ICDE 2005 TOKYO
International Conference on Data Engineering

Proceedings of the International Workshop on

Ubiquitous Data Management
(In Memoriam Prof. Yahiko Kambayashi)

UDM 2005

Organized by

The IEEE Computer Society
Kyoto University 21st Century Center of Excellence (COE) Program:
Informatics Research Center for Development of Knowledge Society Infrastructure
National Institute of Information and Communications Technology (NICT)
The Database Society of Japan (DBSJ)
Information Processing Society of Japan (IPSJ)
The Institute of Electronics, Information and Communication Engineering (IEICE)

April 4, 2005

National Center of Science, Tokyo, Japan

IEEE Computer Society Order Number P2411
ISBN 0-7695-2411-7
Library of Congress Number 2005925952

Additional copies may be ordered from:

IEEE Computer Society
Customer Service Center
10662 Los Vaqueros Circle
P.O. Box 3014
Los Alamitos, CA 90720-1314
Tel: + 1 800 272 6657
Fax: + 1 714 821 4641
http://computer.org/cspress
csbooks@computer.org

IEEE Service Center
445 Hoes Lane
P.O. Box 1331
Piscataway, NJ 08855-1331
Tel: + 1 732 981 0060
Fax: + 1 732 981 9667
http://shop.ieee.org/store/
customer-service@ieee.org

IEEE Computer Society
Asia/Pacific Office
Watanabe Bldg., 1-4-2
Minami-Aoyama
Minato-ku, Tokyo 107-0062
JAPAN
Tel: + 81 3 3408 3118
Fax: + 81 3 3408 3553
tokyo.ofc@computer.org

Individual paper REPRINTS may be ordered at: reprints@computer.org

Editorial production by JD Cantarella

Cover art production by Joseph Daigle, Studio Productions

Printed in the United States of America by The Printing House

IEEE Computer Society

Conference Publishing Services

http://www.computer.org/proceedings/

Proc. of Int'l Workshop on Ubiquitous Data Management
(UDM 2005), pp.86-93 (Apr. 2005).

A Ubiquitous Computing Environment Composed by Cooperation between Visual Markers and Event-Driven Compact Devices

Yasue Kishino, Tsutomu Terada
Grad. Sch. of Information Science and Technology, Osaka University
{yasue, tsutomu}@ist.osaka-u.ac.jp

Masahiko Tsukamoto
Faculty of Engineering, Kobe University
tuka@kobe-u.ac.jp

Shojiro Nishio
Grad. Sch. of Information Science and Technology, Osaka University
nishio@ist.osaka-u.ac.jp

Abstract

In this paper, we propose a new ubiquitous computing environment composed by visual markers and event-driven compact devices. We employ the VCC (Visual Computer Communication) method, which we have already proposed as an information displaying method for visual markers. In the VCC method, a marker is composed of multiple tiles whose color changes at a certain interval of time, by which the marker can send dozens of bytes of information. In our research, to enhance the facility of VCC markers, we use rule-based compact devices to process various types of data and to change information shown in a VCC marker dynamically. The combination of VCC markers and compact devices enables various services such as the presentation of real-world information collected by compact devices and control of appliances via compact devices.

1. Introduction

Recent evolutions in the miniaturization of computers and component devices, such as microchips and sensors, and in the progress of high-speed wireless networks are contributing to the realization of ubiquitous computing environments [4, 10, 12]. Especially, in a wearable computing environment, users can weare and operate their computers anywhere and anytime. In such a situation, we can realize not only conventional computing styles such as mail and Web browsing, but also a new computing style, where in users can share information with surrounding people using location information, or obtain information concerning objects in front of the user [8].

We consider that such information is presented to users by overlaying annotation information explaining the physical object on a real-space image captured from the user's point of view. This method of overlaying information onto the real-space image is called Augmented Reality (AR) [1]. We have previously proposed the VCC (Visual Computer Communication) method as such a method for presenting information. In the VCC method, we can present dozens of bytes of information using a marker comprising multiple LEDs (Light-Emitting Diode) or a marker presented on computer displays, and we can change the presenting information dynamically. Moreover, we can obtain the physical relationship between the camera and the marker and overlay annotation information onto the exact location.

In the previous work, we considered that we have controlled information presented by VCC markers using a conventional PC. However, if this marker is composed of sensors and they change information on the VCC marker according to the values registered by those sensors, we can provide more flexible services to users. Therefore, we consider that it is possible to collect information from sensors using ubiquitous chips and to present this information on the VCC marker. The ubiquitous chip is an I/O (input/output) control compact event-driven device [11], that can collect information from various attached sensors and control connected output devices. We can describe its behavior using a set of event-driven rules, and construct applications by combining rules such as that a ubiquitous chip changes the ON/OFF state of the connected output devices according to the condition of its input ports. There are also sensing devices for ubiquitous computing environments such as Smart-Its [2] and MICA [7]. We employ the ubiquitous chip because of certain merits, the free assortment of various sensors, and the flexibility of its behavior.

In the proposed environment, we connect ubiquitous chips to VCC markers placed in real space and control these VCC markers using a set of rules that send various data according to input state from sensors connected to ubiquitous chips. Moreover, a mobile computer carried by a user detects a VCC marker and controls the ubiquitous chips connected to the computer according to received information from the VCC marker.

The remainder of this paper is organized as follows. Section 2 describes our assumption and our approach to realize the proposed environment. Section 3 outlines the VCC method and the ubiquitous chips then Section 4 describes the details of our proposed ubiquitous environment. Section 5 explains the implementation of prototype system and in Section 6 we discuss the new ubiquitous environment. Section 7 sets forth the conclusion and planned future work.

2. Proposed Environment

2.1. Assumption

In the proposed ubiquitous computing environment, we assume that a user wears a HMD (Head-Mounted Display) with a camera and the user's wearable computer. The wearable computer automatically records the user's behavior such as images of people met.

Various data associated with the situation surrounding user are shown on the HMD. The information is overlaid on the captured image and changed dynamically according to ever-changing factors such as time, temperature, brightness, noise, and the direction of the user. Moreover, various output devices, such as actuators, a vibrator, and LEDs, emphasizes the information and promote awareness. The following points are examples of changing presented information on the marker dynamically.

- Different presentation of information relevant even to the same thing depends on the time. For example, a store guides itself when it opens, and advertises its mail order when it closes.
- Guidance information in the same sightseeing spot differs according to the number of people.
- A system anticipates the information that a user needs by sensing his or her surrounding situation.

In the assumed environment, many computers are embedded around the environment, and they sense the surrounding situation, show the result to users, and provide services by controlling various instruments such as electric appliances and furniture established in the environment.

2.2. Approach

In our approach, we use spatial markers to present environment information. When users find a marker, they know some information is attached to that place. Moreover, when capturing the marker with his or her camera, the user's computer overlays information acquired from the marker on the captured real space image by using the marker's position as a landmark. Such information presentation using real-space markers is attracting attention as a method to get real space information [9].

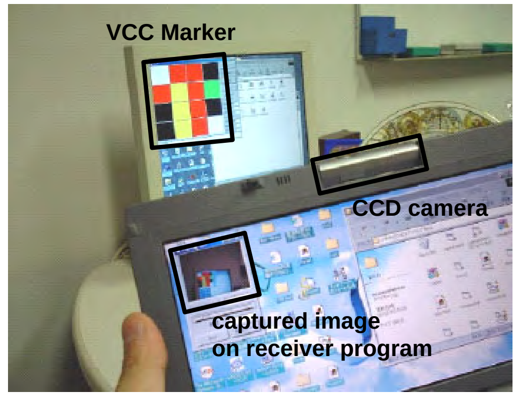


Figure 1. An example use of VCC marker

In conventional research, the paper-printed marker approach is commonly used. However, this is not sufficient for presenting dynamic information. Thus, we use the VCC marker to realize various services in our assumed environment. Let's consider using a wireless communication method such as IR (infrared) and Bluetooth instead of dynamically changing markers to satisfy the requirements. In this approach, we can obtain information more easily since the system can use special wireless modules. However, the system cannot obtain the distance and the direction between the information source and the user, and we cannot realize applications that provide information on the items nearest to the user or provide information on items only in front of a user.

We can also consider employing RFID tags. However, this approach has same problem as the wireless communication one.

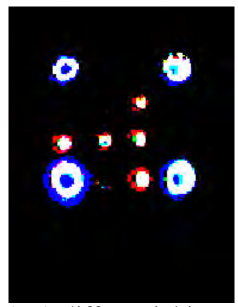
The information presented on VCC markers should change dynamically according to sensor inputs. In this research, we use the ubiquitous chip to control markers since the ubiquitous chip can handle various sensors and change its behavior dynamically. We also want to control a user's wearable devices according to detected information from the markers. It is possible to use the ubiquitous chip for these controls since it can control various output devices flexibly.

Moreover, the cooperation between a user's ubiquitous chip and ubiquitous chips fixed in the environment enables users to control instruments attached to fixed ubiquitous chips via the user's ubiquitous chip.

locator	start	end	locator
1st bit	2nd bit	3rd bit	4th bit
5th bit	6th bit	7th bit	8th bit
locator	parity	2nd byte invalid	locator

Figure 2. A coding example

Figure 3. LED Marker



(a): A differential image

W	B	B	W
B	B	R	B
R	R	R	B
W	B	R	W

(b): Recognized data

Figure 4. A procedure of image processing

3. VCC and Ubiquitous Chip

This section explains our previous work, the VCC method and the ubiquitous chip.

3.1. VCC

In the VCC method, a marker consists of multiple tiles, and their color changes at a certain interval to send dozens of bytes of information [6]. We have implemented two types of VCC marker: a marker presented on a computer display and a marker consisting of 16 LEDs. Both markers can change the presented information dynamically.

Figure 1 shows a snapshot of VCC in use. In this figure, a blinking color matrix code is presented on the display of a desktop PC. The laptop PC equipped with a CCD camera is capturing images of the marker and analyzing those images. When the laptop PC finds the marker, it obtains data from it. The VCC method consists of the marker and the receiving process algorithm.

3.1.1 The Marker

Figure 2 shows a coding example for a VCC marker, where each marker is a square divided into 4×4 small square fragments. This marker changes the color of fragments and presents data using the difference of the color variation. By repeating the color variation, the marker can present data of arbitrary length in an ideal manner. Figure 3 shows an LED marker that represents data by changing the ON/OFF state of the LED on each block.

The four corners of the marker are called location parts. These parts are landmarks for an image processing program to find the marker. These blocks repeatedly blink black/white or ON/OFF in every unit of time.

Data is transmitted within the area with the exception of the location part, which is what we call the data part. When the marker represents “1” in a block using a color, it reverses the color of the block. When it represents “0,” it does not change the color. The eight blocks in the two middle rows show data sent by this marker. These blocks can represent 1 byte using one color for coding at one unit of time and 2 bytes using two colors. The second byte-invalid block reverses its color when the marker represents 2 bytes. The parity bit is used for error detection. When the marker starts/stops sending data, it reverses the start/end bit. The LED marker adopts only 1 color mode.

We assume that screen markers are used where there are computer displays on which we can present VCC markers, and we employ LED markers where there are no computer displays or when users wear VCC markers.

3.1.2 The receiving process algorithm

The operational procedure of the receiving side is as follows. The receiving side captures a video image in the RGB gradients in an interval less than half that needed for the color to change on the marker. The computer makes a difference image of the image just taken from the image captured in the last period, then the image is binarized at a suitable threshold value (Fig. 4(a)). The obtained image is analyzed in order to cluster the white connected areas, and the four largest areas from the clusters are chosen as loca-

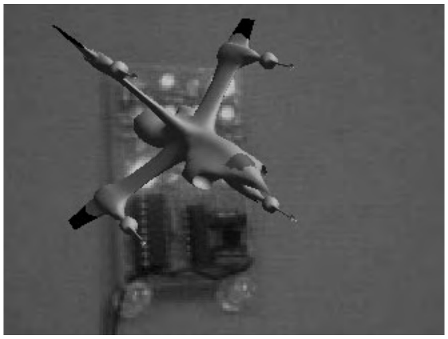


Figure 5. An example of overlaying a virtual object

Figure 6. Ubiquitous Chip

Table 1. Functions of a Ubiquitous Chip

Event	
Name	Contents
RECEIVE_MESSAGE	8 types message reception via a serial port
RECEIVE_DATA	1-byte data reception via a serial port
TIMER	Firing a timer
NONE	Evaluating conditions at all times
Action	
Name	Contents
OUTPUT	On/Off control of output ports
OUTPUT_STATE	On/Off control of state variables
TIMER_SET/KILL	Setting/Killing a new timer
SEND_MESSAGE	Sending a message
SEND_DATA	Sending 1-byte data
SEND_COMMAND	Sending a command
SEND_ADD_ECA	Sending ADD_ECA command of a rule having specified ID
HW_CONTROL	Hardware control
Commands for control ECA rules	
Name	Contents
ADD_ECA	Adding rule
CLEAR_ECA	Deleting rule(s)
REQUEST_DATA	Sending contents of memory in a specified address
ENABLE_ECA	Enabling rule(s)
DISABLE_ECA	Disabling rule(s)
REPLY_DATA	REQUEST_Replaying for REQUEST_DATA

tion parts. When the receiving program detects a transition in the marker, the program reads color information from it (Fig. 4(b)). When the program does not detect the transition, it takes the next image. Furthermore, the program calculates the distance and the angle between the marker and the camera using coordinates of the location parts on the image. Figure 5 shows an example of overlaying a virtual object, where size and direction are detected usig the distance and the angle, respectively.

We also evaluated the performance of the VCC marker. When the marker and the camera remain stationary, the program can obtain information when the camera is as far from the marker as twenty times the distance of the width of the VCC marker in one-color coding. In two-color coding, the distance is fifteen times the width of the VCC marker [6, 5].

3.2. Ubiquitous Chip

The ubiquitous chip (Fig. 6) is an I/O (input/output) control device that has five input ports, 12 output ports, two serial communication ports, a timer, and a function to receive/send messages among other ubiquitous chips. Behaviors of a ubiquitous chip are described by ECA rules. The ECA rules are used to describe the behaviors of an event-driven database. An ECA rule consists of an Event, a Condition, and an Action. We can describe receiving a message and a timer expiring as the Event, the input state and the internal state as the Condition, and the output state, setting of the timer, sending a message, and sending a command as the Action. Table 1 shows functions described in an ECA rule. The rules are stored in 128-byte memory on a ubiquitous chip and they are deleted/added by receiving commands via serial communication.

Every rule has a 1-byte ID, and we can control the rules on a ubiquitous chip by using this ID. We can add/delete/enable/disable rules and demand the content of the memory. Moreover, a ubiquitous chip can add one of its own rules that has a specified ID. By using these commands and actions, the ubiquitous chips can control each others' rule.

The ubiquitous chips exchange information among them by the SEND_MESSAGE action and the SEND_DATA action. The SEND_MESSAGE action sends eight kinds of message and the SEND_DATA action sends a 1-byte analog values of a sensor and arbitrary 1-byte data.

The input ports of ubiquitous chips are connected not only to buttons but also to various sensors such as illumi-

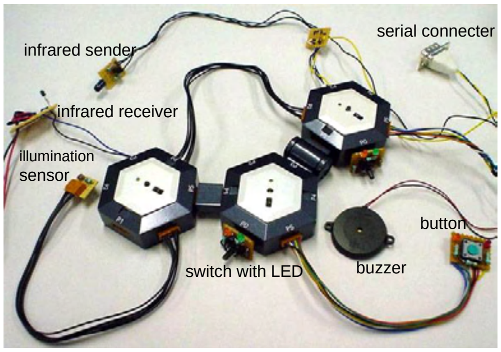


Figure 7. Attachments for Ubiquitous Chip

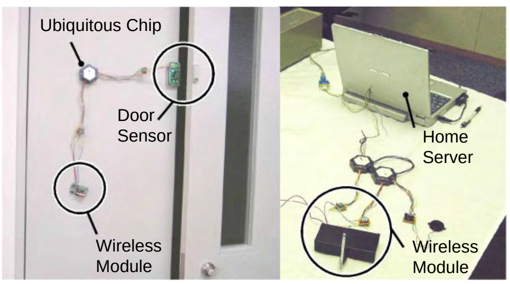


Figure 8. An example of an application

nance sensors and temperature sensors, and the ubiquitous chips can obtain the status of those sensors. A ubiquitous chip also can get an analog input on the analog port. The output ports are connected LEDs, buzzers, vibrators, and so on. As Figure 7 shows we can attach several input/output devices to one ubiquitous chip. Figure 8 shows an example of an application, which in this case is a room automation system. The home server collects the status of the door via the wireless communication module and the ubiquitous chips.

4. Cooperation of VCC Marker and Ubiquitous Chip

We propose the following two types of cooperation between VCC markers and ubiquitous chips.

Control of VCC markers by ubiquitous chips: VCC markers are connected to ubiquitous chips, and provide real-space information collected by ubiquitous chips.

Control of ubiquitous chips by VCC markers: Users obtain real-space information from VCC markers by their mobile computers, and control appliances via ubiquitous chips connected to these computers.

Table 2. The list of commands to control a VCC marker

ID	command	function
C0	Start	Start LED blinking and sending stated data (data0 or data1).
C1	Restart	Restart LED blinking after pose state.
C2	WriteData_0	Write new data to data0.
C3	WriteData_1	Write new data to data1.
C4	ChangeData	Change the sending data. After the marker finishes sending current data, start sending new data.
C5	ClearData	Clear data0 or data1.
C6	Pose	Stop sending the data.
C7	Stop	Stop blinking.

4.1. Control of VCC markers by ubiquitous chips

VCC markers are controlled by the ubiquitous chips via serial communication. In previous work, we proposed a control method for VCC markers [5]. In this method, a VCC marker can store two data (data0 and data1), the length of each being less than 31 bytes. This marker can be controlled by seven commands: Start, Restart, WriteData, ChangeData, ClearData, Pose, and Stop as shown in Table 2.

In this research, we mapped the seven commands into messages and commands in the ubiquitous chip as shown in Table 3. The data0 is used for storing an arbitrary data such as a sensor value sent from a ubiquitous chip. The data1 is used for storing a rule sent by SEND_ADD_ECA from a connected ubiquitous chip. We can use two data-storing modes for the data0. In the former mode, the marker stores the received data after the and of the latest data, whereas in the latter mode, the marker considers two received data as one set. It writes the second data to an address specified by the first data. We can switch these two modes by sending the ChangeMode commands consisting of ID 6 or 7 of the ubiquitous chip messages.

Table 4 shows an example of same rule sets. These rules change information presented in a VCC marker dynamically according to the level of the input voltage from a sensor connected to a ubiquitous chip. When the voltage is higher than the threshold value, RULE 1 sends the SEND_ADD_ECA command and the marker presents RULE 3 in binary format. Otherwise, the marker presents the sensor value by RULE 2.

4.2. Control of ubiquitous chips by VCC markers

Ubiquitous chips connected to users' mobile computers can be controlled by the read data from VCC markers us-

Table 3. How to control a VCC marker from a Ubiquitous Chip

command	command of ubiquitous chip
Start	Message (ID0)
Restart	Message (ID1)
WriteData_0	Data
ChangeMode	Message (ID6 and ID7)
WriteData_1	ADD_ECA
ChangeData	Message (ID2 and ID3)
ClearData	CLR_ECA
Pose	Message (ID4)
Stop	Message (ID5)

Table 4. An example of rules

RULE 1
E: None C: Analog input port more than 2.5 V and S0 OFF A: Sending the SEND_ADD_ECA (RULE 3) command and the message (ID2) and S0 ON
RULE 2
E: None C: Analog input port less than 2.5 V and S0 ON A: Sending the SEND_DATA command (current value of the analog port) and the message (ID2) and S0 OFF
RULE 3
E: Reception of the message C: Message ID is 1 and S1 OFF A: O1 ON

ex. E: Event, C: Condition, A: Action
S0: Flag for sending commands, S1: internal state, O1: Output port

ing image processing. Since the coding formula of VCC markers varies according to the application, we use a 1-byte header to decide the application that processes the data. We allocated IDs to different applications such as applications that use the data on ubiquitous chips, show a page on the Web browser when it reads a URL, and record the data with the image and the date when it reads an individual name or e-mail address.

5. Implementation

We implemented a prototype application as an example of our proposed environment. This application carries out a user's expected operation automatically by measuring the surrounding situation with sensors. For example, when a user presses a lightswitch, we can guess he or she wants to turn on the light when it is dark in that room and turn off it when it is light. If the system changes its behavior dynamically, the user can always get expected result with one operation. The ubiquitous chip can realize such a dynamic operation. For example, if we attach a temperature sensor and a moisture sensor to the ubiquitous chip, we can realize an electric fan that switched on when a user presses the switch if it is hot in that room, and turns off if it is not hot.

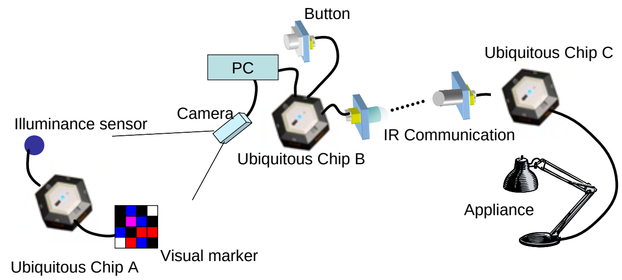


Figure 9. The structure of the prototype system

We realized a prototype system in which a computer reads rules from VCC markers connected to instruments and controls a connected ubiquitous chip and an IR (infrared) communication module.

Figure 9 shows the structure of our implemented system and Table 5 shows the rule set for this application. An illumination sensor is connected to ubiquitous chip A. When it is light and the input from the sensor is ON, RULE 1 sends the ADD_ECA command for storing RULE 8. When it is dark and the input from the sensor is OFF, RULE 2 sends the ADD_ECA command for storing RULE 9. When the marker receives the ADD_ECA command, it starts to present the rule. When the user's mobile computer reads the rule from the marker, it sends the ADD_ECA command of the rule to ubiquitous chip B. An input button is attached to ubiquitous chip B, and RULE 8 or RULE 9 in this ubiquitous chip sends the message of ID0 or ID1 to ubiquitous chip C via the IR module when the button turns ON. The types of message are decided according to the surrounding brightness, since ubiquitous chip A selects RULE 1 or RULE 2 according to the input from the illumination sensor. RULE 3 and S0 avoid the sending of needless messages. When ubiquitous chip C receives the message, it sets ON/OFF to its output port according to the received message ID and thus controls the light.

Moreover, the current system overlays some information generated from received rules on the camera image (Fig. 10).

Table 5. ECA rules for the prototype system

Rules on ubiquitous chip A

RULE 1
E: None C: I0 ON and S0 OFF A: S0 ON and sending the SEND_ADD_ECA command (RULE 8)
RULE 2
E: None C: I0 OFF and S0 ON A: S0 OFF and sending the SEND_ADD_ECA command (RULE 9)
RULE 8 (for ubiquitous chip B)
E: None C: I3 ON and S0 OFF A: S0 ON and sending the message (ID0)
RULE 9 (for ubiquitous chip B)
E: None C: I3 ON and S0 OFF A: S0 ON and sending the message (ID1)
I0: Illuminance sensor, S0: Flag for sending messages I3 (on ubiquitous chip B): Button

Rules on ubiquitous chip B

RULE 3
E: None C: I3 OFF and S0 ON A: S0 OFF
I3: Button, S0: Flag for sending messages

Rules on ubiquitous chip C

RULE 4
E: Reception of the message C: Message ID is 0 A: O1 OFF
RULE 5
E: Reception of the message C: Message ID is 1 A: O1 ON
O1: Light

Figure 10. An example of recognition from a LED VCC marker

6. Considerations

Advantage of spatial marker

In our proposed environment, calculation of the correct position and direction of the camera is important to display annotations at the correct position. The calculation is classified into two approaches: the sensor-based approach [3], and the image-based approach [9]. In the sensor-based approach, users wear various sensors such as GPS and gyro, while in the image-based approach, the system needs only cameras for users. This advantage allows for the easy construction of systems. In addition, the visual accuracy in this approach is high because the image for calculating physical relationships and an image shown to users are the same image. Thus, we use the image-based approach.

Comparison with the static marker approach

Cooperation of static paper-printed markers and a server that combines marker IDs to values of sensors placed near to the marker and rules to control ubiquitous chips also can realize similar applications to those of ours as described in Section 4. However, this approach incurs a cost to initialize the information for associating a marker ID to its position and a ubiquitous chip ID. It also incurs a cost to aggregate ever-changing sensor values to the server and to link the data to marker IDs. Thus, our approach is better from the viewpoint of construction cost. Moreover, even if a user does not connect to the network and does not bring any sensors, he or she can benefit from applications using surrounding sensors.

Information presentment to users

Though only character strings are presented on the user's mobile PC in this implementation, graphical objects are able to be overlaid for various purposes such as pointing out the position of an IR receiver with an arrow.

Moreover, the provision of information without displays is also important because users don't always attend HMD. This kind of features can also be available by using ubiquitous chip B in Figure 9 to control a vibration motor and inform the user.

Security

Though we do not consider security enough in our sample application, security is important in our assumed environment since it can control actual actuators and devices. We can develop secure systems by encrypting the data presented on VCC markers or by adding authentication data.

Applications
Our proposed environment can support many kinds of application.

- Various sensors such as temperature sensors, moisture sensors, and pressure sensors measure climate. A user obtains comparison information between his or her local climate and that of a travel destination.
- Activity sensors such as pyroelectric sensors measure the number of people passing by them. For example, in the case where this sensor is placed in front of a store, the user can know the number of people who visited the store.
- When a user visits a strange room where there are multiple appliances, he or she can control the appliances by collecting rules presented on VCC markers.

We have a plan to extend our framework to realize such various applications.

7. Conclusion

In this paper, we proposed a new ubiquitous computing environment. In this environment, we can provide various services relevant to a real-space situation using VCC markers and ubiquitous chips.

In future, we plan to construct an advanced ubiquitous computing environment by developing new markers and a framework in which a ubiquitous chip inquires information to VCC marker and utilizes the information.

Acknowledgments

This research was supported in part by The 21st Century Center of Excellence Program "New Information Technologies for Building a Networked Symbiotic Environment" and in part by Grant-in-Aid for Scientific Research (B) numbered 15300033 from the Ministry of Education, Culture, Sports, Science and Technology of Japan. The ubiquitous chip is being developed by collaboration of NEC Corp. and Osaka University. We gratefully acknowledge the following people for their collaboration in developing it: Mr. Atsushi Kashitani at Internet Systems Research Laboratories, NEC Corp. and Mr. Keisuke Hayakawa at 3rd Systems Operations Unit, NEC Electronics, and other participants of this project.

References

[1] R. T. Azuma. A survey of augmented reality. *Presence: Teleoperators and Virtual Environments*, 6(4):355–385, 1997.

[2] M. Beigl and H. Gellersen. Smart-its: An embedded platform for smart objects. In *Smart Objects Conference (sOc) 2003*, 2003.

[3] T. Höllerer, S. Feiner, and J. Pavlik. Situated documentaries: Embedding multimedia presentations in the real world.

[4] J. Kahn, R. Katz, and K. Pister. Mobile networking for smart dust. In *Proc. of ACM/IEEE International Conference on Mobile Computing and Networking (MobiCom99)*, pages 271–278, 1999.

[5] Y. Kishino, M. Tsukamoto, Y. Sakane, and S. Nishio. Realizing a visual marker using leds for wearable computing environment. In *Proc. of IEEE Int'l Conference on Distributed Computing Systems Workshops (Int'l Workshop on Smart Appliances and Wearable Computing (IWSAWC2003))*, pages 314–319, 2003.

[6] Y. Kishino, M. Tsukamoto, Y. Sakane, and S. Nishio. A visual marker using computer displays for real space applications. In *Advances in Pervasive Computing (A Collection of Contributions Presented at Pervasive 2004)*, pages 273–278, 2004.

[7] *MICA*. http://www.xbow.com/Products/Wireless_Sensor_Networks.htm.

[8] T. Nishimura, H. Itoh, Y. Nakamura, Y. Yamamoto, and H. Nakashima. A compact battery-less information terminal for real world interaction. In *Proc. of 2nd International Comference on Pervasive Computing (Pervasive 2004)*, pages 124–139, 2004.

[9] J. Rekimoto and Y. Ayatsuka. Cybercode: Designing augmented reality environments with visual tags. In *ACM Designing Augmented Reality Environments (DARE2000)*, 2000.

[10] K. Sakamura. Tron: Total architecture. In *Proc. of Architecture Workshop in Japan'84*, pages 41–50, 1984.

[11] T. Terada, M. Tsukamoto, K. Hayakawa, T. Yoshihisa, Y. Kishino, S. Nishio, and A. Kashitani. Ubiquitous chip: a rule-based i/o control device for ubiquitous computing. In *Proc. of 2nd International Comference on Pervasive Computing (Pervasive 2004)*, pages 238–253, 2004.

[12] M. Weiser. The computer for the twenty-first century. *Scientific American*, 265(3):94–104, 1991.