Journal of Information Processing Vol.21 No.2 304–314 (Apr. 2013)

[DOI: 10.2197/ipsjjip.21.304]

Regular Paper

An Object-Defined Remote Robot Control Interface

Yusuke Suzuki[1,a)] Motoki Terashima[2] Koichi Takeuchi[1] Hideaki Kuzuoka[2]

Received: June 22, 2012, Accepted: January 11, 2013

Abstract: In this paper, the authors propose a new remote robot control interface that reduces the complexity of robot control. The proposed interface constraints the robot's movements depending on the target object that the operator wants to observe. The interface displays the constraints to the operator on a screen with the help of Augmented Reality (AR) technology. We named the interface "Object-defined remote robot control interface" because the interface provides suitable procedures for the objects that need to be operated on. The interface receives information about the robot and candidate objects from a camera that has been set up to capture a bird's-eye view of the target environment and displays this information on a touch screen display. When the operator selects an object as the target by touching it on the display, constrained tracks for the robot's movements and their corresponding AR representations are generated on the screen. A block assembly task was conducted to evaluate this interface. The results showed the system's effectiveness in terms of both task completion time and operation time.

Keywords: robot interface, tele-robotics, augmented reality, block assembly task

1. Introduction

Remote-controlled robot systems have been proven useful in many areas of applications. For example, remote-controlled robot systems can be used where it is dangerous for a human to be present, such as, conducting surveillance inside nuclear power plants [1] or performing activities in space [2]. Such systems can also be used to conduct efficient remote surveillance of distant spaces for remote museum visiting [3], and teleconferencing [4], [5], [6]. In these applications, the remote control system needs to enable the operator to easily observe objects located in the remote environment. Currently, some remote-controlled robot systems for daily use are commercially available [7]. It is possible to predict that in the future, there will be a growing number of such users who need to observe remote objects in distant places. However, previous studies have shown that remote-controlled robot systems have common problems yielded from a complex control of complex robot functions.

To solve these problems, We have used augmented reality (AR) technology to enhance the image from a bird's-eye view camera as an input to the remote control interface. We introduce a new concept here that applies to such remote control interfaces. According to this concept, based on the operator's goals and a target object for an operation, the interface allows only certain suitable procedures to accomplish the desired task. We named our proposed interface "Object-defined remote robot control interface."

In this paper, we have two goals: one is to propose the "Object-defined remote robot control interface" for remote surveillance and the other is to show its effectiveness by user tests. In this paper, we present our work in the following order: In the second section, we introduce some previous researches on the current topic and some common problems for remote-controlled robots. In the third section, we propose the "Object-defined remote robot control interface" that could solve the common problems. In the fourth section, we explain the experiment conducted to test the effectiveness of the new interface. In the fifth section, we summarize the results of the experiment. In the sixth section, we discuss our findings and future works. And in the last section, we summarize the paper and present our conclusion.

2. Problems of Remote Robot Control Interfaces

Existing remote robot control interfaces have some common problems which can be summarized as follows.

- Problem 1. lack of "situation awareness"
- Problem 2. unpredictability of the result of the operator's actions
- Problem 3. complexity of control due to the complexity of robot functions

2.1 Problem of Lack of Situation Awareness

Yanco et al. [8] have defined "situation awareness" of a robot as follows:

> "the perception of the robots' location, surroundings, and status; the comprehension of their meaning; and the projection of how the robot will behave in the near future."

However, as previous researches have shown, employing cameras that are simply mounted on robots is not good enough to acquire "situation awareness." Due to limited sight range of the cameras, the acquired information is also limited. Therefore, when an operator controls the robot moving in a remote site which is unfamil-

1 Oki Electric Industry Co., Ltd. Corporate Research and Development Center, Warabi, Saitama 335–8510, Japan

2 Graduate School of System and Information Engineering, University of Tsukuba, Tsukuba, Ibaraki 305–8573, Japan

a) suzuki543@oki.com

iar to the operator, the operator has to spend much time to investigate the remote site which results in inhibiting the operator from carrying on with actual tasks. To solve this problem, previous researches have proposed adding a camera for acquiring information about the relation between the robot and a situation at the site where the robot is located to a robot controlling system. Murakami et al. [9] have reported that the robot control performance had improved when cameras were installed on the ceiling so that the operator could get a bird's-eye view image of the place where that robot was located. Shiroma et al. [10] also have reported that installing an omni-view camera or a fish-eye camera on top of the robot enables the operator to get complete information about the situation and improves the task performance. Installing cameras in the ways discussed above has reduced the number of times the robot bumped into objects in the environment and has reduced the number of commands that the operator had to send to the robot to complete a task.

As another solution, Sugimoto et al. [11] have proposed "Time follower's vision." Using mixed reality technology, their system enables an operator to see the robot's position and orientation superimposed on the background image of the remote environment with the help of Computer Graphics (CG). The background image of the remote environment is taken by the robot-mounted camera. The operator can see the robot as if the camera had been located away from the robot and can capture images that reveal information about the robot's environment.

As mentioned above, previous researches have shown that: camera views that help capture detailed information about the remote environment are effective for acquiring situation awareness.

2.2 Problem of Unpredictability of the Result of the Remote Control

When using remote-controlled robots, the unpredictability of robot movements that result from operator inputs is another problem. This problem becomes more apparent especially when a time-delay for exchanging data is significant. Consider a case where an operator on Earth is trying to control a robot in space. There is a large transmission delay due to the great distance between the operator and the robot. This delay prevents the operator's commands from resulting in robot movements within an acceptable amount of time. This inhibits the operator from controlling the robot smoothly.

Applying CG is a common solution to this problem. Superimposing CG that draws an expected image of the robot at the point of completion of the control enables the operator to easily understand how the robot's position and the environment will change after the control command finishes executing [12], [13], [14].

Even if the transmission delay is small enough, the same problem arises if a relation of the operator's inputs to robot's outputs is complicated. CG could again be a solution to this problem as shown by previous researches. Arai [15] and Tsumaki et al. [16] have reported the cases of a non-holonomic robot control and an acceleration command based robot control respectively. In these cases, the operator cannot smoothly control the robot because operator inputs and robot movements are neither easily associated nor predicted. However, as they have reported, applying CG to provide visual information cue to show the association on the user interface has improved the usability of the system and task performance.

In addition, applying a bird's-eye view image to provide the situation awareness could have a side effect that could make the operator input and robot movement relationship complicated. Consider a case where the operator controls a robot with a joystick as an interface device while watching a video that provides the view of the remote environment as seen from vertically above. In such a case, the operator constantly sends control commands to the robot in the same manner, however, robot movement directions could vary depending on the robot's current orientation at any given point of time. The operator has to deal with this mental rotation problem [17]. The operator has to predict the robot's movement direction, convert it to input direction and control the joystick accordingly. This mental process could be a burden and can cause mistakes such as moving the robot in an unintended direction.

Applying a bird's-eye view image as the control interface and AR technology would solve this unpredictability. When an operator points to a position on the screen, the system moves the robot to the relevant position in the remote environment [18], [19], [20]. In the case of such a process, the operator does not have to worry about maintaining the relationship between the joystick directions and the robot orientations. In addition, combining the bird's-eye view image and CG, in other words, applying AR technology takes advantage of both the proposed solutions.

2.3 Complexity of Control Due to the Complexity of Robot Functions

A robot's functions vary according to its usage. An operator has to select the appropriate function from the various possible functions to accomplish the desired task. However, when there are too many functions to choose from due to the complexity of robot elements, it could worsen the robot control usability. For example, a humanoid could have approximately 30 Degrees of Freedom (DOF) which would make it difficult for the operator to manually control the robot with a conventional interface such as the control board of an industrial robot.

To solve this problem, Sian et al. [21] have developed a system in which the operator controls a 30 DOF humanoid with two joysticks. They have proposed a method that constrains the robot's functions based on what the operator wants to do. In their system, at first the operator chooses one humanoid joint to pay attention to, and then the system automatically restricts the movement range of the other joints. This restriction could reduce the need for the operator to pay attention to the other joints. In other words, the system always constrains the movement ranges of the robot's joints and allows the operator to choose and control only one joint at a time. This constraint simplifies the operator's work with the control device/interface. In this way it is possible to solve the problem of complexity of the controls due to the complexity of robot functions.

3. Object-defined Remote Robot Control Interface

Based on previous literature survey in the previous section, we propose a new remote robot control interface that has the following three features:

- Feature 1: To provide situation awareness to the operator, a bird's-eye view image of the remote environment is provided (Refer Section 2.1).
- Feature 2: To reduce unpredictability of the result of remote control, the bird's-eye view image superimposed with the robot's CG image (AR technology) on the robot control interface is presented (Refer Section 2.2).

These two features are derived from the common solutions found in previous researches. The feature 3 reflects our new proposition which is inspired by the research that constrains the robot's movements depending on the operator's objectives.

- Feature 3: To reduce complexity of robot control, the robot's movements are constrained based on the relationship between the robot and the target object in the remote environment (Refer Section 2.3).

In this paper, we mainly focus on the effect that yields from Feature 3 which reflects the most important contribution of our work.

3.1 Features of the Proposed Interface

One of the features of the interface that we propose in this paper is to constrain the robot's movements based on the relationship between the robot and the target object for the robot's operation. Too many functions in the robot tend to complicate the robot control. In such cases, constraining robot movements could solve the problem. That is, restricting unnecessary movements for a particular objective may not only prevents the operator from wasting any time making choices between too many possible operations, but it also eliminates the need for extra operations required for recovering from control mistakes.

Note that in this paper, "to constrain" means to control the robot's actuators in a certain manner that results in semi-automatic movements in the robot. They include constraining the robot's vehicle tracks to certain orbits and setting the robot-mounted camera's orientation in a certain direction.

The principal concept behind the proposed interface is the constraining of the robot's movements based on the operator's requirements and the visualization of these constraints using AR technology. The constraints are drawn with CG and are superimposed on the bird's-eye view image.

Next, we need to decide what constitutes "proper" constraints. Constraints vary depending on the situation. In Ref. [21], all humanoid joints are constrained and the operator can choose a joint to be released from the constraint. In such a system however, the operator has to decide the joints to be released and therefore must have deep knowledge and understanding of the robot's mechanism.

We propose a new method here that will free the operator from this requirement. In the proposed method the operator does not have to understand the robot's mechanism. The operator is only required to decide which objects should be paid attention to, that is, the target object in the remote environment for observation by the robot. Then the interface system decides on the "proper" constraints based on the object that the operator selects.

The idea to constrain the robot's movements based on the relationship between the robot and the target object seems natural. This is because remote-controlled robots are usually used to gather information about the target object such as the physical appearance and form which is very difficult to represent exactly in digital form.

In the proposed method, first the operator selects the target object. This informs the system that it needs to gather information about this object. Then the system decides which movements of the robot should be constrained. Finally, it provides a user interface which allows the operator to control only constrained movements of the robot. We named this new interface method as "Object-defined remote control interface" because our method lets the objects provide appropriate interfaces for a user to observe them.

The Object-defined remote control interface has the potential to be used for an expandable range of tele-operations of a remote robot control. In this paper, however, we focus on the remote surveillance which controls a robot-mounted video camera and the robot's movements under the appropriate constraints for observing the target object.

To evaluate the proposed method, we implemented the system using a robot that was previously developed [22]. This robot has two wheels and is mounted with a video camera. The robot can transmit video images to the operator through a wireless LAN connection. The system basically enables the operator to control the robot which is placed in the remote environment. The operator uses the robot-mounted camera to capture video images to gain information about the form and location of the target object. The system then considers the relationship between three elements in the remote environment: firstly the form and location of the target object, secondly the location of the robot, and thirdly the orientation of the robot-mounted camera with respect to the target object. Based on the information collected about this relationship, the system decides the robot's desired trajectory to reach the target object and therefore generates the appropriate constraints to limit the robot's movements. Based on the above relationship information, the system also imposes some semi-automatic constraints on the robot-mounted camera rotations to acquire the desired images of the target object and the remote environment. Once the operator gains situation awareness about the remote environment and indicates the target object by using the "Object-defined remote control interface," the system automatically generates the above two types of constraints.

3.2 System Overview

Drawing upon the system configurations of the previous studies [24], [25], we have been developing a system as shown in **Fig. 1**. The system is composed of the robot, the bird's-eye view camera located on the ceiling, and the touchscreen display that we use as the robot control interface.

Figure 2 shows the robot used for the experiment. This robot

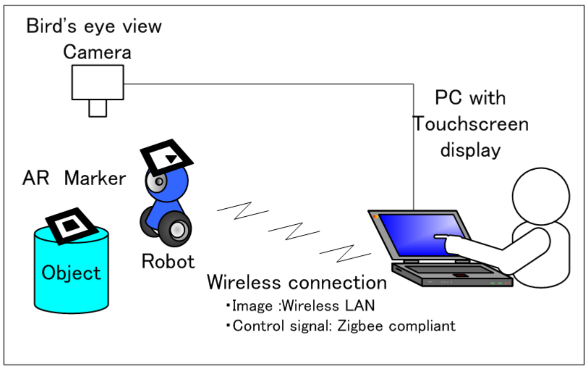

Fig. 1 System overview.

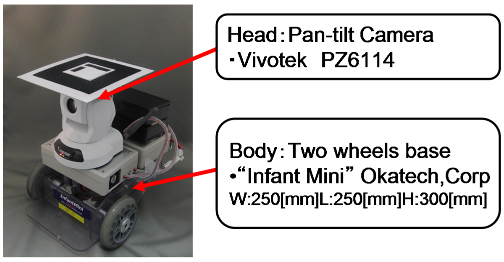

Fig. 2 Robot overview.

has two wheels and is mounted with a video camera. The robot is about 250 mm in width, 250 mm in depth, and 300 mm in height. The robot-mounted camera is a pan-tilt camera PZ6114 (Vivotek Corp.). This camera can transmit video images to a PC via wireless LAN. The vehicle part is Infant Mini (Okatech Corp.). It has two active wheels and a passive, free-moving caster wheel. The camera and the vehicle are powered by an on-board battery. Commands for the robot and video images are transmitted via wireless LAN to a controlling PC. This allows the robot to operate wirelessly.

We placed two-dimensional visual markers on the robot and in the neighborhood of the objects to be captured. Using ARtoolkit [23], the system recognizes marker information such as ID, location, and orientation from the image captured by the bird's-eye view camera. With this information, the system prepares a CG image and superimposes it on the image from the bird's-eye view camera. The system searches an ID database to find the required robot trajectory. The system then generates the appropriate constraints for the robot's movements.

We can prepare for some basic movement constraints and apply them when a new object is registered. In this system, basic movement constraints are applied to both the robot as well as the robot-mounted camera. For example, when the target object is a kind of a block-shaped object, then the robot needs to circumnavigate the target object in order to capture images of it from all sides. The smallest possible radius for the robot's movement around the object can be manually defined by the operator later. When registering a new object with a complex shape, the operator can place an unregistered marker on the new object and register the relevant movement constraints with the system. For such cases, it is necessary to provide an editor that allows the operator to register new robot trajectories and relevant movement constraints while the robot is in motion. This will be a part of our future work.

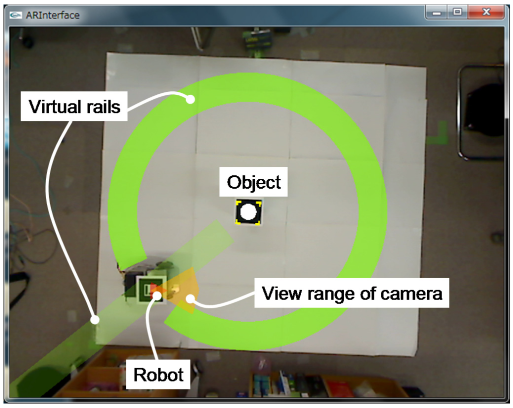

Fig. 3 Screenshot of remote robot control interface.

The bird's-eye view image helps the operator gain situation awareness and therefore solves problem 1. In our system, the operator can directly indicate the target object to the system by touching it on the interface screen and thereafter the system automatically generates the appropriate constraints and makes the end result of the robot operation predictable. This solves problem 2.

In Section 3.3, we explain how we deal with the problem 3.

3.3 Constrains Applied to Robot Movements and its User Interface

In this section, we will explain the constraints applied to the robot's movement and their visualization using AR technology. We will also describe in detail how the system generates the movement constraints for both the robot and the robot-mounted camera based on the relationship information about the robot's location and orientation, and the location of the target object.

First, the operator chooses the target object from the bird's-eye view image. The system then acquires the information about the location and orientation of the marker allocated to the target object and that allocated to the robot. The system then calculates the relationship between the target object and the robot based on the location and orientation information. Then the system generates the movement constraints and displays the constraints to the operator by superimposing the CG image on the bird's-eye view image on the interface screen.

More specifically, **Fig. 3** shows the user interface for the operator. As shown in the figure, the system draws a line and a circle in green color with CG (virtual rails) on the bird's-eye view image. The line connects the target object and the robot. The circle is drawn around the target object. The radius of this circle is the distance between the target object and the robot. The robot moves along the circle to circumnavigate the target object. This robot trajectory is displayed to the operator using CG on the interface screen.

The CG image shows "virtual rails" on which the robot can move. This enables the operator to easily understand the robot's trajectory and the location of the robot at the end of each operation. The operator directly touches a point on the virtual rail on the touchscreen interface and then the robot moves to the corresponding point in the remote environment. If the operator touches a point on the line connecting the robot and the target object, then a red dot appears at the point and the robot moves forward or

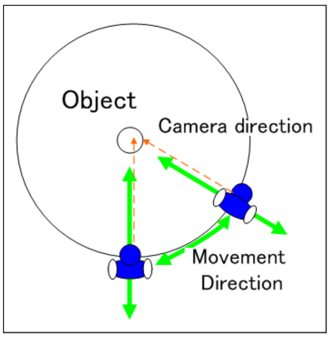

Fig. 4 Movement directions of the robot.

backward on that line. If the operator touches a point on the circle, a red point also appears at the point and the robot moves along the circle as if it were circling around the target object. The robot only moves until it reaches to the red point. Moving directions are indicated with bold arrows in **Fig. 4**. Touching any places on the screen but the green virtual rails and the target object terminates the robot's motion if it is moving. Otherwise, it causes nothing.

While the robot is moving along the constrained trajectories (the line or the circle), the system always controls the orientations of the robot and the robot mounted camera so that the camera can capture clear image of the target object at all times (dotted arrows in Fig. 4). When the robot is moving along the line connecting the robot and the target object, the robot-mounted camera is adjusted to orient toward the target object by facing in front. When the robot is moving along the circle, the robot-mounted camera is semi-automatically oriented toward the center of the circle where the target object is located.

Combining robot movement constraints with the constraints on the semi-automatic camera rotation movements enables the operator to capture an image of the target object from any position without having to think about it. "Virtual rails" change automatically every time the operator chooses a different target object. Movement constraints for the robot and the robot-mounted camera also change accordingly.

Using this interface, the operator can capture video images of any target object that has a complicated form. The operator can acquire images of such a target object from different angles by simply indicating a point on the touchscreen interface. By just touching a point on the virtual rail the operator can move the robot to that point and capture images of the target object from there. The operator can get either overall view or the close up view of the target object simply by touching the appropriate location of the line trajectory.

The rest of the information required to control the robot is also drawn on the bird's-eye view image as CG. As shown in Fig. 3, the small circle in the center depicts the location of the target object, the triangle depicts the location and orientation of the robot and the fan shape depicts the field-of-view of the robot-mounted camera as well as the its orientation.

4. Evaluation

We conducted an experiment to evaluate the effectiveness of the object defined remote control interface. Since the problem 1 and 2 are supported by the bird's-eye view camera and the AR visualization, we especially focus on how our system supports the problem 3.

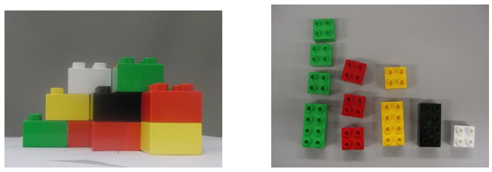

Fig. 5 LEGO®blocks used for the experiment.

4.1 Outline of an Experiment

Participants were asked to complete a task by operating the robot using the proposed interface method and another method. We used the block assembly task, which is a common task in CSCW researches [26], [27]. The robot and an assembled LEGO®block structure (**Fig. 5**) are placed somewhere that is not directly visible from the participant. A visual marker is placed on the block structure such that the participant cannot see it from the bird's-eye view image. The participants were asked to control the robot with each of the two interfaces to look at the block structure that has been placed in the remote environment using the robot-mounted camera and to assemble the blocks provided to match the assembled structure in the remote environment. We think that this task is suitable for evaluating the usability of our system because participants have to look at the blocks in detail to complete the task and from the task results we can measure whether participants inspected the target objects precisely.

We measured the task-completion time and counted the number of error occurrences to analyze how our proposed method facilitates the robot operation compared to the other method that we will explain later. We also conducted subjective assessments.

In this experiment, we focused on confirming the fundamental effects of our proposed method and so we used only one target object. The evaluation for the case with multiple target objects is a part of our future research.

4.2 Interface for Comparison

For the purpose of comparison with the proposed interface, we used an interface that is similar to the interface which Sekimoto et al. have proposed [18]. That interface applies a bird's-eye view image and a touchscreen display as an input device and the robot moves to the point that the operator indicates on the image. The robot orients toward the point indicated by the operator and moves to that point. The point indicated by the operator is depicted as a red dot by CG superimposed on the bird's-eye view image while the robot is in motion and disappears once the robot reaches the indicated point. The camera orientation does not change when the robot's orientation or location changes. The camera's field-of-view is depicted as a fan just like in our proposed interface. The operator can suspend the robot's movements just like in our proposed interface.

This interface is composed of similar system elements as our proposed interface and has solved Problem 1 and 2 described in Section 2 of this paper. However, this interface does not have the following features that our system offers, as mentioned in Section 3:

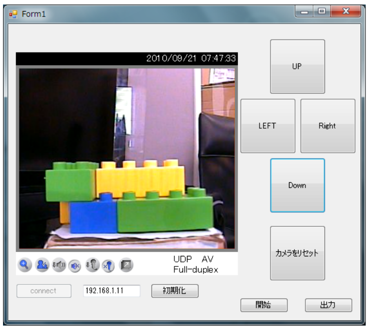

Fig. 6 Screenshot of camera control interface.

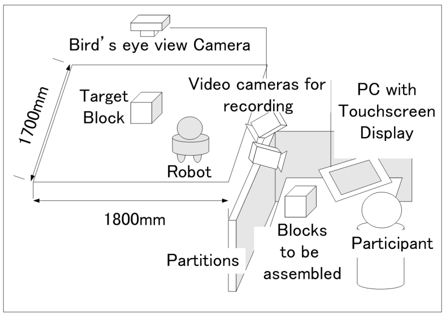

Fig. 7 Experimental setup.

- System-generated constraints for both the robot's movements as well as the robot-mounted camera rotation movements.
- Depiction of the system-generated constraints as CG superimposed on the bird's-eye view image.

Except for these two features, the rest of the features of the other interface, henceforth known as the "comparison interface" in this paper, are the same as in our proposed interface. Therefore, we think that it is a suitable candidate for comparison with our proposed interface. It is reasonable to think of evaluating the effect of one of the above features at a time through our proposed interface. However, in this research, our goal is to evaluate the proposed method in comparison with other methods that do not have these two features. To examine the effect of each feature separately is a part of our future research.

4.3 Camera Control Interface

For both robot control interfaces, we also provided an interface to manually rotate or tilt the robot-mounted camera (**Fig. 6**). This interface has five buttons: "UP," "DOWN," "RIGHT," "LEFT" and "Reset the camera." Pressing any of the direction buttons rotates or tilts the camera in the appropriate direction by a constant angle value. Pressing the "Reset the camera" button turns the camera to the initial orientation, that is, facing the front.

4.4 Setup

Figure 7 and **Fig. 8** show the experimental environment and Fig. 8 shows interfaces arrangement image on the display. The participant sits at a desk in a place from where he or she cannot directly see where the robot and the assembled block structure are located. The participant controls the robot using a touchscreen display connected to a PC. Video images captured by the robot-mounted camera are also displayed on the touchscreen interface

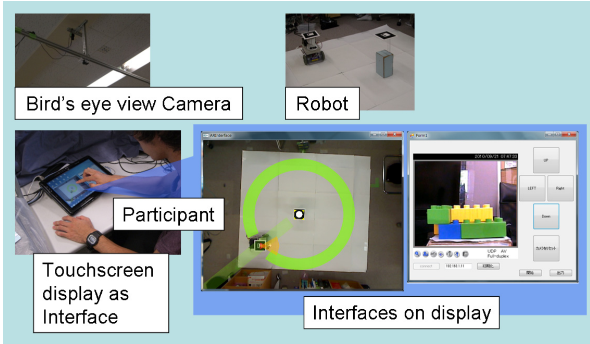

Fig. 8 Experimental environment.

as a part of the camera control interface (Fig. 6).

The PC transmits commands as serial signals to the robot wirelessly. The robot receives these commands, interprets them, and moves. The area within which the robot moves is 1,800 mm × 1,700 mm; this is the maximum area coverage possible under the bird's-eye view camera's range. We set up two video cameras. We recorded the robot's movements and the participants' operations for analysis of the results. We also used the operation data log for the robot-mounted camera control interface, the video captured by the robot-mounted camera, the video recording of the robot movements during the experiment, and the video recording of the participants using the two interfaces for evaluating our proposed interface.

4.5 Participants

31 students participated in the experiment. Students included both undergraduate and graduate students. Valid data for 28 participants were analyzed by excluding the cases of system failure. 26 of the analyzed participants were in their twenties and two of them were teenagers. 16 of the participants were male and 12 were female. 26 of them were right-handed and two of them were left-handed (self-reported).

4.6 Procedure

The experiment was conducted in a room at the university. Participants entered the room and saw the robot and the area where the robot would move. Then we explained them the procedure of the experiment. We also explained them how to control the robot-mounted camera and the robot interface. Finally, we explained about the block assembly task (Refer Section 4.7). After training on how to use the controls, they conducted a practice trial. After the practice trial, they conducted the actual experimental task. After a five-minute break, the participants performed the same task again with the other interface. We followed the same sequence as before, that is, the explanations, followed by the training, then the practice, and finally the actual task with the other interface. The order of two interfaces was counter balanced among the participants. After the experiments, the participants were asked to answer a subjective assessment questionnaire (Refer Section 4.8).

4.7 Task

Two sets of blocks (eleven pieces, five colors) were used in this experiment task. One set was assembled into a structure

Journal of Information Processing Vol.21 No.2 304–314 (Apr. 2013)

and placed in the remote environment and another set was provided to the participants (Fig. 5). We prepared two block structures and confirmed through prior lab experiments that they were both equally difficult to assemble.

The participants conducted a simplified task (four blocks in three colors) as a practice trial. The participants were also asked to press a button on the interface display to indicate when they began or finished the task. Thus, the system was able to record the task completion time.

The participants were instructed that the speed of task completion is preferred over task accuracy. However, they were also motivated about the accuracy of the task by informing them that the participant who assembles the block structure correctly in the least amount of time will receive a special prize. They were also instructed to neither bump the robot against the block structure in the remote environment nor move the robot out of the working area, that is, the area covered by the bird's-eye view camera. All of the participants were paid regardless of their performance or results.

4.8 Questionnaire

The following five subjective assessment items were analyzed.

- Q1. Was it easy to understand the form of the assembled block structure by operating the robot?
- Q2. Did you worry about bumping the robot against the block structure?
- Q3. Could you move the robot to the place where you wanted to move it to?
- Q4. Could you easily control the robot?
- Q5. Did you need much time to get used to controlling the robot using this interface?

All the questions were rated on a 5-point scale (where 1 = completely disagree, 3 = neutral, and 5 = completely agree). Each trial of the experiment took about one hour including breaks [*1].

4.9 Measures

We performed a statistical analysis of the task completion time, the number of times the participant performed an operation to control the robot (num_robot), the time taken by the robot to move (time_robot), the number of times the participant performed an operation on the camera control interface (num_camera), and the time taken by the camera to move (time_camera).

The number of times the participant performed an operation to control the robot (num_robot) was measured by the number of times the participant touched the touchscreen interface to perform a robot operation. The number of times the participant performed an action on the camera control interface (num_camera) was measured by the number of times the participant pressed a camera control button on the interface. The time for robot movement was measured as the time that lapsed after the participant touched the interface display till the robot stopped moving (time_robot). Similarly, the time taken for the robot-mounted camera movement was measured as the time between the moment a camera control button was pressed and the moment when the robot-mounted camera image stopped moving (time_camera). These items were measured by observing the two video recordings, one that was focused on the robot and the other that was focused on the participant.

[*1] Although we expected the difference between the answer values of the two interfaces because of the features of the proposed method, we asked this question anyway to confirm whether participants were able to understand the intended effect of the proposed method and to confirm the extent of its influence through the results of the subjective assessments.

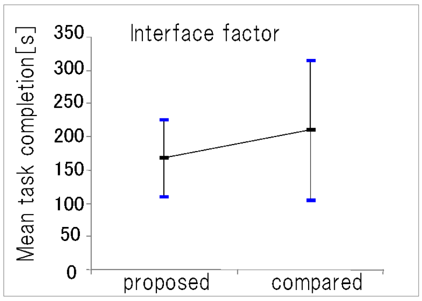

Fig. 9 Mean task completion time for the two interfaces.

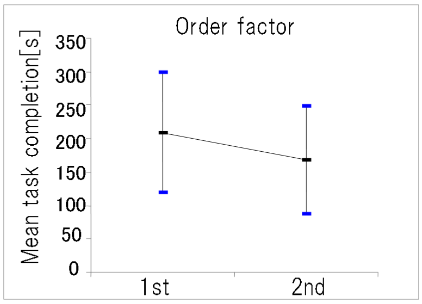

Fig. 10 Mean task completion time for the order in which the trial was experienced.

We also counted and analyzed the number of error occurrences. An error was defined as the case in which the participant was not able to complete the task within 10 minutes or was not able to assemble the blocks to match the assembled block structure in the remote environment. The task completion time given to the participants who were not able to complete the task was set to be the same as the longest task completion time among the participants who were able to complete the task.

To examine the effect of each interface and the effect of the order of the interface which the participants experienced, we evaluated the mean values of the task completion time measurements by means of a two-way ANOVA. Interface factors are the proposed interface and the comparison interface (**Fig. 9**). Order factors correspond to whether the the task is first or second in order (**Fig. 10**). We also statistically analyzed the subjective assessments.

5. Experimental Results

5.1 Task Completion Time

Figure 9 and Fig. 10 show means of the task completion time on interface factors and order factors respectively. **Figure 11** shows the interaction of both factors. The horizontal lines show the levels of each factor and vertical lines show the time. Error bars mean standard deviations.

The mean task completion time for the proposed method and the comparison method were 168 seconds and 210 seconds re-

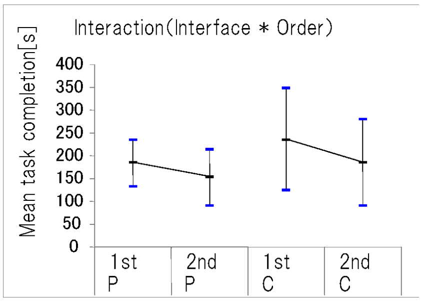

Fig. 11 Interaction of interface factor and order factor.

spectively. As for the order factors, the mean task completion time for the first task was 209 seconds, and that of the second task was 168 seconds. As the result of ANOVA, the main effect of interface factor and order factor show marginal significance. (Interface: $F(1, 52) = 3.56$, p = 0.0648, Order: $F(1, 52) = 3.36$, p = 0.0723). The interaction was not significant. (Interaction: $F(1, 52) = 0.183$, p = 0.670). According to the Levene test, the variance of mean task completion time in both levels of the interface factor was not equivalent.

For the interface factor, the task completion time differences between the comparison method and the proposed method (comparison - proposed) was confirmed to be normally distributed by the Kolmogorov - Smirnov test. A paired t-test for mean task completion time difference of the comparison method and the proposed method also shows marginal significance (t(27) = 2.02, p = 0.0532)).

5.2 Error Occurrences

In 56 task trials by 28 people, we observed 10 errors (9 times incorrect block assemblies, one time task incompletion within the time limit). Participants made errors only in one of the interface factors but none made errors in both interfaces. Comparing between interface factors, the proposed method saw 3 errors and the comparison method saw 7 errors. Concerning the order factor, both the first trial condition and the second trial condition have 5 errors each and thus no difference was found.

5.3 Number of Robot Operations and Robot Movement Time

For the interface factor, the mean num_robot for the proposed method and the comparison method were 9.96 (S.D. 4.81) and 16.8 (S.D. 11.3) respectively. Paired t-test showed significant differences at 1% level ($t(27) = -3.16$, $p < 0.01$). The mean time_robot during the proposed method was 48.0 seconds (S.D. 20.3) and during the comparison method it was 44.8 (S.D. 20.3). Paired t-test showed no significant differences (t(27) = 0.706, p = 0.486). The mean number of times the robot moved during a single trial using the proposed method was 8.96 (S.D. 4.11) and the comparison method was 10.2 (S.D. 5.52). Paired t-test showed no significant differences ($t(27) = -1.03$, p = 0.312). Among the controlling commands, the command to suspend all robot movements during the proposed method was 8.20% (S.D. 1.00) and the comparison method was 30.0% (S.D. 6.64).

Table 1 Summary of subjective assessments.

	Proposed (median.)	Comparison (median.)	p-value
Q1	4	4	$p = 0.0401^{*}$
Q2	1	4	$p < 0.01^{**}$
Q3	4	4	$p = 0.0464^{*}$
Q4	5	4	$p < 0.01^{**}$
Q5	1	2	$p < 0.01^{**}$

$*$ significant at 5% level, $**$ significant at 1% level

5.4 Number of Camera Control Operations and Camera Movement Time

Mean time_camera during the proposed method was 0.257 seconds (S.D. 0.793) and that during the comparison method was 11.5 seconds (S.D. 9.34). Paired t-test between the two showed significant differences at the 1% level (t(27) = 6.21, $p < 0.01$). Camera movement time per command during the proposed method was 1.11 seconds (S.D. 0.261) and that during the comparison method was 1.15 seconds (S.D. 0.236).

5.5 Correlation between Task Completion Time and the Number of Camera Operations

For the interface factor, Pearson's product-moment correlation was used to examine the relationship between the difference of mean task completion times for each condition (comparison - proposed) and the number of camera operations. A moderate positive correlation was observed ($r = 0.429$).

5.6 Subjective Assessment

Table 1 shows the results of the subjective assessments. For the interface factor between the two conditions, Wilcoxon signed rank test was conducted for each question. Q1 and Q3 showed at 5% level and Q2, Q4, Q5 showed at 1% level of significant difference.

5.7 Operational Tendencies of the Participants

In the case of the proposed method, totally seven camera operations were observed by 3 persons among 28 participants. However, most of them seemed to be thoughtless operations and the results of the operations were overwritten by the next operation. Therefore, we could not find any specific tendency.

Some of the participants who used the proposed method before the comparison method tried to operate the robot so as to circumnavigate the target object even when they were using the comparison method. Some participants examined the blocks while operating the robot; they made the robot move along the circle and while observing the changing images from the robot-mounted camera they inspected the block they had in their own hands and confirmed their shape and form. This operation was common; it was observed in 15 out of 28 participants while using the proposed method. When using the comparison method although some participants tried to do a similar operation, the number of such participants was limited (2 out of 28). We asked the participants about their experiences with 3D games or block toys, however, we found no effect of such experiences.

6. Discussion

From the results of the experiments we can see that there is a marginally significant difference between the proposed method and the comparison method. With respect to the num_robot, the num_camera, and the time_camera, there were significant differences. The mean of all answer values of subjective assessments showed a significant difference between the two methods. The main effect of the order factor showed marginally significant difference, however, interaction effect with interface factor was not significant. Also, since we balanced the order among participants, we think that the order in which the participants experienced the proposed method did not influence the results.

6.1 Constraints Decrease the Number of Control

We could show that the decrease in time_camera explains the faster task completion time when using the proposed method. However, the mean time_camera when using the proposed method only accounts for a 10-second decrease and therefore could not suffice as an explanation for the reduced mean task completion time of 40 seconds.

We also investigated the decrease in num_robot which was significantly less when using the proposed method. Then, we further investigated the number of robot movement suspension commands and it was also significantly less in the proposed method. However, neither the number of robot movements nor the time taken by the robot to move showed no difference between two interfaces. From these results, we could say that in the proposed method, participants are less likely to suspend robot movements before it reaches a goal position than in the comparison method.

Anticipating the best position to observe the assembled block structure before issuing a movement command to the robot may have been a difficult task for many participants. When using the proposed method, since the number of possible goal positions was limited by the constraints imposed by the system, there was a reduced need to make such decisions. While using the comparison method, many participants were observed to give commands that would make the robot move a long distance and then they would suspend the robot's movements mid-way. This strategy became noticeable especially when the robot moved close to the assembled block structure.

Combining this observation with the result of the question 2 (Q2), we could speculate that the participants were afraid of bumping the robot against the assembled block structure when using the comparison method. This kind of operation forced the participant to keep an eye on the robot's movement in the bird's-eye view image. In consequence, the participant could neither observe the block structure in the remote environment in detail nor could he or she assemble the blocks while the robot was in motion.

When the proposed method was used, the robot moved along a circular path around the target object. Hence, the participants did not have to worry about bumping the robot against the assembled block structure and therefore they did not need to suspend the robot's movements that frequently.

We speculate that when using the proposed method, the participants were able to observe the block with greater attention even while the robot moved around in the remote environment. In fact the participants were observed to carry on with the task of assembling the blocks while the robot was moving. We are assuming that these two features of the proposed method influenced the task completion time difference.

6.2 Constraints Reduce the Errors

Because the number of data was limited, we could not conduct statistical analysis of the error occurrences. However, error occurrences in case of the proposed method were fewer than in the case of the comparison method.

To investigate the relationship between the number of error occurrences and the interface factor we closely observed the video recordings of the participants' operations. Some participants moved the robot along the circular path after they had finished assembling the blocks on their side to confirm whether the assembled block structure in the remote environment resembled the block structure that they had assembled. In case of the comparison method, some participants answered in the interview that they did not confirm the accuracy of the block structure that they had assembled because the robot operation was troublesome and therefore they did not bother to operate the robot again to confirm the accuracy of their block structure.

This kind of an answer, combined with the fact that there were fewer error occurrences when using the proposed method lead us to believe that the proposed method was easier to use. Since more participants checked the accuracy of their block structures in the case of the proposed method than in the case of the comparison method, there were fewer error occurrences in the case of the proposed method. It is interesting to find that the proposed method influenced the task-accuracy-checking behavior of the participants.

6.3 Other Results

The proposed method displays more information such as superimposed CG information and so on than the comparison method. This could have made the participants feel that the operation using the proposed method was more difficult than the comparison method. Yet, the results of the assessment of the answer values of question 4 and question 5 (Q4 and Q5) showed that it was just the opposite. Although value differences themselves were slight, we could confirm that the participants felt that the proposed method was easier to use and easier to learn because of the statistically significant difference between the answer values of the above two questions for the two interfaces.

We could see from the results of the Levene test, the task completion time variances with the proposed method were significantly smaller than with the comparison method. Therefore we can say that the proposed method is more stable and is independent of the participant's skill level.

6.4 Limitations of the Proposed Interface

There might be some situations where the operator might have to control the robot without constraints. In such cases, the system should enable the operator to switch between two modes; object-defined remote control mode, and free movement mode. Such existence of controlling modes often bother operators. And the operator should always be aware of the present mode. This might

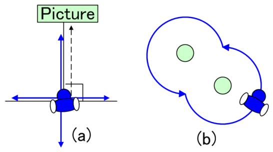

Fig. 12 Example of different movements.

be another burden for the operator. Whether the system should switch modes automatically or whether the operator should select the mode manually remains to be investigated. A suitable notification of the present mode is also one of our future works.

Registering objects before using the system also might cause problems. It is a part of our future work to find out how we can define the appropriate constraints depending on the target object, maybe automatically generating suitable tracks depending on the shape of the target object. At the moment, we think we can build an editor that allows users to edit and to register movement constraints to the object. If an object to be registered has a common shape, a user can choose the required constraints from basic options (such as straight or circular paths) and can combine these constraints, so the registration would not be a major burden. And in case of tasks to be repeated (such as remote surveillance), once objects are registered, a user can use such constraints repeatedly so the benefits which the user will gain surpasses the registering cost.

6.5 Applications and Future Works

Our results show that the proposed method is effective for a certain observational motion, i.e., a circular motion around a single object. Although we cannot immediately generalize this result to a user interface for all kinds of tele-operation robots. However, we think object surveillance is a fundamental task and there are a lot of scenes where our method can be applied. For example, a remote surveillance of an industrial plant, a remote museum visiting, a tele-conference, and so on.

In this experiment, we applied two relatively simple movement constraints: straight path and circular path. It is a part of future work to find out how we can decide on the appropriate constraints depending on the target object, the operations that should be allowed, whether we should generate suitable tracks depending on the shape of the target object and so on.

Figure 12 (a) is an example that shows how the motion constrains might be applied for the remote museum visiting robot. Due to the constraint, a user can always observe the paintings from the best angle while not worrying about bumping into the precious exhibits.

In this experiment, we investigated the case where there was only one target object in the robot's working field. Figure 12 (b) we show an example case where constraints of the closely placed object are merged automatically. Developing an algorithm for merging several nearby objects is also our future work.

7. Conclusion

In this paper, we summarized problems that are common in remote robot controlling and proposed a new interface method; an Object-defined interface as a possible solution to the problems. This interface features a bird's-eye view image, a direct manipulation interface using a touchscreen display that shows the bird's-eye view image and the movement constraints depending on the target object that the operator focuses on and visualizes these elements on the interface screen using AR technology.

We implemented the proposed method as a control interface for a robot that is mounted with a camera. To investigate the effect of movement constraints and their visualization, we conducted an experiment comparing the proposed method with a reference interface method which does not have these features.

From the experiment results, we confirmed that the proposed method could reduce the task completion time, reduce the number of error occurrences, make the users feel that they could easily control the robot, and help users get easily accustomed to the usage of the control interface.

We will try to adopt this finding in other robot controls. Our goal is to provide a new remote robot control interface that can enable a lot of non-specialist users to easily control remote robots.

Acknowledgments We thank all the participants.

References

[1] Mano, T. and Hamada, S.: Development of Robotic System for Nuclear Facility Emergency Preparedness, *Journal of the Robotics Society of Japan*, Vol.19, No.6, pp.38–45 (2001).

[2] Wakita, Y., Hirai, S., Machida, K., et al.: Application of Intelligent Monitoring for Super Long Distance Teleoperation, *Proc. IEEE International Conference on Intelligent Robotics and Systems*, Vol.3, pp.1031–1037 (1996).

[3] Maeyama, S., Yuta, S. and Harada, A.: Experiments on a Remote Appreciation Robot in an Art Museum, *Proc. 2000 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp.1008–1013 (2000).

[4] Kuzuoka, H., Kosuge, T. and Tanaka, M.: GestureCam: A video communication system for sympathetic remote collaboration, *Proc. CSCW'94*, pp.35–43 (1994).

[5] Kuzuoka, H., Oyama, S., Yamazaki, K., et al.: GestureMan: A mobile robot that embodies a remote instructor's actions, *Proc. CSCW'00*, pp.155–162 (2000).

[6] Adalgeirsson, S.O. and Brazieal, C.: Mebot, a robotic platform for socially embodied telepresence, *Proc. HRI'10* (2010).

[7] Ueki, M., Yasukawa, Y., Murase, Y., et al.: Development of the Home Robot MARON-1, *Proc. Mechatronics & Robotics Conference 2004* (2004).

[8] Yanco, H.A. and Drury, J.: Where Am I? Acquiring Situation Awareness Using a Remote Robot Platform, *Proc. SMC'04*, Vol.3, pp.2835–2840 (2004).

[9] Murakami, Y. and Nakanishi, H.: Proposing GUI for Remotely Controlling Guide Robots, IPSJ SIG Technical Reports, 2008-HCI-127, pp.79–86 (2008). (In Japanese).

[10] Shiroma, N., Sato, N., Chiu, Y., et al.: Study on Effective Camera Images for Mobile Robot Teleoperation, *Proc. 13th IEEE International Workshop on Robot and Human Interactive Communication*, pp.107–112 (2004).

[11] Sugimoto, M., Kagotani, G., Nii, H., et al.: Time Follower's Vision: A Tele-Operation Interface with Past Images, *IEEE Computer Graphics and Applications*, Vol.25, No.1, pp.54–63 (2005).

[12] Noyes, M.V. and Sheridan, T.B.: A Novel Predictor for Telemanipulation Through a Time Delay, *Proc. Annual Conference on Manual Control* (1984).

[13] Bejczy, A.K., Kim, W.S. and Venema, S.C.: The Phantom Robot: Predictive Displays for Teleoperation with Time Delay, *Proc. IEEE International Conference on Robotics and Automation*, pp.546–551 (1990).

[14] Stilman, M., Michel, P., Chestnutt, J., et al.: *Augmented reality for robot development and experimentation*, Robotics Institute, Carnegie Mellon University, Technical Report CMU-RI-TR-05-55 (2005).

[15] Arai, H.: Human Interface for Maneuvering Non-holonomic Systems, *Proc. 2001 IEEE International Conference on Robotics and Automation*, pp.1870–1877 (2001).

[16] Tsumaki, Y. and Yokohama, M.: Predictive Motion Display for Acceleration Based Teleoperation, *Proc. 2006 IEEE International Conference on Robotics and Automation*, pp.2927–2932 (2006).

[17] Menchaca-Brandan, M.A., Liu, A.M., Oman, C.M., et al.: Influence of Perspective-Taking and Mental Rotation Abilities in Space Teleoperation, *Proc. HRI'07*, pp.271–278 (2007).
[18] Sekimoto, T., Tsubouchi, T. and Yuta, S.: A Simple Driving Device for a Vehicle -Implementation and Evaluation, *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp.147–154 (1997).
[19] Skubic, M., Anderson, D., Blisard, S., et al.: Using a hand-drawn sketch to control a team of robots, *Auton. Robots*, Vol.22, No.4, pp.399–410 (2007).
[20] Sakamoto, D., Honda, K., Inami, M., et al.: Sketch and run: A stroke-based interface for home robots, *Proc. CHI'09*, pp.197–200 (2009).
[21] Sian, N.E., Yokoi, K., Kajita, S., et al.: Whole body teleoperation of a humanoid robot integrating operator's intention and robot's autonomy: An experimental verification, *Proc. 2003 IEEE/RSJ International Conference*, pp.1651–1656 (2003).
[22] Suzuki, Y., Fukushima, H., Miyamoto, I., et al.: Control interface for shooting a movie for communication supporting robot system, Interaction 2010, PB05 (2010). (In Japanese).
[23] Kato, H. and Billinghurs, M.: Marker Tracking and HMD Calibration for a Video-based Augmented Reality Conferencing System, *Proc. 2nd International Workshop on Augmented Reality*, pp.85–94 (1999).
[24] Sugiura, Y., Igarashi, T., Takahashi, H., et al.: Graphical Instruction for a Garment Folding Robot, *ACM SIGGRAPH 2009 Emerging Technologies* (2009).
[25] Zhao, S., Nakamura, K., Ishii, K., et al.: Magic cards: A paper tag interface for implicit robot control, *Proc. CHI'09*, pp.173–182 (2009).
[26] Kirk, D. and Stanton, F.D.: Comparing remote gesture technologies for supporting collaborative physical tasks, *Proc. CHI'06* (2006).
[27] Fussell, S.R., Setlock, L.D. and Parker, E.M.: Where do helpers look?: Gaze targets during collaborative physical tasks, *CHI'03 Extended Abstracts on Human Factors in Computing Systems*, pp.768–769 (2003).

Yusuke Suzuki received his M.E. degree from University of Hokkaido in 2003. He has been working at Oki Electric Industry since 2003. His research interests include Assistive Technology, Human Interface, Tele-robotics and Telecommunication. He is a member of RSJ (Robotics Society of Japan).

Motoki Terashima received his M.E. degree from University of Tsukuba in 2012. He is currently a software engineer in NTT Comware Corporation.

Koichi Takeuchi received his B.S., M.S. degrees from Keio University in 1991 and 1993 respectively. He received Ph.D. degree at The University of Tokyo in 2011. He has been working at Oki Electric Industry since 1993. He has involved in many research projects relating to design and evaluation of interactive systems (especially for people with disabilities). He is a member of HIS (Human Interface Society).

Hideaki Kuzuoka received his B.S., M.S., and Ph.D. degrees from The University of Tokyo in 1986, 1988, and 2002 respectively. He has been working at University of Tsukuba since 1992 and now he is a professor in the Faculty of Engineering, Information and Systems. His research interests include computer supported cooperative work, science education, and elderly support. He is a member of ACM (Association for Computing Machinery), IPSJ (Information Processing Society of Japan), IEICE (Institute of Electronics, Information, and Communication Engineers), RSJ (Robotics Society of Japan), VRSJ (Virtual Reality Society of Japan), HIS (Human Interface Society).