ActiveCube: A Bi-directional User Interface using Cubes

Yoshifumi Kitamura Yuichi Itoh Toshihiro Masaki Fumio Kishino

Graduate School of Engineering
Osaka University
2-1 Yamadaoka, Suita, Osaka 565-0871, Japan
kitamura@eie.eng.osaka-u.ac.jp

Abstract: *ActiveCube is a bi-directional user interface that supports a person's creative work through constructing and interacting with 3-D virtual objects by using actual physical cubes. With this flexible interface, the user can easily construct various 3-D structures in a virtual environment by simply combining the cubes. All of the faces of the cubes are the same and can be connected to any face of any other cube. A computer recognizes the constructed 3-D structure in real-time, so consistency is always maintained between the real environment and its corresponding representation in a virtual environment. Each cube is equipped with a processor for autonomous simulation and communication functions between cubes. A cube also utilizes a sensor for detecting the operational intention of the user and/or a display/actuator used as an output channel. Users can interact with ActiveCube in various ways by using the sensors, and the response to the interaction is also shown by displays/actuators installed on the cube. Each ActiveCube is equipped with both input and output devices, which makes the interface intuitive and helps to clarify the causal relationship between the input of the user's operational intention and the output of simulated results.*

Introduction

GUIs that use metaphors in a real world have helped to improve the efficiency of various works on computers. Recently research on more flexible and sophisticated user interfaces that use intuition and kansei (sensitivity)-based GUI design is increasing. For example, research on virtual reality utilizes a spatial proprioception and experience in a 3-D environment, and multi-modal interfaces use various sensory channels effectively. However, these approaches toward the sophisticated user interface have a problem: users feel difficulty in operating or manipulating objects because the objects are not substantial. To solve this problem, research on tangible interfaces that use physical substantial objects has recently started. Examples of research topics include architecture design [1] using machine readable models [2], object shape modeling interfaces using Lego-type blocks or triangular planes [3-6], and program coding interfaces by using blocks [7]. Though these examples use substantial objects instead of unsubstantial ones as the subjective objects, the degrees of freedom of building blocks are restricted because of the limitation of connectable faces of blocks. Moreover, the causal relationship between the input of the operational intention of the user and the output of simulated results is not clear because the input and output devices are separated.

In this paper, we propose a bi-directional user interface using ActiveCube in order to realize a flexible direct manipulation interface that utilizes the user's intuition, kansei (sensitivity), and proprioception to the full. Because the faces of the cubes are the same and each cube can be connected to any other cube, users can construct various 3-D objects by combining the cubes, as they want according to their imagination. Each ActiveCube is equipped with both input and output devices, which make the interface intuitive and help to clarify the causal relationship between the input of the user's operational intention and the output of simulated results.

0-7803-6400-7/00/$10.00 ©2000 IEEE

User Interface using ActiveCube

ActiveCube has been designed with the following features to achieve a flexible direct manipulation interface that maximizes the utilization of a user's intuition, kansei (sensitivity), and proprioception.

- Each ActiveCube is a rigid cube, and users can construct various 3-D structures by combining the cubes as they want. The faces of the cubes are the same so that each cube can be connected to any other cube.
- The constructed 3-D structures are recognized by a computer in real-time; therefore, users can construct a 3-D structure in the virtual environment that uniquely corresponds to the physical structure of the physical cubes in the real environment.
- ActiveCube is equipped with sensors that obtain the user's operational intention and/or other information from the real environment.
- ActiveCube is equipped with a display system and/or actuators to show the simulated results or internal status of the computer system.
- Consistency between the physical 3-D structure made from the cubes in the real environment and its representation in the computer (i.e., in the virtual environment) is always maintained by utilizing real-time communication channels.

ActiveCube

Hardware

ActiveCube system is developed based on the techniques used in Robocube (System Watt Co., Ltd.)[8]. ActiveCube is equipped with a CPU called Neuron Chip (Echelon Corporation). A Neuron Chip consists of three processors; one for controlling and executing application programs, and two for network communication. Executable programs for each cube, coded in NeuronC, are compiled and written in non-volatile memory on each Neuron Chip. An ID number is assigned to each cube (cube ID) for the unique identification of cubes. An infrared sensor on each face of a cube detects connections and disconnections with other cubes. An ID number is assigned to each face (face ID) of the cube to identify the connecting face. Connected cubes constitute a network where parallel RS-485 is used for communication between cubes. Four electro communication lines are needed on each face for communication between cubes. For this purpose, three contact terminals positioned on a concentric circle at the center of the face are used for communication#1, communication#2, and Vcc, as shown in Figure 1. The ground is supplied through a connecting snap.

The cubes are connected to the host PC through a special cube called a base cube. Communication is done by translating RS-485 to FT-10 on a router.

Real-time Recognition of 3-D Structure

When a new cube (a child cube) is connected to a cube that is already connected to the network (a parent cube), the parent cube sends its cube ID and connected face ID to the host PC. The child cube is supplied with power when it is connected, and broadcasts its cube ID and connected face ID. The host PC updates the status of the connection from the

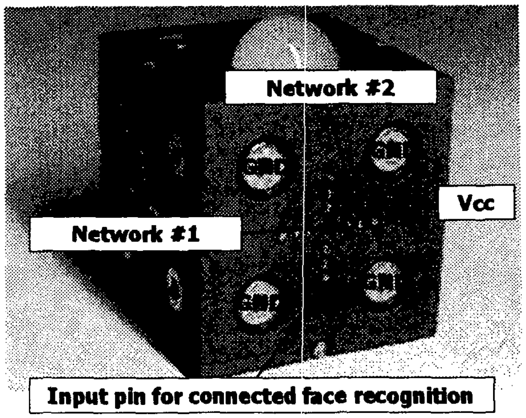

Figure 1: ActiveCube

received information. A connection-status tree (CS-tree) is used to manage the connection status information, storing cube IDs and connected face IDs in its nodes. Through constant updating of the CS-tree, real-time recognition of connection/disconnection is possible.

The 3-D structure of the connected cubes can also be recognized by using the CS-tree. The location of the parent cube is recognized by traversing the CS-tree. Through repeated analyses from the base cube to the leaves, the entire 3-D structure of the connected cubes is recognized as shown in Figure 2.

"Active" Cubes

Because each cube is controlled by a Neuron Chip, it is possible to obtain the user's operational intention and other environmental information from the sensors on the cube. The sensors can be gyros, ultrasonic, luminous, temperature sensors. It is also possible to execute an action or show a simulated result by using an actuator/display system on the cube. Examples of such systems are lights, buzzers, and motors.

Conversion to Objects

After the 3-D structure of constructed cubes is recognized, a meaning and a function are added to the structure represented in the virtual environment on the computer. The shape of the constructed cubes reminds the added meanings and functions. The operational intention of the user and other environmental information obtained from the sensors on the ActiveCube also affect the conversion. For example, when a 3-D structure constructed from several cubes is recognized by the computer and converted to an airplane, a meaning of "airplane" and a function of "fly" are added to the simple 3-D structure.

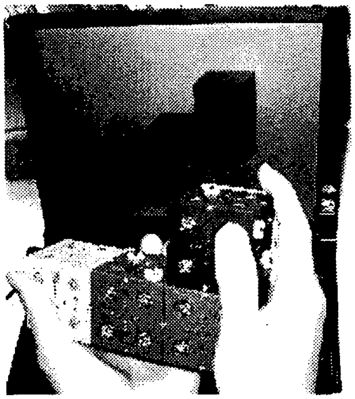

Figure 2: A snapshot of real-time recognition of 3-D structure

Once the structure is converted to an object, the sensors and display/actuators function as parts of the converted object. The appearance of the converted object also changes.

Interaction

The cubes can be combined to form various 3-D structures. A computer recognizes the 3-D structure in real-time, so consistency between the real environment and its corresponding representation in a virtual environment is always maintained.

Users can interact with ActiveCube in various ways, for example, by interacting with either the physical real cubes or their corresponding virtual representations. The response caused by the interaction is shown by using some displays/actuators on the cube. Each ActiveCube is equipped with both input and output devices, which make the interface intuitive and help to clarify the causal relationship between the input of the user's operational intention and output of simulated results. An example of interaction is shown in Figure 3.

Application

We could imagine possible applications could include: educational experience for children, learning how to assemble real and virtual objects; edutainment: new types of toys; training: increasing skills (and speed) in assembling objects for persons working in real/virtual environments; perhaps: artistic applications that support person's creativity.

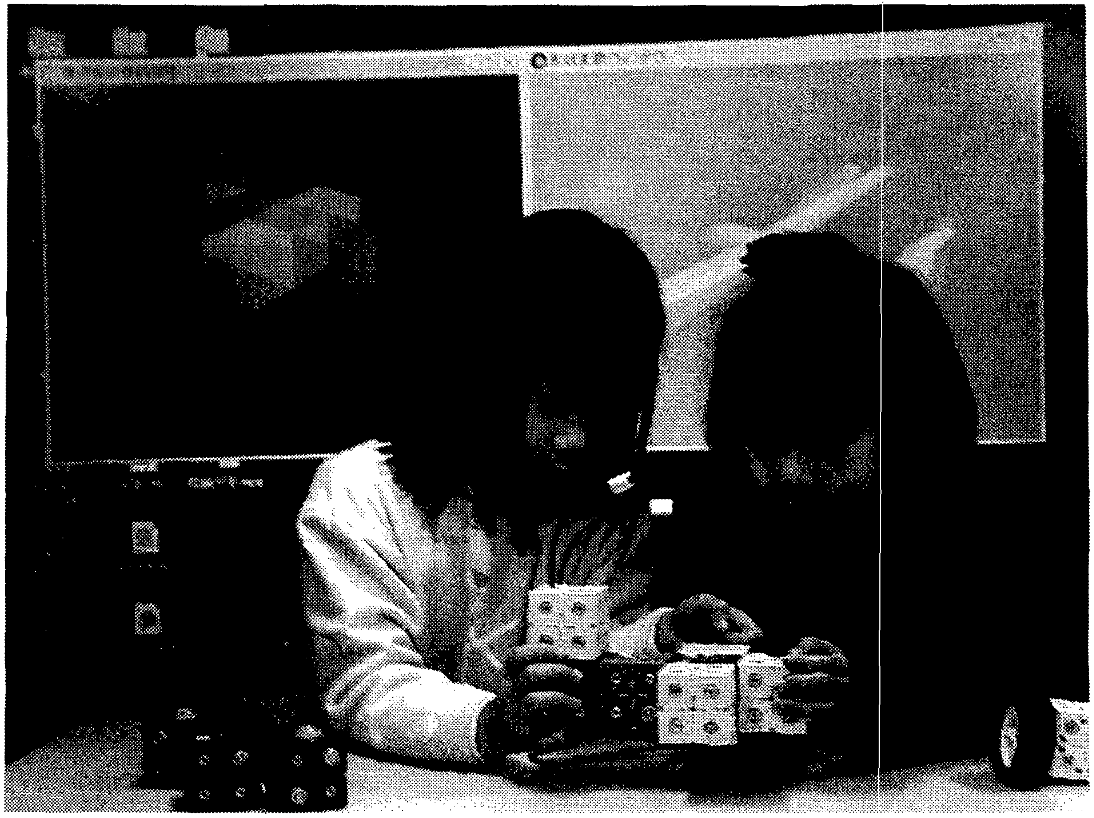

Figure 3: Children playing with ActiveCube.

Acknowledgment

We gratefully acknowledge the technical support of Hideo Kikuchi, president of System Watt Co., Ltd.

References

[1] R. Aishi and P. Noakes. Architecture without numbers – CAAD based on a 3D modeling system, Computer-Aided Design vol. 16, no. 6, pp. 321-328, 1984.

[2] J. H. Frazer. An evolutionary architecture, Architectural Association, London, 1995.

[3] G. Anagnostou, D. Dewey, and A. T. Patera. Geometry-defining processors for engineering design and analysis, The Visual Computer, 5, pp. 304-315, 1989.

[4] M. Resnick, F. Martin, R. Sargent, and B. Silverman. Programmable bricks: toys to think with, IBM Systems Journal, vol. 35, no. 3-4, pp. 443-452. 1996.

[5] M. G. Gorbet, M. Orth, and H. Ishii. Triangles: tangible interface for manipulation and exploration of digital information topography, in Proc. of Conference on Human Factors in Computing Systems (CHI '98), pp. 49-56. 1998.

[6] D. Anderson, J. Frankel, J. Marks, D. Leigh, K. Ryall, E. Sullivan, and J. Yedidia. Building virtual structures with physical blocks, in Proc. of Symposium on User Interface Software and Technology (UIST'99), pp. 71-72, 1999.

[7] H. Suzuki and H. Kato. AlgoBlock: a tangible programming language, a tool for collaborative learning. in Proc. 4th European Logo Conference, pp. 297-303, 1993.

[8] http://www.watt.co.jp/