ABSTRACT
“MADO Interface” is a tangible user interface consisting of a compact touch-screen display and physical blocks. “MADO” means “window” in Japanese, and MADO Interface is utilized as the real window into the virtual world. Users construct a physical object by simply combining electrical blocks. Then, by connecting MADO Interface to the physical object, they can watch the virtual model corresponding to the physical block configuration (shape, color, etc.) The size and the viewpoint of the virtual model seen by the user depend on the position of MADO Interface, maintaining the consistency between the physical and virtual worlds. In addition, users can interact with the virtual model by touching the display on MADO Interface. These features enable users to explore the virtual world intuitively and powerfully.

Keywords
MADO Interface, tangible user interface, mixed reality, 3D modeling, real-time interaction, bi-directional interface

INTRODUCTION
Incorporating direct manipulation of physical objects into a computer interface can improve ease of learning, use and retain over time. Recently, based on this idea, research in the area of physically based user interfaces such as tangible user interfaces (TUIs) has become active, with the goal of improving the intuitiveness of 3D object modeling and interactive manipulation of 3D scenes. Such TUIs aim to support the manipulation of objects in the virtual world through manipulation of corresponding physical object. This enables users to interact with computers without being conscious of a boundary between the physical and virtual worlds.

Such systems, however, do not generally include a display showing the virtual world. Instead, the virtual world is usually shown on a separate display connected to the host PC. In this situation the relationship between the physical and virtual worlds is maintained, but there remains a problem that they are not in the same place spatially and the operational object and the place to confirm the operational results are separated. This could potentially confuse users and it does not support intuitive interaction with the computer.

On the other hand, “Mixed Reality” has become an active area of researches. These researches are focusing on the development of compound environments unifying the physical and virtual worlds. Mixed reality systems present the virtual environment through the use of head-mounted displays or by projecting directly on physical objects. This approach provides a new world where physical and virtual objects co-exist and interact in real time. However, such systems are limited by the need to utilize devices such as projectors or video cameras, and there is also the limitation of users’ interactions since the virtual world can be occluded with users’ hands during interaction with the physical objects. Moreover, the virtual elements in mixed reality systems do not take physical form, so there is no means of supporting natural physical interaction.

In this paper we present MADO Interface, which extends existing research on TUIs and mixed reality. MADO Interface leverages the advantages of TUIs while simultaneously providing superior intuitive operability. The system makes use of the “ActiveCube” system as the operational physical object, and MADO Interface serves to unify the physical and virtual worlds by providing a contextual window into the virtual world represented by the block shape. Users can connect MADO Interface to any part of the 3D shape structure they construct, and it provides a view into the corresponding virtual model from customizable viewpoints. In addition, users can interact with the virtual model to edit in detail by touching MADO Interface.

RELATED WORK
Physically based user interfaces such as TUIs have been proposed for making digital information accessible through the manipulation of physical objects [4,5,6,7,8,11].
“Tangible Bits” [5] presents a vision of Human Computer Interaction using the TUI. It allows users to grasp and manipulate digital information (bits) by coupling them with everyday physical objects and environments. The goal of Tangible Bits is to bridge the gaps between cyberspace and the physical environment.

Several research groups have investigated TUIs for 3D object shape modeling. This approach allows users to work in the physical world, where they can leverage their natural abilities, as opposed to being limited to virtual 3D environments shown on 2D displays or the non-tangible elements of mixed-reality systems. “CUBIK” [7] is a bi-directional tangible user interface consisting of a wire frame cube. Users can change the 3D virtual model configuration by manipulating the physical cube. “CubeExplorer” [11] is also a cube interface providing a hybrid 3D conceptual aid that combines physical interaction and digital modeling. This system lets users perform subtractive 3D geometric operations on a simple paper based cube model. Users can also make a subtractive 3D model in the virtual world by using a digital pen to specify the section to remove on the physical cube.

All of the TUI systems so far described output the 3D virtual model to the display of the host PC. Thus, the operational object and the created model are separated spatially, resulting in the need for a user to coordinate interaction with the two different representations of the same conceptual object.

On the other hand, some researchers have attempted to realize the fusion of the physical and virtual worlds [1,2,3,9,10,12,13]. Several approaches, including mixed reality, have been proposed which allow users to recognize and physically interact with the virtual object created in the virtual world. An example of such a system is “MagicBook,” [2] an illustrated book with content enhanced with a 3D virtual model shown using the aid of special glasses. The system recognizes a reader’s location in a book using an image marker unique to each page, and shows the 3D model allocated to the page on the glasses. Another system, “Boom Chameleon,” [12] is a novel input and output device consisting of a flat-panel display mounted on a tracked mechanical armature. The display is utilized as a physical window, and it provides a one-to-one mapping between the physical and virtual environments. The system recognizes the bent angle of armature and calculates the position of the display, so that the fusion of both environments is achieved. Users can look into the virtual environment from any direction by moving the display in the movable range of the device. Both of these approaches, however, are limited in that the physical object does not actually exist, and it is not necessarily intuitive for a user to understand the spatial position of the 3D model.

ACTIVECUBE SYSTEM

MADO Interface is realized by using the ActiveCube system. ActiveCube (shown in Figure 1) is comprised of physical blocks (5 centimeters on a side) that can be connected with one another. Each block contains a microprocessor, allowing it to be controlled in real time. The blocks create a network dynamically as they are connected and communicate with each other over the network. The host PC then communicates with the blocks and recognizes the block configuration by analyzing the data sent from them: cube ID, cube face ID, etc. The host PC manages the connection status information in a tree structure, so that the host PC can correctly recognize the 3D structure in real time.

A variety of ActiveCube blocks for special-purpose have been developed. One kind of block is equipped with a motion sensor (NEC TOKIN MDPA3U9S) that allows the host PC to recognize orientation of the block structure. With this block, the 3D virtual structure shown on the computer can be synchronized with the physical blocks as the user manipulates them. In addition, ActiveCube has various input and output devices that can reflect user’s operational intention and other information from the real world.

A number of input-specific ActiveCube blocks have been developed. These include an ultrasonic sensor, a tactile sensor, a button, etc. There are also output-specific blocks, including a propeller, buzzer, vibrator, etc. The host PC can also control these devices via the network, and users can create block structures with various customized input and output functions by simply attaching or detaching these blocks. In this way, the ActiveCube system allows users to construct and interact with 3D environments using physical cubes as a bi-directional user interface.

Using the ActiveCube system, we have developed various applications for various fields, including: a retrieval system
for 3D shape models [4], an intellectual toy for children [6], a 3D modeling system, and a medical system. As an interface to the ActiveCube system we have developed a C++ class library. This library makes it possible for an application to recognize the structure of connected blocks and to control the input and output devices with which each block is equipped.

**MADO INTERFACE**

**Overview**

Figure 2 shows the “MADO” (window) device which allows users to observe the virtual world. This device is realized by utilizing Apple’s iPod Touch which consists of a high-resolution LCD and a multi-touch input. Figure 3 shows the concept of the MADO Interface system. The system always maintains consistency between the physical and virtual worlds. MADO Interface is connected to the blocks and its display shows images from the virtual world. As a result, users discover that the operational object and the corresponding virtual object both exist in the same place spatially. Therefore, users are able to recognize and intuitively interact with the 3D model displayed on MADO Interface.

**System configuration**

The MADO Interface system makes use of a web-server based approach, where any device with a web browser is capable of displaying the virtual world. In response to users’ manipulations of either ActiveCube or the attached MADO Interface, an image of the virtual world is generated in the host PC and transmitted to MADO Interface by HTTP, where it is shown in a web browser window. Interactions with the ActiveCube blocks are dependent on the capabilities of the cubes which have been connected, whereas interaction with MADO Interface is performed through direct touch. We make use of iPod Touch’s native capabilities to accept multi-touch input and display output through the Safari web browser.

Figure 4 shows the configuration of system components and indicates the data flow. The system consists of three modules; ActiveCube Controller, Image Generator, and TCP Server. Next, we describe the roles of these modules.

**ActiveCube Controller**

ActiveCube Controller controls the ActiveCube-related data. It was developed using the ActiveCube Library, and it can recognize the connection and disconnection of blocks, maintains a model of the 3D shape structure, and also recognizes the orientation of ActiveCube, using information from a motion sensor. Furthermore, it calculates the viewpoint in the virtual world by using the connected position of MADO Interface. After obtaining this information, ActiveCube Controller transmits it to Image Generator to create the image of the virtual object.

**Image Generator**

Image Generator generates images of the virtual world. Based on the information sent from ActiveCube Controller, this module determines the virtual model and generates a small (240x320 pixels) JPEG image which contains small data enough to avoid bandwidth issues in image transmission. Then it conveys the path name of the image file to TCP Server. This module also serves to interpret the touch data on the MADO device sent from the TCP Server. Firstly it judges whether a user touched the virtual model or a button to select the type of manipulation. If the user touched the virtual model, this module transforms 2D
position data on the physical display into 3D position data on the surface of the 3D virtual model. The result of touch interaction is reflected in the appearance of the virtual object.

TCP Server
The TCP Server provides communications with MADO Interface via a web browser. This module is a generic server that streams images of the virtual world to any device, including MADO Interface. In this system, MADO Interface accesses the server as a client and requests the transmission of the image updated at a regular interval of 500 msec. In response to the request, the TCP Server transmits the path name of the image generated by Image Generator to MADO Interface. A shorter interval is not appropriate as it can result in communication collision between the server and MADO Interface. On the other hand it is desirable to have interactive refresh rates, therefore we have chosen the given 500 msec interval. In this system, the state of the virtual world is dynamically changed by the manipulation of ActiveCube and MADO Interface. Accordingly, the HTML web application employs an Ajax approach that enables asynchronous communication, allowing the system to update the picture dynamically without reloading the entire page. Consequently, users can look into the virtual world on the screen of MADO Interface in real time. In addition, they can input position data on the display by touching it since the data is recognized by the web browser and transmitted to TCP Server.

These three modules run as their own processes. This allows for synchronous execution, resulting in improved performance.

Expansion of the viewpoint with “JointBlock”
In this section, we describe a detailed method for calculating the viewpoint of MADO Interface in the virtual world. Using ActiveCube as the TUI, the system can easily determine the viewpoint of the virtual world.

As described before, the microprocessor in each block of ActiveCube can communicate with other blocks. In addition, the system can easily recognize the connection/disconnection and 3D shape structure of ActiveCube by receiving this data from the blocks in real time. As shown in Figure 2, since the MADO device contains the same circuit that is equipped in the other ActiveCube blocks, the system can also detect the existence of MADO Interface and its 3D position relative to the entire block structure. Then the system calculates the viewpoint of MADO Interface in the virtual world based on the position data of the 3D constructed structure and MADO Interface.

Unfortunately, once users connect MADO Interface to the 3D constructed structure, its position relative to the blocks is fixed due to the mechanical connection. We addressed this limitation by creating new blocks named “JointBlock.” There are three kinds of JointBlock; expansion/contraction, tilt, and rotation (shown in Figure 5). These blocks offer users to flexibly adjust the viewpoint of MADO Interface. Each JointBlock is equipped with a potentiometer, and its resistance value is used to calculate the length or angle of the block.

The expansion/contraction block (shown in Figure 5(a)) can measure the length on one axis, and the tilt block can measure the amounts of tilt on two axes (shown in Figure 5(b)), whereas the rotation block can measure the amount of rotation of one axis (shown in Figure 5(c)). Figure 6 shows the connection between MADO Interface and ActiveCube using a JointBlock (Tilt type). The connection among the blocks is made by connecting magnets on each block face which connect to corresponding magnets on the opposite face, meaning the direction of the connection is uniquely decided. In this way, the system can recognize how users move JointBlock and calculate the viewpoint in the virtual world.
**Display of the 3D virtual model**

The described system allows users to construct a 3D shape structure simply by building with physical blocks. A virtual 3D shape model similar to the constructed structure is retrieved from a database by the shape retrieval system [4].

In this system, we set that the most closely matched 3D model is shown on the display of MADO Interface when it is connected to the blocks. The retrieval of the 3D model is performed in ActiveCube Controller since it is calculated based on the configuration of ActiveCube blocks. Once a virtual model has been identified, ActiveCube Controller notifies Image Generator of the result of the retrieval. If there are several models which are close matches to the physical block structure, they are all transmitted to Image Generator and it gives the user the choice of which virtual model to display. Thus if users are not pleased with the provided virtual model, they can freely change it into other model whose shape is similar to the physical structure by operating MADO Interface. Figure 7 shows the virtual model of an airplane and users can look into any part of the virtual model as they like by attaching MADO Interface appropriately. Figure 7(a) shows several different parts of the airplane made visible by connecting MADO Interface to the blocks, and Figure 7(b) shows a dynamic presentation of the 3D model from various viewpoints with MADO Interface and JointBlock. To maintain the correspondence between the physical structure and virtual model, the size and the orientation of the 3D model seen in the image are changed in accordance with the physical structure of the blocks. The size of the image is calculated by comparing the bounding boxes of the physical and virtual models, while the orientation of the 3D model is calculated by mapping sides of the two bounding boxes to one another. In other words, the 3D virtual model shown on the display is translated and rotated in response to the shape of blocks.

In this way, the system maintains consistency between the physical constructed structure and the virtual model, so that users can intuitively interact with the virtual model through MADO Interface and the physical blocks.

**APPLICATION**

As described before, users can manipulate the virtual model by touching the surface of it through MADO Interface. The iPod Touch which we use as MADO device can recognize multi-touch and be applied to several applications. We introduce some examples of interacting with the virtual object in a 3D modeling application.

**Drawing a line**

This application allows users to draw lines on the 3D virtual model displayed on MADO Interface. Users can perform free-form sketching, and complete tasks such as drawing characters or words on the surface of the model by touching MADO Interface. Additionally, this technique is used for specifying a region on the 3D virtual model.

**Specifying a region**

We have implemented two input methods for specifying a region to edit for tasks such as painting or geometry modification. The first method is drawing a line on the surface of the 3D virtual model and selecting the inside of the area made by linking the starting point and the ending point. The second method is specifying the inside of the area that is surrounded by multiple points generated by using multi-touch.

After choosing a part of the 3D virtual model in this way, users can edit the virtual model as described below.

**Changing the color or the shape**

Users can change the color or the shape of the 3D virtual model using touch commands. The user first selects the interaction mode, such as changing the color, by pressing the button on the display. The user then specifies the region and selects the new color from the color palette on the display to change the color. In this application, changing the color is realized by modifying it of the meshes which are the elements of the virtual model. In the case of shape, users specify the area in the same way and move the touch point, and the shape of the 3D model is changed according to the direction and the distance of the movement.

**Feedback to the physical blocks**

With users’ inputs to the 3D virtual model, they can manipulate the virtual object, and the system gives feedback to the physical blocks. Figure 8 shows the interaction between the physical and virtual object of a chameleon. In this application, when users partially change the color of the chameleon on MADO Interface, the color of corresponding part of the physical block is changed into another color.
the same color as the virtual model using the LED in the block. It is also possible to give different kinds of feedback to the physical blocks by using other output devices of ActiveCube.

CONCLUSION

In this paper, we proposed a new TUI named “MADO Interface” which is utilized as a real window into the virtual world. This interface maintains the consistency between the physical and virtual worlds, so that users can intuitively look into the 3D model corresponding to the constructed structure of the blocks. It also supports flexible viewpoints through the use of JointBlock.

Moreover, MADO Interface offers users to edit the 3D virtual model in detail; for example by changing the color and the shape, etc. This solves one of the limitations of ActiveCube that users can only perform a rough modeling.

MADO Interface makes use of a web application for displaying the virtual world, which is independent from the device used as MADO Interface. As a result of this design there are some performance limitations in updating of the image representing the virtual world. We believe that this problem can be solved by developing a native application specifically for use with the iPod Touch component of the MADO Interface system. This will improve interaction with MADO Interface.

In the future, we are planning to addresses the performance problems, and develop new applications using this interface.

ACKNOWLEDGMENTS

This work was supported in part by Grant-in-Aid for Scientific Research of the Ministry of Education, Culture, Sports, Science and Technology. And we wish to thank Mr. Garth Shoemaker and members of our laboratory for their helpful comments.

REFERENCES