Team Description of SOCIO 2008

Nobuhiro Ito*2, Kenji Ozoe*1, Noriaki Shimizu*2, Takeshi Uchitane*3, Toshiharu Hatanaka*3, Youichi Akabane*3, Tomoharu Nakashima*4, Yukio Shoji*4, and Seiya Fujii*4

*1Department of Information engineering,
Nagoya Institute of Technology
Gokiso-cho, Showa-ku, Nagoya City,
Aichi Prefecture 466-8555, JAPAN
agent-staff@phaser.elcom.nitech.ac.jp

*2Software Science Course,
Department of Applied Information Science,
Faculty of Management and Information Science,
Aichi Institute of Technology
1247 Yachigusa, Yakusa-Cho, Toyota City,
Aichi Prefecture 470-0392, JAPAN
n-ito@aitech.ac.jp

*3Department of Information and Physical Sciences
Osaka University, JAPAN
Yamadaoka 2-1, Suita, Osaka 565-0871, JAPAN
t-uchitane@ap.eng.osaka-u.ac.jp, hatanaka@ist.osaka-u.ac.jp,
y-akabane@ap.eng.osaka-u.ac.jp

*4Department of Computer Science and Intelligent Systems
Osaka Prefecture University
Gakuen-cho 1-1, Naka-ku, Sakai, Osaka 599-8531, JAPAN
nakashi@cs.osakafu-u.ac.jp, shoji@ci.cs.osakafu-u.ac.jp, fujii@ci.cs.osakafu-u.ac.jp

Abstract. This paper is the description of team SOCIO, which applied for the participation in RoboCup 2008 Mixed Reality league. First, we introduce the team from various aspects such as the composition of the team. Next, we describe a simulator of the mixed reality agents that will be our contribution to the development competition. Finally, some methods for implementing robot controllers for the mixed reality league are briefly shown.

1 Introduction

Mixed reality league is a new league that officially started at RoboCup 2007. The main goal of the league is to bridge the gap between real robots and simulated robots. Thus, the league consists of both real and simulated objects: Real robots are used for agents while their working field is realized in a simulated environment (see Fig. 1).

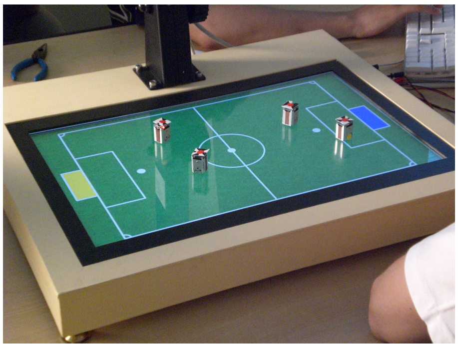

Fig. 1. Snapshot of Mixed Reality League.

Team SOCIO, which is the abbreviation for SOccer Computational Intelligence Organization, is a mixed team by Osaka University, Aichi Institute of Technology, Nagoya Institute of Technology, and Osaka Prefecture University. From Osaka University one faculty and two graduate student members join the team. They have participated in RoboCup soccer simulation 3D league since 2006. They participated in RoboCup 2006 in Bremen and 2007 in Atlanta. In Japan Open 2006, they won the second place and the first place in Japan Open 2007. The group from Aichi Institute of Technology and Nagoya Institute of Technology consist of one faculty and one graduate student members. They have participated in RoboCup rescue simulation league from the very beginning of the event (i.e., 1997). They have won in numerous RoboCup events (in both domestic and international RoboCup events). They also participate in RoboCup soccer simulation 3D league. The group won the first place in Japan Open 2006 and the third place in Japan Open 2007. The group from Osaka Prefecture University has participated in RoboCup soccer simulation 2D league since 2002. The member of the group consist of one faculty and two graduate student members. They won the second place in Japan Open 2007 and the fourth place in RoboCup 2007. Each participating organization in SOCIO has its expert knowledge such as control systems, multi-agent systems, and learning systems. Also they all have experience in participating other RoboCup competitions. By forming the team, we can work together in a mutually complementary manner.

We participated in the first mixed reality league in RoboCup 2007, which was held in Atlanta, U.S.A. In RoboCup 2007, we could not participate in two of the three competitions, i.e., the platform development competition and the new application competition while we participated in the soccer tournament competition. In the soccer tournament competition, the team won the third place. However, it seems that there is not any big difference among the participating teams as all teams have only a little time to develop their soccer agents.

In this paper, we describe our plan towards the participation in RoboCup 2008. Specifically the new application competition and the soccer tournament competition are focused on this year. First, our plan for the new application competition is described. Next, we show how to implement soccer agents by using computational intelligence methods. Finally future work of the team is discussed.

2 New Application: Mixed Reality Simulator

2.1 Current Problem in Mixed Reality

In the current setup of the Mixed Reality (MR) league, the following things make experiments with robots difficult:

- Expensive equipment requirement such as a camera, robots and a battery charger,
- Limited number of experiments due to the short duration of rechargeable battery, and
- Difficulty in implementing robot movement with insufficient experience of the league.

In order to solve the above problems, we develop a simulator of MR robots called MR simulator. In the MR simulator, a command set in the 2D soccer simulator is used. By using the MR simulator, we can perform the development of MR robot movement without using the expensive equipment. The MR simulator is based on the command set of the 2D soccer simulator as a large number of implemented methods for the 2D soccer agents can be easily and effectively transferred to MR robots. Also the development of the MR simulator is expected to be paid much attention by participants of the 2D league. In the following subsections, we describe the details of the MR simulator.

2.2 Current development procedure

Before we describe MR simulator, let us summarize the current situation of the robot development. The current development procedure of MR robot movement can be described as follows:

1. Implement the movement of MR robots by some programming language
2. Experimental test of the movement using MR robots according to the following procedure:
   2.1 Calibrate camera images of the MR system
   2.2 Start the command server (soccer server in the case of soccer tournament)
   2.3 Connect MR robot program to the command server
   2.4 Start experiment
3. Recharge the batteries of MR robots

As can be seen from the above procedure, it is difficult to iterate the experiment procedure many times because both the calibration of camera images and the recharge of batteries are necessary in each experiment. Therefore we suggest another development way using the MR simulator. We describe the details of the MR simulator in the following sections.

2.3 New development procedure

We show below the new development procedure that we suggest.

1. Implement the movement of MR robots by some programming language
2. Start the command server
3. Connect MR robot program to the command server
4. Modify the implemented MR robot program based on the experimental result
5. Repeat Step 2 to 4 until the experimental results become sufficiently good
6. Perform the final validation test using the MR system

In the above procedure the real MR system is used only in Step 6. Therefore, we can perform operation check easily and frequently. In addition, the MR simulator uses the command set in the 2D soccer simulator, which makes it easy to use the existing methods that has been already studied in the 2D soccer league.

2.4 Architecture

We show the 2D architecture in Fig. 2, and we also show the MR architecture in Fig. 3. In the 2D architecture in Fig. 2, the soccer server and soccer agents communicate with each other. On the other hand, in the MR architecture commands from MR robots are first sent to the server. Then the server converts the commands from the MR controllers into signals to MR robots.

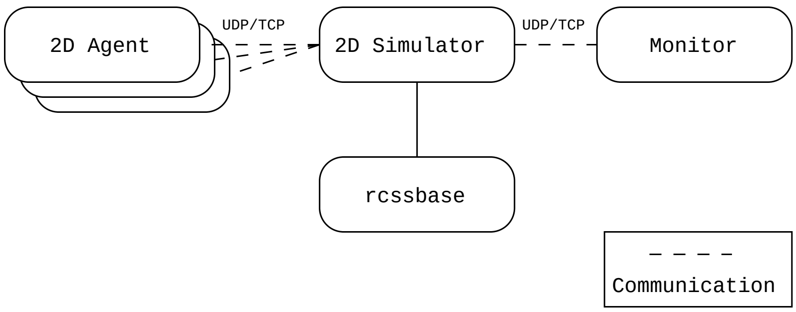

Fig. 2. 2D Architecture

Next, we show the new architecture we propose in Fig. 4. Let us consider how the architecture in Fig. 4 is developed. At first, we have to develop an MR simulator to test the movement of MR controller that is implemented with the command set of 2D soccer server. The MR simulator can be a replacement for the 2D simulator in Fig. 2. That is, the MR simulator should be able to connect to 2D agents and to communicate with them as in the 2D architecture. In addition, it is necessary to develop a transfer module that converts the commands from 2D agents to the commands for MR robots. We call this module a translator. The

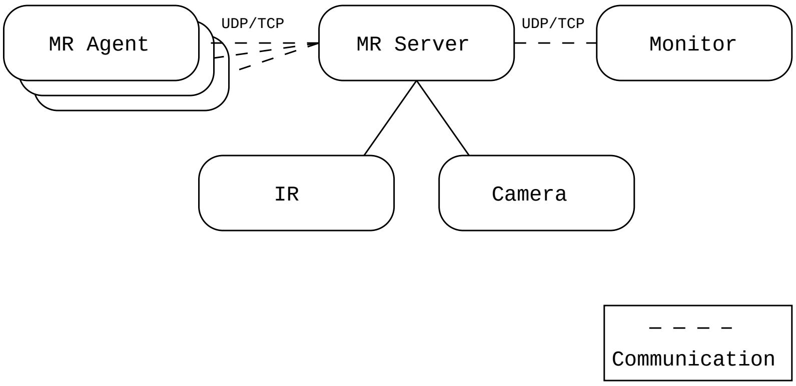

Fig. 3. MR Architecture

translator is used in the phase of final validation test in Step 6. The translator also has to be able to connect to the MR server.

In summary, we develop the following two modules:

- MR simulator that is based on command form same as 2D soccer simulator
- Translator called conversion server which connects to the MR server using the command set in the 2D simulator

The two modules (i.e., MR simulator and translator) are shown in Fig. 4 (Dotted boxes).

Since the command sets in the MR league and the 2D league are not exactly the same, it is necessary to specify the relationship between 2D commands and MR commands. We specified the relation by hand as shown in Table 1. Here, "–" shows that there is no counterpart in the MR command set. In addition, We do not use 2D commands that is not in the MR command set.

Table 1. Command

Command	2D command	MR command
Catch a ball(keeper)	catch	–
Acceleration of an agent to the front or back	dash	mr_set_velocity(x,x)
Acceleration of a ball to the given direction	kick	mr_kick
Turns in the direction of the body	turn	mr_set_velocity(+x,-x)
Move to the specified point (Feasible only after goals or before kickoff)	move	–
Acceleration a ball to the direction of the body	tackle	–

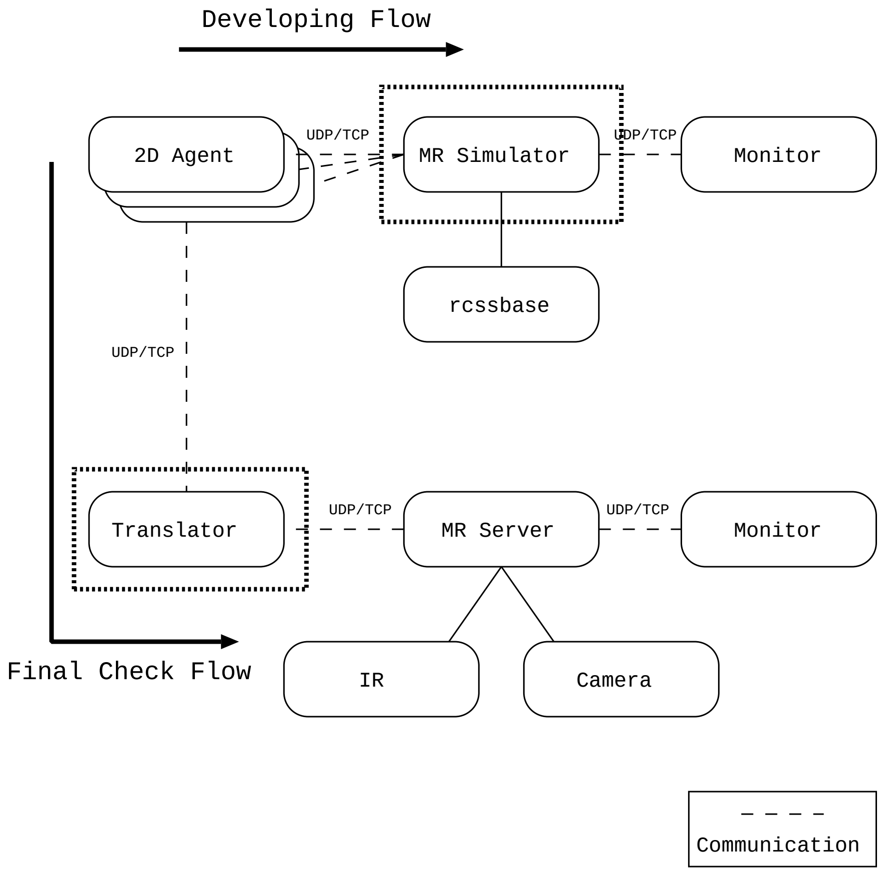

Fig. 4. NEW Architecture

3 Soccer Tournament: Self-Localization for MR Robots using Particle Filter

We deal with so-called tracking problem for self-localization of MR robots. The server system calculates the position of MR robots by observing the camera images, and they include not only the information of the real position of MR robots, but also some kinds of noises caused by vibration, uneven light volume, computational errors and so on. To estimate more accurate position of MR robots, we applied particle filter [1] [2]. With particle filter, we can estimate more noiseless position of MR robots in the situation that the information includes noises and the targets are multiple and interactive.

3.1 Particle filter for Tracking problems

The position of target is calculated by the system equation (1) and the observation equation (2). The system equation is defined according to the dynamics of the target and the system noises, and the observation equation is defined by the way of observing the position of target and the observation noise.

$$X_t = F_t[X_{t-1}, V_t] \tag{1}$$
$$Y_t = G_t[X_t, W_t] \tag{2}$$

where X_t is the position of target at time t. F_t is the function according to the dynamics of the target at time t. V_t is the system noises. Y_t is the observed position. G_t is the function according to the way of observing the position. W_t is the observation noise at time t.

By the system equation, a lot of candidates of positions X_t are calculated from the previous estimated positions X_{t-1}. The candidates X_t have the likelihood according to the probability distribution $P(Y_t|X_t)$, then the position at time t is estimated by the likelihood. In this way, particle filter estimates more accurate position in each time t.

3.2 Particle filter for MR robots

In the system equation (3), MR robots movements function F_t is formed by the two wheel velocities L_t and R_t depending on a set of received commands (BF, BM, MS, ST, FS, FM, FF) at time t. The two wheel velocities are not constant even if the same commands are received, because MR robots have some noises caused by their friction, vibration and individual difference. So, they are defined as: $L_t = vl_t + el_t$ and $R_t = vr_t + er_t$. Here, let vl_t and vr_t denote the average velocities, and el_t and er_t denote Gaussian noise $N(0, \sigma)$. To calculate the average and standard deviation of velocities, we measured the MR robot's velocities many times with commands "FSFS", "FMFM" and "FFFF".
In the observation equation (4), the observation noise W_t is the Gaussian noise caused by uneven light volume and computational errors. To calculate the variance of noise W_t, we measured the fluctuations of the observed positions of stopping MR robots many times.

3.3 System Equation for MR robots

$$X_t = F_t[X_{t-1}, vl_t, vr_t, el_t, er_t] \tag{3}$$

where $X_t = (x_{1t}, x_{2t}, \phi_t)^T$ is the position and direction of MR robots in the real world at time t. F_t is the movements function of MR robots. vl_t and vr_t are the average velocities depending on commands. el_t and er_t are the system noises.

3.4 Observation Equation for MR robots

$$Y_t = G_t[X_t, W_t] \quad (4)$$

where $Y_t = (y_{1t}, y_{2t}, \phi_t)^T$ is the position and direction of MR robots in the virtual world at time t. G_t is the function defined by camera-feedback programs. W_t is the observation noise.

References

1. N. M. Kwok and G. Dissanayake, "An Efficient Multiple Hypothesis Filter for Bearing-Only SLAM," Proceedings of 2004 IEEE/RSJ International Conference on Intelligent Robots and Systems, September 28 - October 2, 2004.
2. Zia Khan, Tucker Balch, and Frank Dellaert, "MCMC-Based Particle Filtering for Tracking a Variable Number of Interacting Targets," IEEE Transaction on Pattern Analysis and Machine Intelligence, Vol.27, No.11, November 2005.