, we are improving upon it.

4 Automatic Adjustment by Auxiliary Landmarks

If there exists a method of calculating the position of robots correctly, robots can automatically adjust parameters for locomotion, color table, localization, and so on. In our laboratory cameras are set up on the ceiling, to measure the positions of robots and balls, and apply that information to the machine learning method for locomotion. However, this equipment cannot be used at the competition.

This year we plan to use auxiliary landmarks (Fig. 2) for automatic parameter adjustment. The auxiliary landmarks are sequences of four kinds of markers, which consist of red and blue squares that surround the soccer field.

Robots can estimate their positions more accurately by using auxiliary landmarks as opposed to using only standard landmarks, i.e. two goals and two poles. Moreover, estimating to positions by using auxiliary landmarks can be done in real time as opposed to using ceiling camera. Because the ceiling camera system communicates with the robots via wireless network, and there is a time lag between the robots and the PC.

We can carry the landmarks, which are made from paper, to the competition easily. We hope to adjusting parameters can be done automatically.

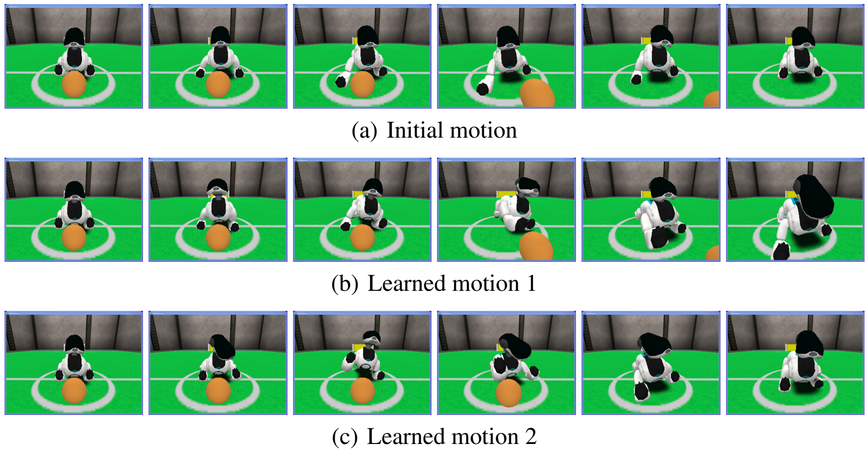

Fig. 3. Initial motion and learned motions.

5 Shot Learning Using Thinning-out

Shooting is one of the most important skills in soccer games, as it directly affects scoring points. In the four-legged robot league, we must develop accurate and strong shot motions only by changing the parameters of the motions, since the hardware is limited to AIBO. This task consumes much of developers' time and energy and must be adjusted whenever the environment (e.g., friction of the field) is changed. By using machine learning, we can efficiently reduce these burdens. Existing learning methods, however, take much evaluation time for each trial in the motion learning.

In this year, we proposed a new concept, thinning-out, for reducing the number of trials in the motion learning. Thinning-out means to skip over such trials that are unlikely to improve motions, in the same way that gardeners thin-out weak seedlings. Our thinning-out technique significantly reduced the number of trials. In addition, our robot was able to acquire a sophisticated motion that is much different from the initial motion as the result of utilizing a hybrid learning method combining meta-heuristics and the thinning-out technique. As shown Fig. 3, our robot learned (b) the motion that the robot uses its whole body and (c) the motion that the robot utilizes its own weight, on the AIBO 3D simulator which Zaratti et al. developed [2].

6 Tools

We have developed many tools by using Python/GTK and Processing. However, we had to prepare a lot of tools for using our tools, and it is not always easy for new members. Then, we created our develop environment as a Web application.

Our develop environment on the Web is based on ‘Ruby on Rails’. In the environment we can create and edit motions, locomotions, and Lua scripts. Moreover, we can monitor the robots' onboard camera images, object detecting statuses, and localization

(a) The motion editor (b) The camera monitor

Fig. 4. The motion editor.

statuses. Because it is a Web application, one server can handle any number of clients, i.e. web browers, which are easy to use for new members.

7 Conclusion

This paper described the progress of Jolly Pochie in this year. We developed the decision engine based on fuzzy logic, which enables us to write programs for soccer strategy simply and descriptively, we reconstructed image processing to work more effectively, and created our web-based developing environment. Lastly, we developed auxiliary landmarks for automatic adjustment. We hope to present these techniques at this RoboCup 2007.

References

1. Naomi Henderson. Digital image processing in robot vision. Technical report, 2005.
2. Marco Zaratti, Marco Fratarcangeli, and Luca Iocchi. A 3D Simulator of Multiple Legged Robots based on USARSim. In *RoboCup 2006: Robot Soccer World Cup X*, LNAI. Springer-Verlag, 2007. to appear.