

# TouchMe: An Augmented Reality Based Remote Robot Manipulation

Sunao Hashimoto<sup>1,\*</sup>

Akihiko Ishida<sup>1,2,†</sup>

Masahiko Inami<sup>1,3,‡</sup>

Takeo Igarashi<sup>1,4,§</sup>

<sup>1</sup>JST ERATO Igarashi Design Interface Project

<sup>2</sup>Tokyo University of Science

<sup>3</sup>Keio University

<sup>4</sup>The University of Tokyo

## ABSTRACT

A general remote controlled robot is manipulated by a joystick and a gamepad. However, these methods are difficult for inexperienced users because the mapping between the user input and resulting robot motion is not always intuitive (e.g. tilt a joystick to the right to rotate the robot to the left). To solve this problem, we propose a touch-based interface for remotely controlling a robot from a third-person view, which is called “TouchMe”. This system allows the user to manipulate each part of the robot by directly touching it on a view of the world as seen by a camera looking at the robot from a third-person view. Our system provides intuitive operation, and the user can use our system with minimal user training. In this paper we describe the TouchMe interaction and its prototype implementation. We also introduce three scheduling methods for controlling the robot in response to user interaction and report on the results of empirical comparisons of these methods.

**KEYWORDS:** Remote robot control, third-person view, augmented reality, touch screen, direct manipulation.

**INDEX TERMS:** H.5.2 [Information Interfaces and Presentation]: User Interfaces — Interaction styles; I.2.9 [Artificial Intelligence]: Robotics — Operator interfaces

## 1 INTRODUCTION

There are many environments where it is hard for humans to work, such as in water, high places, high/low temperature environments and environments contaminated with poison or radioactivity. Various robots have been developed to perform tasks in these dangerous environments. Fully autonomous robot operation is desired, but it is difficult because of recognition problems. One way to alleviate recognition problem is to put tags onto objects, and use a pre-structured environment model. However, these methods are not applicable to unstructured environments and human supervisor controls are necessary.

A robot that can grab and deliver physical objects generally has a multi-DOF (degree of freedom), but it is not easy to control them for an inexperienced operator. For example, a robotic arm has generally 4 or 6 DOF. When the robotic arm is mounted on a mobile vehicle, the total DOF is increased. The most popular method for the control of multi-DOF robots is a joystick and a gamepad; however, in these control devices, the number of controllable DOF is limited by the number of buttons and axes of the devices. The controllable DOF can be increased by using

some combinations of 2 or 3 keys, but it will make operability more difficult and demand a longer training time from the user. To facilitate controlling a multiple-link robot, inverse kinematics is widely employed. Generally a robotic arm using inverse kinematics maps the control of the end-effector to a joystick, and the angles of each joint are calculated appropriately. However, in a general joystick based controller, the moving velocity of the robot is given in proportion to the timing or degree of key pressing. This also needs user training.

We propose a tele-operating system that allows the user to manipulate a multi-DOF robot intuitively with touch interaction from a third-person view, which we call “TouchMe”. The system is shown in Figure 1. This is a touch screen based interface, and it displays an image acquired from a camera observing the robot from a third-person view. The user can directly specify the desired pose and position of the robot by touching and dragging the part that he/she wants to control. The camera image showing the robot is overlaid with a computer graphics (CG) model synchronized with the user’s manipulation to help the user predict how the robot will move. This is an augmented reality application applying the direct manipulation of a posing tool of a virtual human such as Poser to the multi-DOF robot in the real-world. Typical remotely controlled robots use a robot-mounted camera providing a first-person view, but we use a third-person view camera because it allows the user to understand the situation of the entire working space (the controlled robot, target objects, and obstacles). We discuss advantages and disadvantages of various third-person view camera settings for the proposed method such as a fixed surveillance camera, a flying camera, and other robot’s eyes.

In this paper, we describe the TouchMe interaction and our prototype implementation. We also introduce three scheduling methods for the robot control in response to user interaction. One is to move the robot after the touch, one is to move the robot during the touch, and one is to move the robot during and after the touch. We compared these three methods in an empirical evaluation and report on the results.

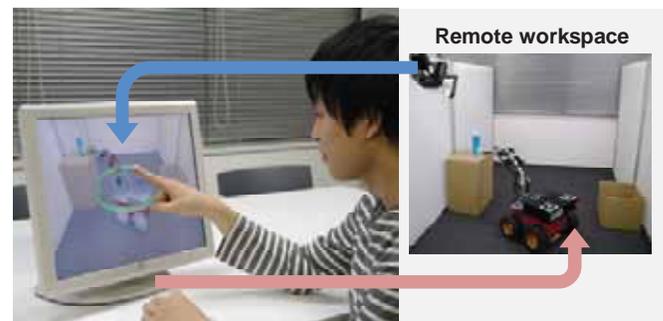


Figure 1: TouchMe. The user can directly control each movable part of the robot by touching the camera image.

\*e-mail: hashimoto@designinterface.jp

†e-mail: ishida@designinterface.jp

‡e-mail: inami@inami.info

§e-mail: takeo@acm.org

## 2 RELATED WORKS

There are various video-based interfaces for controlling a robot and appliances.

Tani et al. presented interactive video techniques that allow interaction with objects in live video on the screen, by having models of the objects monitored by cameras [1]. They explored two strategies for modeling objects imaged by cameras in 2D and 3D. They implemented a system called HyperPlant for monitoring and controlling an electric power plant, by using 2D modeling. Seifried et al. developed a video-based interface for controlling home appliances in the project CRISTAL [2]. In their work, the camera is mounted for a top-down view of the living room, and the image is displayed on a multi-touch tabletop surface. This system allows multiple users to operate multiple appliances collaboratively. We also control a device through a camera image, but our controlled object is a multi-DOF robot, and we aim to achieve more complicated tasks using it.

Top-down view has been used in several video-based robot control interfaces. Sakamoto et al. proposed a video-based Tablet PC interface to control vacuum cleaning robots [3]. In this system, ceiling mounted cameras provide the user a top-down view that allows the user to control robots and design their behaviors by sketching using a stylus pen. Kato et al. developed a multi-touch tabletop interface for controlling multiple robots [4]. They proposed a method to control multiple mobile robots simultaneously by manipulating a vector field on a top-down view from a ceiling camera. Guo et al. presented two interfaces for remotely interacting with multiple robots using toys on a large tabletop display showing a top-down view of the workspace [5]. This research shows the fact that a top-down view is easy for controlling the locomotion of mobile robots on a 2D surface, however it is hard to control multi-DOF robots.

There are several interfaces for controlling a robot through a first-person view (robot's-eye view). Sekimoto et al. proposed a simple driving interface for a mobile robot using a touch panel and first-person view images from the robot [6]. Once the operator gives a point of the temporary goal position by touching on the monitor displaying the front view of the robot, the system generates a path to the goal position and the vehicle is controlled to follow the path to reach the goal position autonomously. Fong et al. also developed a similar system on a handheld device (PDA) [7]. Correa et al. proposed a handheld tablet interface for operating an autonomous forklift, where users provide high-level directives to the forklift through a combination of spoken utterances and sketched gestures on the robot's-eye view displayed on the interface [8].

Third-person view is also used in video-based remote robot control systems. Hosoi et al. proposed a robot control technique using a camera-mounted mobile device such as a PDA and a mobile phone, which is called Shepherd [9]. In this system, the operator holds a camera-mounted mobile device in his/her hand, and he/she instructs the robot how to move by moving the device. Sugimoto et al. proposed a visual presentation system for controlling a robotic vehicle remotely, which is called Time Follower's Vision [10]. This allows the operator to control a remote rescue robot by observing a virtual third-person view which is created from a first-person view camera mounted on the robot. They show the effectiveness of a third-person view to allow even inexperienced operators to easily control the robot.

There are several robot interfaces using augmented reality and mixed reality techniques. Nawab et al. proposed a method that overlays a color-coded coordinate system on the end-effector of the robot using augmented reality to help the user to understand the key mapping of a joystick [11]. Kobayashi et al. developed a

mixed reality environment which can overlay internal statuses of a humanoid robot such as recognition results and planning results [12]. Their method enables the operator to understand the robot internal statuses intuitively, which is helpful for debugging and actual operation. Chen et al. also developed a mixed reality environment for performing robot simulations involving physical and virtual objects [13]. Drascic et al. developed an augmented reality through graphic overlaying on a stereo video [14]. In their application, the user wearing a data glove controls a robotic arm by manipulating a virtual cursor overlaid on the video image. Xiong et al. also developed a tele-robotic system based on augmented reality to control a six DOF robotic arm [15]. In this system, a virtual robot works as an interface between the operator and the real robot, mitigating the problem of time-delay between user operation and real robot action. This idea is also used in our research, but we use a touch screen for the interface and we empirically compare three touch interaction methods, while they used a head-mounted display, a data glove, and voice commands for their interface.

## 3 USER INTERACTION

TouchMe is an augmented reality interaction technique for remote robot control. The system overview is shown in Figure 2. The camera captures the image of the workspace in real-time, and the image is shown on the touch screen with a CG model of the real robot. The CG model is overlaid on the robot, and it is shown semi-transparently (like a ghost). The user controls the robot by touching the overlaid CG model. The user touches the part of the CG model where he/she wants to move, and he/she then drags it to the desired position and direction. For example, the user slides the body to move the robot to a specific position, and then drags the top of the arm to reach an object. This is similar to manipulations performed in 3D modeling and posing software, in which a 3D object is manipulated using user operations on a 2D image plane. As the user moves the CG model, it eventually moves away from the physical robot on the screen, and the system drives the robot so that it matches with the CG model.

### 3.1 Third-person view

We use a third-person view camera because it allows the user to understand the situation of the entire work space composed of the controlled robot, target objects and obstacles. A typical approach is to use a first-person view image obtained from the robot-mounted camera, but we did not use this because it is difficult to avoid collisions with obstacles on the side or behind the robot when the robot is rotating or moving backwards. We will now discuss various third-person view camera. We only implemented and tested the first method. The implementation of the remaining two methods is our future work.

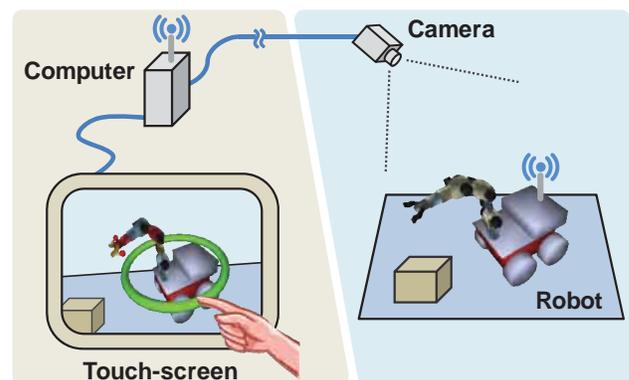


Figure 2: Overview of the system.

**Fixed surveillance camera:** Fixed surveillance cameras are already installed in various places such as roads, parks, stations, museums, factories, stores and homes, for security and recording. A surveillance camera is mounted in a high position that is higher than human height and provides a bird's-eye view. The advantage of a surveillance camera is that it gives the user a good stable view for understanding the entire surrounding environment. However, the movement of the fixed camera is limited to panning, tilting and zooming, making it difficult to resolve possible occlusion.

**Flying camera:** Various unmanned aerial vehicles (UAV) such as a remote-controlled helicopter and an airship with a camera are used for scouting. A UAV's camera also provides a bird's-eye view, and it can move freely unlike a fixed camera. A flying camera can use viewpoint operation (such as the CG modeling software) in the real-world. Moreover, the camera can track the target robot automatically to keep the robot in the field of view. This gives the operator a view like a 3D action game where a third-person view camera follows after the game character such as in Nintendo's Super Mario 64. The disadvantage of this camera is that it demands very stable and highly precise control for having such free viewpoint movement.

**Another robot's camera:** When two or more robots are employed in a workspace and one of them has an eye (a first-person view camera), we can operate the other robots in a third-person view by borrowing the view of the robot. If all robots have cameras, the user can operate the robots while switching first-person views and third-person views freely. For example, a first-person view is used when the user operates the hands of the target robot, and a third-person view is used when the user wants to move the target robot to another position avoiding obstacles.

The viewpoint operation can be performed by touching on the region outside of the robot or touching a special icon for manipulating the view point. We guess that automatic camera control would be useful. For example, when the user manipulates the end-effector of a robotic arm, the camera moves to the position where it can give good the operator a good view, and is zoomed in automatically.

### 3.2 Virtual handles

Virtual handles are user interfaces to make the CG model easy to manipulate. It is useful for understanding the controllable direction of the mounted part. Figure 3 shows two types of virtual handles. The ring type is used for manipulating a rotating part (e.g. rotation of the body, rotation of a link of the arm). The lever type is used for manipulating a small part such as an end-effector. These ideas are used widely in CG modeling software [16]. We apply them to the real robot by using an augmented reality technique.



Figure 3: Virtual handles. Ring type for rotation of the vehicle, and lever type for manipulating end-effector of a robotic arm.

### 3.3 Inverse kinematics

We use inverse kinematics (IK) to facilitate controlling a multi-joint robot. When one of the links is manipulated, related links are moved automatically. For example, when the user pulls the wrist of the arm, the elbow and the shoulder (or the body) are controlled

by the system automatically. We manipulate 3D multi-joint structures in 3D space on a 2D display surface, and we use an IK method that is used in general posing tools for virtual human models such as Poser.

### 3.4 Scheduling of robot motion

The robot only moves with a limited speed, so the CG model and the robot on the screen do not always match during the user interaction. The system resolves this mismatch by moving the robot towards the CG model, but there are multiple ways to achieve this. Here we introduce three possible scheduling methods.

**Move-after-touch:** The robot does not move while the finger is touching the screen and is manipulating the CG model. When the user releases their finger, the CG model is fixed and the robot begins to move toward the CG model. The robot stops when the pose matches with the CG model (Figure 4).

**Move-during-touch:** The robot begins to move toward the CG model immediately after the finger begins to manipulate the CG model by touching the screen. While the finger touches the screen, the pose and position of the CG model is continuously updated, and the robot continuously tracks the CG model. When the finger is released, the robot stops immediately and the CG model pose is set to the robot pose at the time of release (Figure 5).

**Move-during-and-after-touch:** This is a combination of the above two methods. The robot begins to move toward the CG model immediately after the finger begins to manipulate the CG model by touching the screen and continues moving during user manipulation. When the user releases their finger, the CG model is fixed to the pose at the time of release and the robot continues moving toward the fixed CG model. The robot stops when the pose and position of the robot matches with those of the CG model (Figure 6).

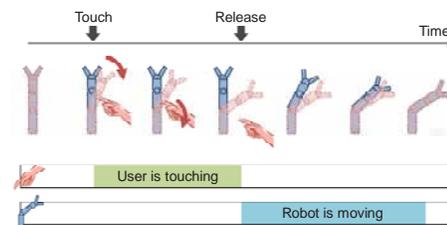


Figure 4: Move-after-touch.

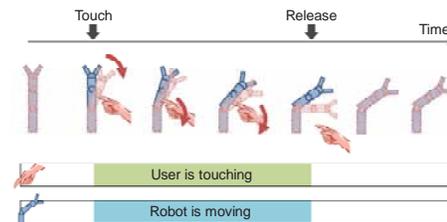


Figure 5: Move-during-touch.

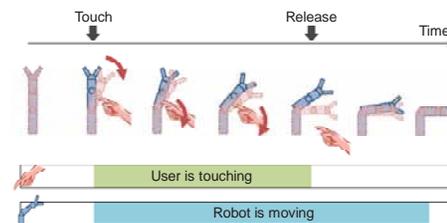


Figure 6: Move-during-and-after-touch.

## 4 PROTOTYPE SYSTEM

We developed a prototype system in which the user controls a robot vehicle using our proposed interface.

### 4.1 Robot

We used a robotic vehicle (MobileRobots PIONEER3-DX) equipped with a robotic arm (Neuronics Katana). Figure 7 shows our robot and its DOF. The vehicle has a mechanism for locomotion using four wheels. It allows the user to rotate and to move forward or backward (2DOF). The mounted robotic arm has 6DOF but we limited controllable parts to the hand (1DOF) and the three joints (3DOF) to simplify the operation. Therefore the whole robot has 6 DOF in total.

We made the CG model of this robot, and gave two kinds of virtual handles to facilitate control of the robot; ring type for the rotation of the vehicle and lever type for manipulating the hand of the robotic arm. When the user manipulates the arm, all or part of the three joint angles (joint 3, 4 and 5, shown in Figure 7) are updated according to the result of IK computation.

The vehicle and the mounted arm are controlled remotely by a host computer. The host computer controls the joint angles of the arm individually, and obtains each angle's value. In our prototype, the host computer communicates with the robot via USB wired connection. To help the user grab an object with the arm, we mounted a green flashlight on the wrist of the robotic arm to light the target when it is in front of the hand. This is important because depth information is missing in the single camera view.

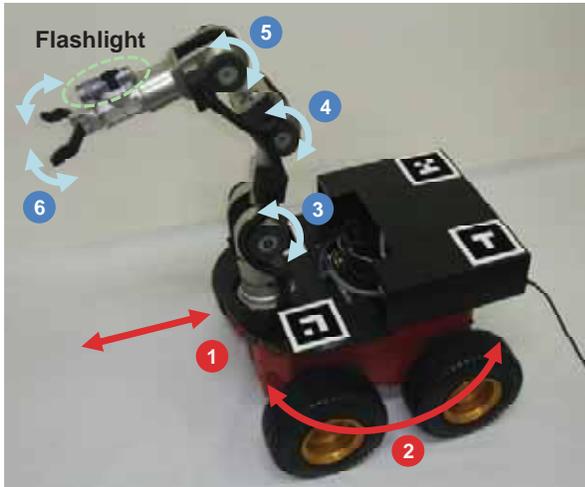


Figure 7: DOF of the robot used in the prototype.

### 4.2 Camera view and registration

We use a commercial webcam (Logicool QCAM-200V) as the third-person view camera fixed in the workspace. The camera image is displayed on a 19 inch LCD desktop touch screen. The resolution of the image is  $800 \times 600$  pixels, and the frame rate is 15fps. The camera does not support any physical movements such as panning and zooming.

We use fiducial markers (ARToolKit [17]) for registration between the real robot and virtual robot (CG). We put four markers ( $10 \times 10 \text{ cm}^2$ ) on the top of the vehicle. At the initial state and when the robot stops, the system gives the CG model the actual joint angles obtained from the robotic arm, and physical position and direction obtained from the fiducial markers. The markers are also used for visual feedback when the robot moves to the specified goal.

## 5 EMPIRICAL COMPARISON OF THE SCHEDULING METHODS

We ran a user study using our prototype system to test general usability of the system and to compare the three scheduling methods. Figure 8 shows the experimental workspace displayed on the touch screen with robot and overlaid CG model. A  $190 \times 250 \text{ cm}^2$  workspace is divided by partition walls. In this space, a blue labeled plastic bottle (with diameter of 7 cm, 25 cm high) is placed on a rack with a height of 58 cm, and a trash-box ( $42 \times 33 \text{ cm}^3$ ) is placed on the opposite side. The camera is fixed at 123 cm high from the floor.

We recruited 12 people aged 20-25 years old, 8 males and 4 females, to participate in our study. All of them are students from a university, and they use a computer in their daily lives. Most of them had no experience with robot control, and they were not familiar with our robot. The sessions lasted about an hour.

We gave each participant the task of controlling the robot to pick up a blue bottle and drop it in the trash-box using our touch screen interface. The hand angle is limited to a pre-defined angle to prevent breakdown of the hand when it grabs the bottle.

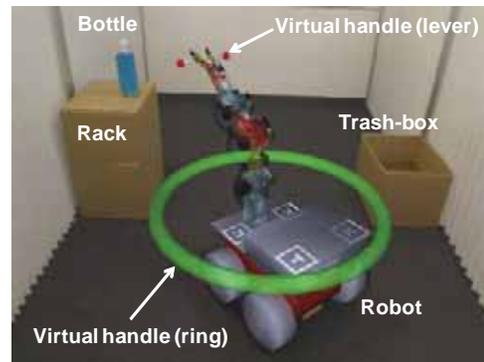


Figure 8: The superimposed image displayed on the touch screen.

### 5.1 Conditions

We conducted our user test for three conditions, move-after-touch (A), move-during-touch (D), and move-during-and-after-touch (DA).

We did not allow the participants to enter or see the workspace directly, therefore the workspace was a completely unknown environment for them. They tried to operate the robot by only observing the camera image. When the user test began, we explained to the participant how to control the robot, the DOF of the robot, and the fact that the flashlight was mounted on the wrist of the arm for aiming. The comparison was performed as within-subjects, where each participant tested all conditions. Each participant performed a task on three conditions in balanced order. For each condition, we gave the participant a training time of up to five minutes before the trial. All objects and the robot were placed in their initial positions for each trial. If the robot dropped the bottle on the floor due to an operator's mistake, we recorded the trial as a failure. If it was caused by a system error, we gave the participants a chance to retry. For each trial, we recorded the task completion time and asked the participant to answer a questionnaire. After three trials we interviewed them.

### 5.2 Results

All 12 participants except one person succeeded in the task. The one who failed dropped the bottle in all trials. Table 1 lists the time to complete the task for three conditions (only successful cases). The tasks were finished in approximately two minutes. The results indicate no significant differences between conditions (by ANOVA,  $p=0.77$ ).

The results from the questionnaires (seven-point Likert scale with high scores positive) are shown in Figure 9, and we show the detailed questionnaires in Table 2. The only negative question is Q4. After ANOVA, Ryan's method was performed for the results. Statistically significant results ( $p < 0.05$ ) are shown in Q2, Q5 and Q6. Condition A got the highest positive value of all the questions, and there are no significant differences between D and DA through all the results. The result of Q2 shows that the participants controlled the robot with stronger confidence in A than D ( $p = 0.01$ ). The result of Q5 shows that most participants expected that many people can control the robot easily by using A, stronger than D ( $p = 0.01$ ). The result of Q6 shows that the participants could control the robot as they expected by using A, stronger than D ( $p = 0.001$ ).

Table 1: Average time to complete tasks.

Condition	Time (m:ss)	St.Dev. (m:ss)
A	1:52	0:30
D	2:00	0:22
DA	1:53	0:22

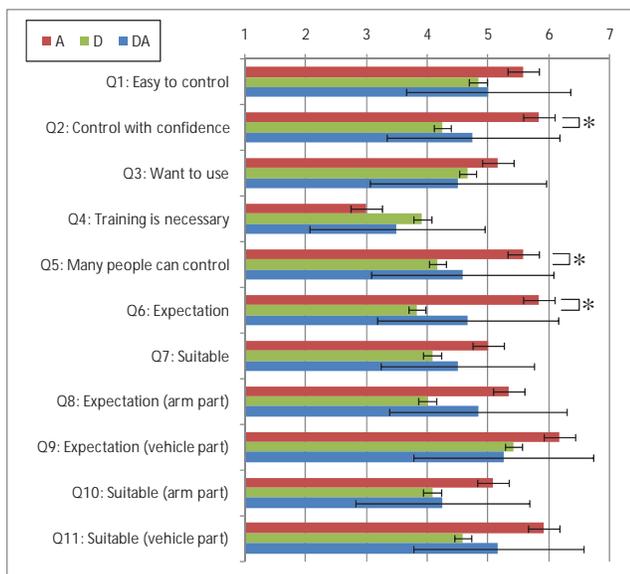


Figure 9: Comparison of results.

Table 2: Questionnaires for the conditions.

Q1	This method is easy to control the robot.
Q2	I controlled the robot with confidence.
Q3	I want to control the robot using this method.
Q4	A lot of training is necessary for using this method.
Q5	Many people can control the robot easily using this method.
Q6	I could control the robot as I expected.
Q7	This method is suitable for controlling the robot.
Q8	I could control the arm part as I expected.
Q9	I could control the part of vehicle as I expected.
Q10	This method is suitable for controlling the arm part.
Q11	This method is suitable for controlling the vehicle part.

### 5.3 Discussion

The comparison results show that A tends to be supported most among the three conditions, in particular, in Q2, Q5 and Q6. We found no significant differences about the easiness (Q1). However, in the interview, 9 of 12 people answered that A is the easiest, and the main reason was that they could operate calmly without being rushed. On the other hand, 9 of 12 people answered that A might take the longest time to complete a task among the three conditions in the interview, despite the fact that the task completion time showed no significant difference between conditions in the trials. They mentioned "D and DA are faster than A because these track manipulated CG model immediately". The condition A seems that it gives the impression that it is not efficient mentally because the robot moves after the finger is released from the touch screen.

In Q4 (Training is necessary), there are no significant differences; however, all conditions were less than four. In addition, 11 of 12 participants completed the tasks with training taking less than five minutes. These results show that the proposed system needs less time for training than for using.

We found no significant differences in Q3 (want to use) and Q7 (suitable). In the interview, we asked them about these criteria in detail. Several people answered that they wanted to use A in an actual situation and that A is the most suitable method. The main reason was that A was the easiest to use for them. People who preferred D or DA found great value in the physical feedback that the real robot moves immediately according to the user's operation. Particularly, people who preferred D liked a property that allowed the user to stop the robot when he/she wants to stop it by releasing his/her finger from the screen. We also got several negative comments for D as follows: "D makes me tired (or it is troublesome) because I should keep touching until the robot arrives to the destination", "Sometimes I have released my finger by mistake".

### 5.4 General observations

All participants understood how to use the virtual handles, and they used them efficiently when they rotated the robot and when they manipulated the hand of the arm. Participants commented about manipulation of the CG model as follows: "I want to give an inertia to CG model because it might make the operation more light", "It was hard to recognize that the hand is grabbing the target object or not, therefore some visualization is necessary, e.g. hand's color is changed when target object is grabbed".

For the IK based arm control, several people said that it was not clear how the arm's pose transforms when they manipulated the end of the linked parts. This fact shows that it is necessary to visualize the moveable range of the links of the arm. They also wanted to use both IK and FK (forward kinematics) for controlling the robotic arm.

The third-person view camera was well received by all participants through the experiments. In particular it was fully appreciated by the people who have experience playing 3D action games using a third-person view camera. The biggest negative comment about the third-person view is that it is hard to understand the depth of the space from the image obtained from the single fixed camera. The moveable part would be uncontrollable when the view direction was orthogonal to the rotation axis of the moveable part, or when the part was occluded by another part. We believe that these problems can be solved by using a moveable camera or multiple cameras. The participants requested 1 or 2 additional viewpoints such as a top-down view, a side view or a view on the wrist for our future development. Zooming was also requested to observe the working area more

closely. One of the participants requested a moving camera that follows the robot from behind such as those seen in a third person shooter game.

Several people requested a stylus pen. The main reasons are that it might allow the user to have more precise manipulation and that the display area hidden by a pen is smaller than that of a finger. A multi-touch screen was also requested for the pointing device, with the requested gestures being a pinching gesture for zooming in and out of the view, a pinching gesture for open-close manipulation of the robotic hand, and a two fingers gesture where one finger rotates a link of the arm while another finger holds the anchor point of the joint. Two-finger interaction might be a good method for switching between IK and FK for controlling a robotic arm.

## 6 LIMITATIONS

We will now discuss the current limitations of this work. The proposed method needs a third-person view camera, and the moveable area of the controlled robot is limited to the field of view of this camera. The camera needs to keep a certain amount of distance from the controlled robot to give the image of the robot including the controlled part. The possible position where the camera is put is limited physically in real-world environments, by obstacles and limited small spaces. As a result, it might cause bad views in which it is hard to operate the robot. The resolution of operation depends on the display resolution, and the amount of operation given by a pixel is different if the controlled part is near or far from the camera. To control a CG model by touching, the controlled part needs a certain amount of surface area, and the display also needs a certain amount of physical area to accommodate the CG model.

## 7 CONCLUSION AND FUTURE WORK

In this paper, we presented the design, implementation and an initial evaluation of an augmented reality interface for controlling a multi-DOF robot. TouchMe allows the user to manipulate each part of the robot by directly touching it on a view of the world as seen by a third-person view camera. We compared three scheduling methods on our first prototype system. Most participants found that the easiest method was when the robot began to move after the participant's finger was released from the touch screen.

The results of the user study provided further design recommendations for future iterations of TouchMe and for similar robot control systems. The virtual handles were well received by all participants. The users requested richer visualization for understanding the state of the robot. We found that both IK and FK are desirable for controlling a robotic arm. The third-person view was well received by all participants in the user study, though they also claimed that the third-person view caused the problems such as occlusions and rotation axis aligning with the camera view. Several people requested a stylus pen for more precise manipulation, and also requested a multi-touch screen for advanced manipulation.

Our immediate work in the future is to solve the viewpoint problem that causes occlusions and uncontrollable situations in the third-person view. We expect that this problem can be solved by employing a moveable camera or multiple cameras. Introducing a multi-touch screen is also our future work. We expect that it will allow multiple users to manipulate multiple robots on a screen collaboratively. Moreover, in the future, a model-based tracking would be introduced to relate the virtual

robot to the real robot, instead of fiducial markers used in the current prototype.

Our proposed method has a flexible scalability. We can apply this method for various kinds of robots such as humanoids, tabletop robots, bulldozers, power shovels and cars. We expect that extremely small robots can be controlled by touching on a microscope image, and a very large robot could also be controlled by viewing from a distance. We plan to extend our implementation and explore the applicability for various platforms.

## REFERENCES

- [1] M. Tani, K. Yamaashi, K. Tanikoshi, M. Futakawa and S. Tanifuji, Object-oriented video: interaction with real-world objects through live video, In *Proceedings of the CHI'92*, pp.593-598, 1992.
- [2] T. Seifried, M. Haller, S. D. Scott, F. Perteneder, C. Rendl, D. Sakamoto and M. Inami, CRISTAL: A Collaborative Home Media and Device Controller Based on a Multi-touch Display, In *Proceedings of the Tabletop'09*, pp.33-40, 2009.
- [3] D. Sakamoto, K. Honda, M. Inami and T. Igarashi, Sketch and Run: A Stroke-based Interface for Home Robots, In *Proceedings of the CHI'09*, pp.197-200, 2009.
- [4] J. Kato, D. Sakamoto, M. Inami and T. Igarashi, Multi-touch Interface for Controlling Multiple Mobile Robots, In *Proceedings of the CHI'09*, pp.3443-3448, 2009.
- [5] C. Guo, J. E. Young and E. Sharlin, Touch and toys: new techniques for interaction with a remote group of robots, In *Proceedings of the CHI'09*, pp.491-500, 2009.
- [6] T. Sekimoto, T. Tsubouchi and S. Yuta, A Simple Driving Device for a Vehicle Implementation and Evaluation, In *proceedings of the IROS'97*, pp.147-154, 1997.
- [7] T. Fong, C. Thorpe and B. Glass, PdaDriver: A Handheld system for Remote Driving, In *Proceedings of the ICAR'03*, 2003.
- [8] A. Correa, M. R. Walter, L. Fletcher, J. Glass, S. Teller and R. Davis, Multimodal Interaction with an Autonomous Forklift, In *Proceedings of the HRI2010*, 2010.
- [9] K. Hosoi and M. Sugimoto, Shepherd: A Mobile Interface for Robot Control from a User's Viewpoint, In *Proceedings of the ROBIO'06*, pp.908-913, 2006.
- [10] M. Sugimoto, G. Kagotani, H. Nii, N. Shiroma, M. Inami and F. Matsuno, Time follower's vision, In *Proceedings of the SIGGRAPH'04*, p.29, 2004.
- [11] A. Nawab, K. Chintamani, D. Ellis, G. Auner and A. Pandya, Joystick mapped Augmented Reality Cues for End-Effector controlled Tele-operated Robots, In *Proceedings of the IEEE Virtual Reality'07*, pp.263-266, 2007.
- [12] K. Kobayashi, K. Nishiwaki, S. Uchiyama, H. Yamamoto, S. Kagami and T. Kanade, Overlay what Humanoid Robot Perceives and Thinks to the Real-world by Mixed Reality System, In *Proceedings of the ISMAR'07*, pp.1-2, 2007.
- [13] I. Y. H. Chen, B. MacDonald and B. Wünsche, Mixed reality simulation for mobile robots, In *Proceedings of the ICRA'09*, pp.922-927, 2009.
- [14] D. Drascic, J. J. Grodski, P. Milgram, K. Ruffo, P. Wong and S. Zhai, ARGOS: A Display System for Augmenting Reality, In *Proceedings of the INTERACT'93*, p.521, 1993.
- [15] Y. Xiong, S. Li and M. Xie, Predictive display and interaction of telerobots based on augmented reality, *Robotica(2006)*, 24, Cambridge University Press, pp.447-453, 2006.
- [16] B. D. Conner, S. S. Snibbe, K. P. Herndon, D. C. Robbins, R. C. Zeleznik and A. Van Dam, Three-dimensional widgets, In *Proceedings of the SI3D'92*, pp.183-188, 1992.
- [17] H. Kato and M. Billinghurst, Marker Tracking and HMD Calibration for a Video-based Augmented Reality Conferencing System, In *Proceedings of the IWAR'99*, pp.85-94, 1999.