Design of an Intelligent Interface for the Software Mobile Agents using Augmented Reality

Kazuto Kurane[1], Munehiro Takimoto[2] and Yasushi Kambayashi[3]
[1]*Graduate School of Computer and Information Engineering, Nippon Institute of Technology, 4-1 Gakuendai, Miyashiro-machi, Minamisaitama-gun, Saitama, Japan*
[2]*Department of Information Sciences, Tokyo University of Science, 2641 Yamazaki, Noda, Chiba, Japan*
[3]*Department of Computer and Information Engineering, Nippon Institute of Technology, 4-1 Gakuendai, Miyashiro-machi, Minamisaitama-gun, Saitama, Japan*

Keywords: Agent, Mobile Agent, Augmented Reality, Kinect, Gesture.

Abstract: In this paper we propose an intelligent interface for the mobile software agent systems that we have developed. The interface has two roles. One is to visualize the mobile software agents using augmented reality (AR) and the other is to give human users a means to control the mobile software agents by gesture using a motion capture camera. Through the interface we human beings can intuitively grasp the activities of the mobile agents, i.e. through augmented reality. In order to provide proactive inputs from user, we utilize a Kinect motion capture camera to recognize the human users' will. The Kinect motion capture camera is mounted under the ceiling of the room, and they monitor the user as well as robots. When the user points at a certain AR image, and then points at another robot through gesture, the monitoring software recognizes the will of the user and conveys instructions to the mobile agent based on the information from the Kinect. The mobile agent that is represented by the image moves to the robot that was pointed. This paper reports the development of the intelligent user interface described above.

1 INTRODUCTION

In the last decade, robot systems have made rapid progress not only in their behaviors but also in the way they are controlled. We have demonstrated that a control system based on multiple software agents can control robots efficiently (Kambayashi et al., 2009). We have constructed various mobile multi-robot systems using mobile software agents (Abe et al., 2011).

In the constructions, the mobile agents are bulit as autonomous entities. Human users have no roles to control the mobile software agents. However, in an application of multiple robots searching for an object, and if the user knows the location of the object, the user should be able to tell the mobile software agent that control the mobile robots where to migrate, then the controlling software agent can move to the most conveniently located mobile robot, thus can capture the object without unnecessary movements. In this paper, we propose an intelligent interface that has two roles. One is to visualize the mobile software agents using augmented reality (AR) and the other is to give human users a means to control the mobile software agents by gestures using a motion capture camera. We believe that our intelligent interface system opens a new horizon of the interaction between human users and autonomous mobile software agents and robots.

The structure of the balance of this paper is as follows. In the second section we describe the background. The third section describes the design of the system, and the fourth section describes how the design is implemented. We conclude the discussion and present the future directions in the fifth section.

2 BACKGROUND

It is common to observe that sensors and robots are connected in networks today. We have studied hierarchical mobile agent systems in order to control them systematically. In particular, we have pursued a development of the control system for autonomous robots using mobile software agents (Abe et al.,

Kurane K., Takimoto M. and Kambayashi Y..
Design of an Intelligent Interface for the Software Mobile Agents using Augmented Reality.
DOI: 10.5220/0004331304390442
In *Proceedings of the 5th International Conference on Agents and Artificial Intelligence* (ICAART-2013), pages 439-442
ISBN: 978-989-8565-38-9

2011); (Kambayashi et al., 2009). On the other hand, we have not investigated any user interface for mobile agents, even though there are quite a few research and developments of visualizing software entities using AR (Azuma, 1997); (Feiner et al., 1993); (Tomlinson et al., 2005). We have not provided any means for users to intervene the behaviours of the mobile software agents after their departure.

If there is an interface that let a user can control software agents intuitively, the user can communicate with agents and thus with robots. The interface will widen the range of the applications by using such techniques. Therefore, we are developing a visual representation of the mobile software agent by using AR. The user will be able to intuitively grasp the presence of the mobile software agent by their AR representations. In addition to the representation, we are developing an intuitive means of control them by gesture using Xbox Kinect.

Xbox Kinect was developed by Microsoft to provide intuitive interface in game machines manufactured by the same company (Microsoft, 2012). The development of Kinect SDK opened the various human computer interface researches using Kinect. Kinect can be employed in the human interface of the robot systems with the SDK. We have developed an interface that recognizes users' gestures using the Kinect. The interface also directs the mobile agents to migrate over the robots when the user points at. If a user can communicate with software agents as well as mobile robots by an intuitive method the width of the applications over mobile robots should widen.

We have developed the mobile agent environment using Agentspace. Agentspace is a framework to build mobile agent environment and is developed by Ichiro Sato (Satoh, 1999). In the environment, a mobile software agent is defined as a set of callback methods in Agentspace.

In our development, we utilize AR using ARToolKit. ARToolKit is a library for realizing augmented reality developed by Hirokazu Kato (Kato, 2002). The library makes the implementation of the real time AR relatively easily. Since each robot has a unique marker, when the web camera finds the marker in the image, ARToolKit calculates the position and orientation of the marker. Then we can project a virtual object on the basis of information on the position and posture. An agent's existence is represented as a virtual object on the marker. When the agent moves from a robot to another robot, ARToolKit erases the virtual object on the image of the source robot and put the same virtual object on the image of the destination robot.

The interface provides visual representations of the mobile software agents, and depicts how they move between the robots in the manner described above. Figure 1 shows a virtual object that represents a software agent that sits on the marker. Since ARToolKit can acquire the three-dimensional coordinates of the marker seen from the camera, the system determines which robot the user points using three-dimensional coordinates of the joints of the human user and the three-dimensional coordinates of the markers.

Figure 1: A virtual object on a robot.

3 SYSTEM DESCRIPTION

Figure 2 shows the schematic diagram of the system. The system is assumed to be used indoors. The system consists of a host PC, Kinect, multiple robots with PCs. Host PC and PCs on the robots are connected by the Internet.

In the system, there are two types of agents. They are robot agents, which migrate to robots to drive them, and a host agent. Robot agents can reside in any PC on the robots that are connected by the networks. A robot agent controls the residing robot while it is moving. The host agent is an agent and resides in the host computer, and occasionally visits PCs on the robots to give the moving instructions to robot agents.

When the user performs a gesture, the host agent conveys the instructions for the robot agent by migrating to the robot on which the agent resides. When the robot with which a robot agent exists is captured by the camera of Kinect, ARToolKit projects the virtual object which expresses a robot agent on the robot in the screen. Kinect is connected to host PC. Kinect is installed under the ceiling of the room and monitors the user and the robots.

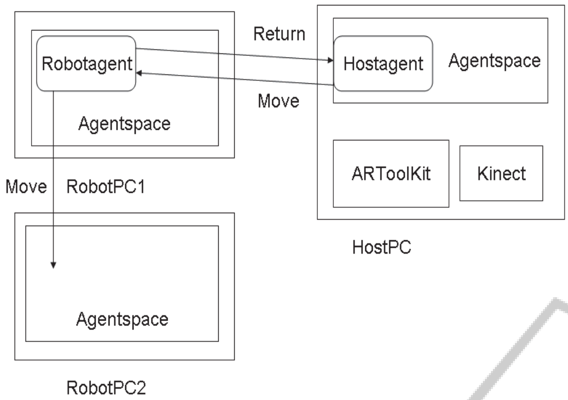

Figure 2: Schematic diagram of the system.

ARToolKit expresses a robot agent's movement in the following manner. Each robot has a unique marker on it. ARToolKit projects a virtual object expressing a robot agent on the marker on the robot on which the robot agent resides. When a robot agent migrates from a robot to another robot, ARToolKit changes the marker on which the virtual object is projected from the marker on the source robot to the marker on the destination robot. Thus the virtual object representing the robot agent moves from the source robot to the destination robot.

When the user performs a certain gesture, Kinect recognizes the gesture and determines which robot the user points by the following methods. The user raises the left hand while pointing at the robot on which the desired agent resides and then points at the robot to which the user wants the agent moves. The user declares that he is performing gesture to Kinect by raising his left hand. Kinect acquires the three-dimensional coordinate of the right hand, right shoulder, and right knee by the captured image from the camera. Kinect makes a straight line connecting the point of the right shoulder and the right hand of the user. The straight line is extended up to the height of the user's knee. Kinect gets the point that the field in the knees and straight line crossed. That point is where the user is pointed. The system acquires the three-dimensional coordinate of each marker by the camera and projects on the screen using ARToolKit. The system can measure the distance between the coordinates that the user points and the coordinate of each marker. The robot agent moves to the robot with that marker.

4 IMPLEMENTATION

The system we are developing needs to use two or more markers. However, ARToolKit can read only one kind of marker. Therefore we extended ARToolKit so that it can read several markers. Furthermore, we have created a special file that assists the management of the projection of virtual objects by the extended ARToolKit. The information which marker the extended ARToolKit should use to project a virtual object on the screen image is written in this file. The extended ARToolKit decides which object it should project on a particular marker by reading this file. The host agent can switch the markers on which the virtual object be displayed by rewriting this file. Thus the host agent can manage the projection of the virtual objects that represent the robot agents.

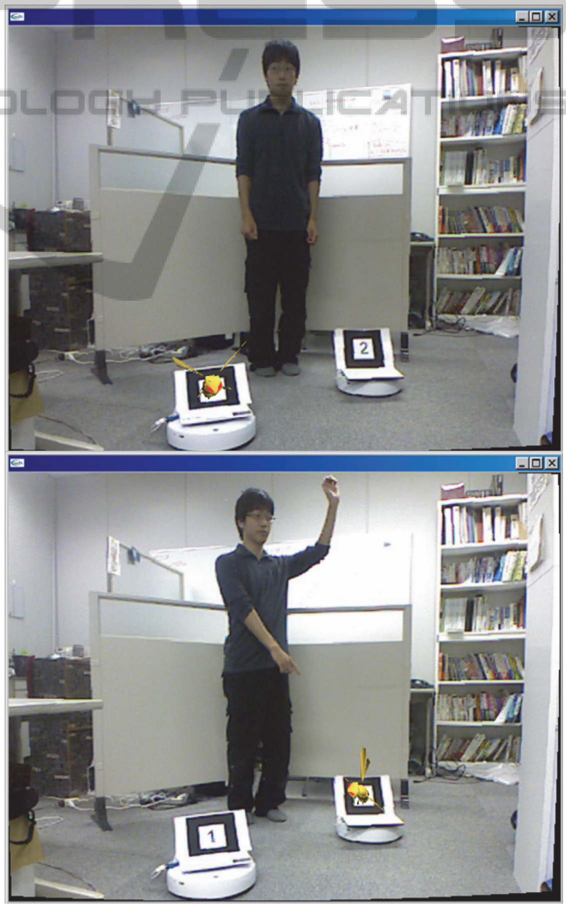

Figure 3: A virtual object moves from a robot to a robot along with a mobile agent.

The followings are how the system moves the virtual objects in the cyber space as well as the robot agents in the physical space.

1. After having pointed at the robot on which the user wants the robot agent to move by the right hand, the user raises the left hand.
2. Kinect recognizes the user has performed a gesture.
3. Kinect recognizes which robot the user is pointing at by using the coordinates of the markers and the coordinates of the joints of the user.
4. Kinect sends the information about the pointed robot to the host agent.
5. The host agent rewrites the file which has the managing information of the display of the virtual objects in the extended ARToolKit.
6. The extended ARToolKit projects the virtual object which represents the mobile agent on the robot which the robot agent is supposed to move to.
7. The host agent moves to the PC on which the robot agent resides.
8. The host agent passes the information to the robot agent to which robot it should move.
9. The robot agent moves to the robot that the user has pointed.
10. The host agent is returned to the host PC.

By the operations described above, the system makes it possible to visually represent the software agents that are moving between the robots and the user directs the agents by gestures. Figure 3 shows a user performing the gestures to direct a software agent.

5 CONCLUSIONS AND FUTURE DIRECTIONS

We have proposed an intelligent interface for the mobile software agent systems that we have developed. Through the interface, the users can intuitively grasp the activities of the mobile agents. In order to provide proactive inputs from user, we have utilized the Kinect motion capture cameras to capture the users' will expressed by the gestures. Since the Kinect motion capture camera is mounted under the ceiling of the room, we currently have the following problems.

1. This system is confined indoor. Kinect must be placed on a place where it can monitor the robots as well as the user. Therefore the system cannot be used outdoors.
2. ARToolKit cannot recognize markers when a robot leaves the scope of the Kinect. If Kinect is about 3.0m or more away from a marker, ARToolKit cannot detect the marker.
3. Kinect sometimes makes mistakes about the gestures. Kinect recognizes that the user is performing a gesture when a user raises the left hand. Even if the user does not intend to do a gesture, Kinect may misidentify that the user has done a gesture.

In order to mitigate these problems, we are extending the ARToolKit so that Kinect can cognizes the users' gestures more precisely as well as looking for some other medium to capture the scene in the open field.

REFERENCES

Abe, T., Takimoto, M. and Kambayashi, Y. (2011). Searching Targets Using Mobile Agents in a Large Scale Multi-robot Environment, *Proceedings of the Fifth KES International Conference on Agent and Multi-Agent Systems: Technologies and Applications*, LNAI 6682, 221-220.

Azuma, R. (1997). Survey of Augmented Reality, *Presence: Teleoperators and Virtual Environments*, vol.6, no.4, 355-385.

Feiner, S., Maclintyre, B. and Seligmann, D. D. (1993). Knowledge-Based Augmented Reality, *Communications of the ACM*, vol. 36, no.7, 52-62.

Kambayashi, Y., Ugajin, M., Sato, O., Tsujimura, Y., Yamachi, H. and Takimoto, M. (2009). Integrating Ant Colony Clustering Method to Multi-Robots Using Mobile Agents, *Industrial Engineering and Management System*, vol.8, no.3, 181-193.

Kato, H. (2002). ARToolKit: Library for Vision-based Augmented Reality. *IEIC Technical Report*,vol.101, no.652, 79 - 86.

Microsoft Kinect Homepage (2012). http://research.microsoft.com/en-us/um/redmond/projects/kinectsdk/default.aspx.

Satoh, I. (1999). A Mobile Agent-Based Framework for Active Networks, *Proceedings of IEEE System Man and Cybernetics Conference*, 71-76.

Tomlinson, B., Yan, M. L., O' Connell, J., Williams, K., Yamaoka, S. (2005). The Virtual Raft Project: A Mobile Interface for Interacting with Communities of Autonomous Characters, *Proceedings of ACM Conference on Human Factors in Computing Systems*, 1148-1149.