Mobile Magic Hand:Camera phone based interaction using visual code and optical flow

Yuichi Yoshida[1], Kento Miyaoku[1] and Takashi Satou[1]

[1] Nippon Telegraph and Telephone Corporation, NTT Cyber Communications Laboratory
Hikari-no-oka 1-1 Yokosuka, Kanagawa, Japan
{yoshida.yuichi, miyaoku.kento, satou.takashi}@lab.ntt.co.jp

Abstract. We propose the "Mobile Magic Hand" interface; it is an extension of our previous visual code-based interface system. Once the user acquires the visual code of interest, the user can then manipulate the related virtual object/system without having to keep the camera centered on the visual code. Our new interface does this analyzing the optical flow as captured by the camera. For example, consider a visual code that represents a 3D object, such as a dial. After selecting the code, the user can freely rotate and/or move the virtual object without having to keep the camera pointed at the code. This interface is much more user friendly and is more intuitive since the user's hand gestures can be more relaxed, more natural, and more extensive. In this paper, we describe "Mobile Magic Hand" , some applications, and a preliminary user study of a prototype system.

Keywords: mobile, visual code, gestural interface

1 Introduction

Interaction methods that use a visual code and a camera phone have been proposed[1][2][3]. This approach allows users to control virtual attributes with real world actions that have 6 degrees of freedom(6DOF), see Fig. 1. and Table 1. The ID of a visual code is detected and the pose and position of the mobile device relative to the code is determined by analyzing the image sequences captured by the built-in camera. The content, which corresponds to its ID, and pose and position are displayed on the mobile device. Fig. 2 shows the user adjusting a parameter, such as audio volume, by rotating the mobile device as if rotating a control dial.

The existing methods require the user to keep the camera centered on the visual code while manipulating the related virtual object/system. This prevents the use of natural and intuitive gestures. For instance, the gesture of tilting, shown in Fig. 3-(a) is intuitive. The user can't keep the camera centered on the visual code if he/she makes the natural gesture of tilting. The user's gesture must be constrained to keep the camera centered on the visual code, see Fig. 3-(b).

"Mobile Magic Hand" extracts the code's ID and gets the related virtual object/system which includes pictures, movies or music, according to the ID. "Mobile Magic Hand" measures its pose and position by analyzing the shape of the visual code

in the image when detecting the visual code of interest[4] “Mobile Magic Hand” measures its pose and position by analyzing the optical flow and feature points when the code is not detected[5].

In this paper, we describe previous works related to gestural interfaces which use a visual code and camera phone and propose “Mobile Magic Hand”. We introduce the results of preliminary user tests and examples of “Mobile Magic hand” applications.

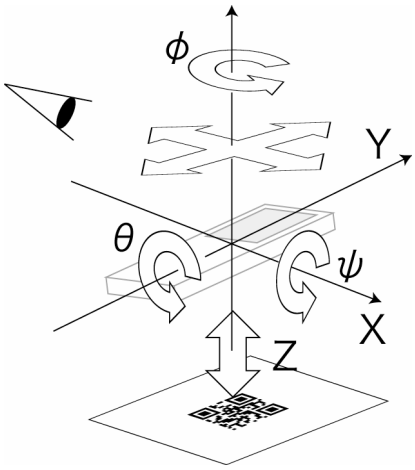
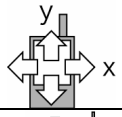
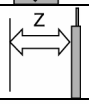
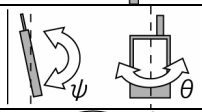
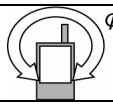


Fig. 1. Concept

Table 1. Primitive operations

Name	Image	Description
pointing		Translation in X or Y axis direction.
distance		Translation in Z axis direction.
tilting		Rotation around X or Y axis.
rotation		Rotation around Z axis.

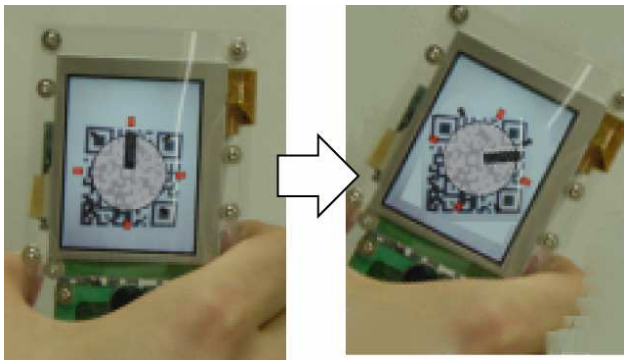


Fig. 2. Adjusting volume

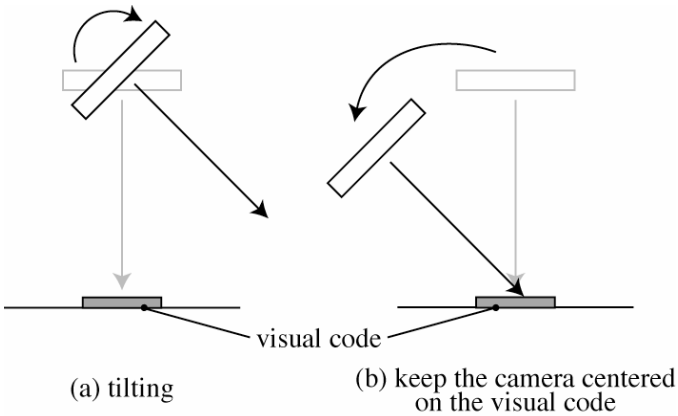


Fig. 3. One weakness of existing methods

2 Related works

2.1 Interface that uses the position and pose between the mobile phone and a visual code

This interface, based on a visual code and a camera phone, uses the distance along X, Y and Z axes and the rotation around the axes as parameters to control a virtual object/system related to the code. We associate the distance along the X and Y axes with ***pointing***, the distance along the Z axis with ***distance***, the rotation around the X and Y axes with ***tilting*** and the rotation around the Z axis with ***rotation***(Table 1). We

call them spatial operations. Spatial operations enable us to design an operation which is a metaphor for a real behavior. For example, the user can adjust sound volume as if rotating a dial knob, can acquire more detail of a map as if using a magnifying glass by zooming.

MIXIS is the interface that permits manipulation of the information related to a circle visual marker[1]. The circle visual maker consists of a circle and a figure inside it. MIXIS acquires the information by recognizing the figure inside the circle and calculates the position and pose of the phone relative to the marker by analyzing the shape of the marker's circle in the captured image. Authors built the application ImageViewZoom; it permits panning and zooming of a map by moving the phone in front of the circle. They tested this application by comparing the proposed MIXIS to the existing button operation. They reported that the examinees prefer MIXIS over the existing interface.

Rohs et al. proposed ***keystroke*** operation which lets users use buttons, and ***stay*** operation which lets users keep a position or pose for a moment in addition to our spatial operations[2] They evaluated its usability. No published work has, however, tackled the problem that the gestural operation is highly restricted because users have to keep the camera centered on the visual code.

2.2 Method of estimating the mobile phone's position and pose

There are some methods other than the visual code approach that can be used to estimate the position and pose of a mobile phone.

Acceleration sensor
We can estimate the position and pose of the mobile phone with an acceleration sensor. Acceleration sensors have already been used as the input device for mobile phones[6].

Rotary encoder
We can estimate the planar distance (for example on paper) with a rotary encoder. FieldMouse uses a barcode reader[7]. Users can point to any location by capturing the visual code and then tracing paper. But, FieldMouse enables users to operate by making a planar gesture.

Using extract of feature quantity
We can estimate position and pose by analyzing the movement of feature quantity which is extracted from the captured image. The optical flow, which is based on the motion vectors of the subject, and block matching, which uses a color histogram every area, have been reported[5][8].

We can't cheaply utilize the methods that use acceleration sensors and rotary encoders because we have to add devices to the mobile phone. It is easier to implement the method that analyzes feature quantity because existing mobile phones already have digital cameras. This method has the problem is that the unit of results is

different from the real one. But, we can convert the unit of results into the real one by analyzing the size and shape of a visual code in the captured image.

3 Proposed method ~ "Mobile Magic Hand"

3.1 Implementation

"Mobile Magic Hand" estimates the pose and position of the mobile phone relative to the visual code by analyzing the shape of the code in the captured image when detecting it. On the other hand, when the visual code can't be detected, it estimates them by analyzing the optical flow of the captured image, see Fig. 4. "Mobile Magic Hand" uses QRCode as the visual code[9].

Fig. 5 shows a flowchart of "Mobile Magic Hand" software operation. At first, it captures the desired QRCode, extracts the Code's ID, and gets the related virtual object/system which includes pictures, movies or music. When the QRCode is detected, the position and pose are calculated using the coordinates of the four corners of the QRCode (assuming a square QRCode). "Mobile Magic Hand" measures its pose and position by analyzing the optical flow and feature points if the visual code cannot be detected.

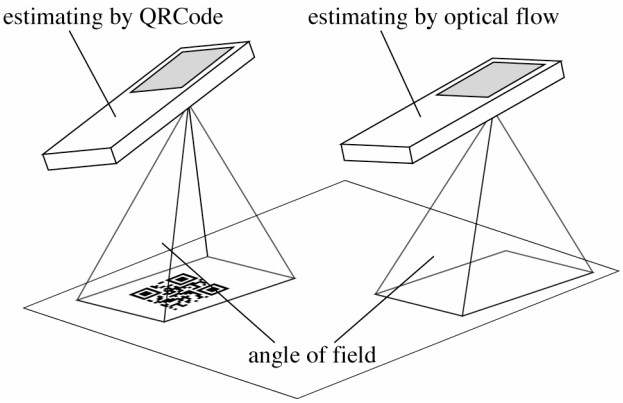


Fig. 4. Switching acquisition mode

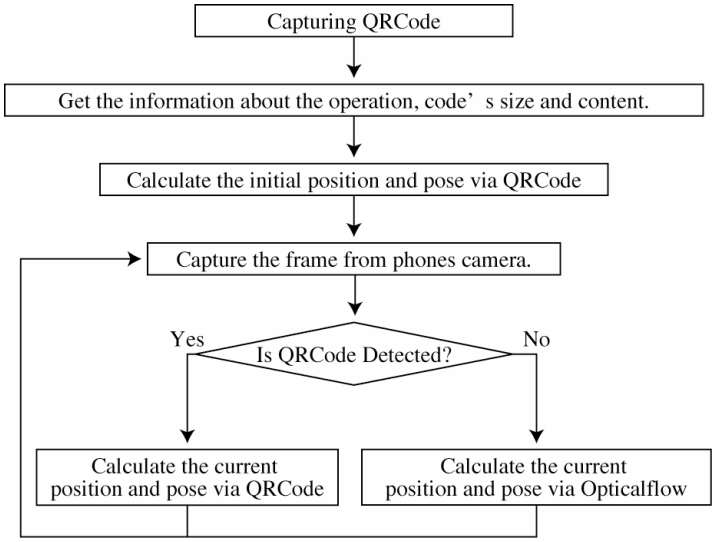
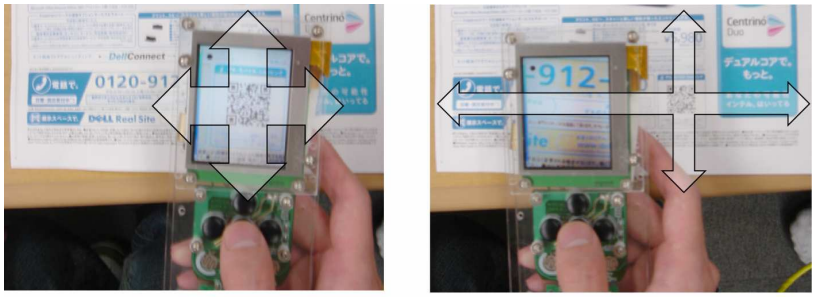


Fig. 5. Flowchart of software operation

3.2 Feature

"Mobile Magic Hand" allows users to manipulate the related virtual object/system by making a gesture even when the terminal's camera can't detect the visual code. Therefore, our proposal has two advantages over the existing methods. First, the user can access a wider range of values because the spatial range is extended. For example, if ***pointing*** is used for menu selection, the number of items that can be selected is increases, see Fig. 6. If ***distance*** is used for map zooming, greater scale changes are possible.

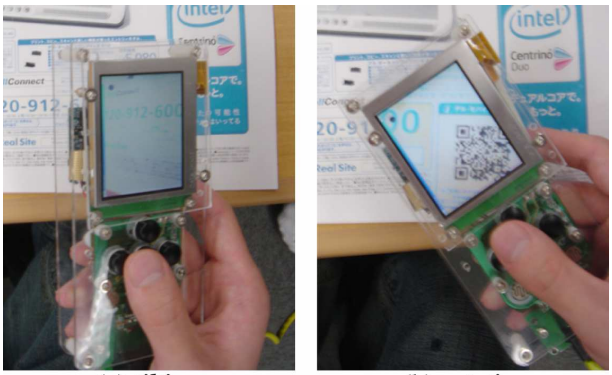


(a) The existing method
4 choices

(b) The proposed method
4 + α choicces

Fig. 6. Advantage of *pointing*

Second, once the user acquires the visual tag of interest, the user can then manipulate the related virtual object/system without having to keep the camera centered on the visual code. It allows hand gestures to be more intuitive, relaxed, natural, and extensive, see Fig. 7. For example, the users can rotate the virtual object on the terminal's screen as if rotating a real world object by rotating the terminal.



(a) ***tilting*** (b) ***rotation***

Fig. 7. No need to keep the camera centered on the visual code

4 User test

We constructed a prototype terminal that is as small as current cellular phones and used it to evaluate the performance of "Mobile Magic Hand". The prototype terminal was connected to a PC by a cable in order to run the software, see Fig. 8.

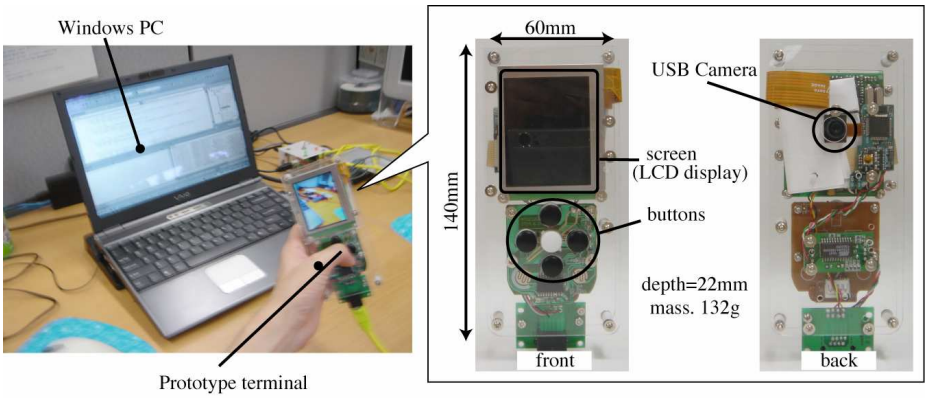


Fig. 8. Prototype system

We asked some subjects with no previous experience of "Mobile Magic Hand" to use the prototype terminal. The tasks for ***distance***, ***tilting*** and ***rotation*** were to adjust

the parameters of the pose and position to the target values. The task for ***pointing*** was to select the target objects from among the objects surrounding the visual code.

After all subjects were taught how to use the prototype system, all could perform the tasks for ***distance***, ***tilting*** and ***rotation***. Some subjects had difficult with ***pointing*** because the prototype system couldn't calculate the ***pointing*** parameters within an acceptable degree of error if the prototype terminal was moved too quickly. After being instructed to move the terminal more slowly, they could perform this task.

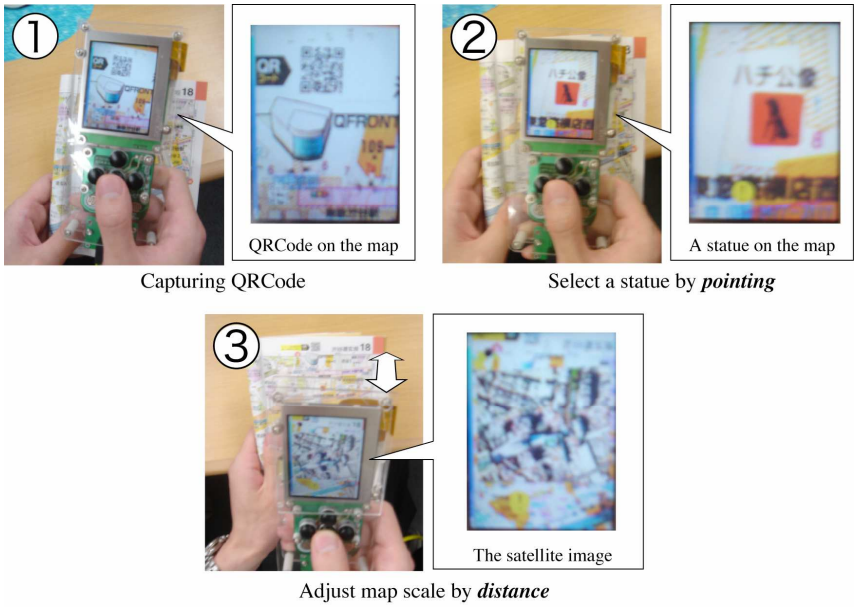
The results of the preliminary user study show that the advantages of the proposed system make ***distance***, ***tilting*** and ***rotation*** more useful.

5 Applications

"Mobile Magic Hand" will yield many new applications. We describe two prototypical applications: the interactive map and an adventure game paper.

Interactive map

The interactive map uses ***pointing*** and ***distance***. ***Pointing*** enables users to select a region of interest on the map, and ***distance*** enables users to adjust system parameters. "Mobile Magic Hand" extracts the position and operation of landmarks and content, which includes satellite images, according to the ID.



Capturing QRCode

Select a statue by ***pointing***

Adjust map scale by ***distance***

Fig. 9. Interactive map

The user moves the mobile phone to the landmark of interest after capturing the QRCode(***pointing***). As the user scans the terminal, “Mobile Magic Hand” detects the lateral movement and determines the landmark of interest. The user can get a satellite image of the area around the landmark by pushing the select button. Moreover, the user can change the scale of the satellite image by bringing the phone toward or away from the map(***distance***).

Adventure game paper

The adventure game requires the user to manipulate virtual objects printed on a piece of paper by making natural gestures. Fig. 10 shows an adventure game paper that contains a door, a cashbox, a chest, and a piece of paper on a table. User can open the door by ***tilting***, unlock the cashbox by ***rotation***, and open a drawer of the chest by ***pointing***. In addition, user can also move the paper and check the table by ***pointing***.

We can combine these operations and design complex adventure games. For example, a possible scenario is that the user opens the drawer, finds a paper on which a number is written. The number is used to unlock the cashbox.

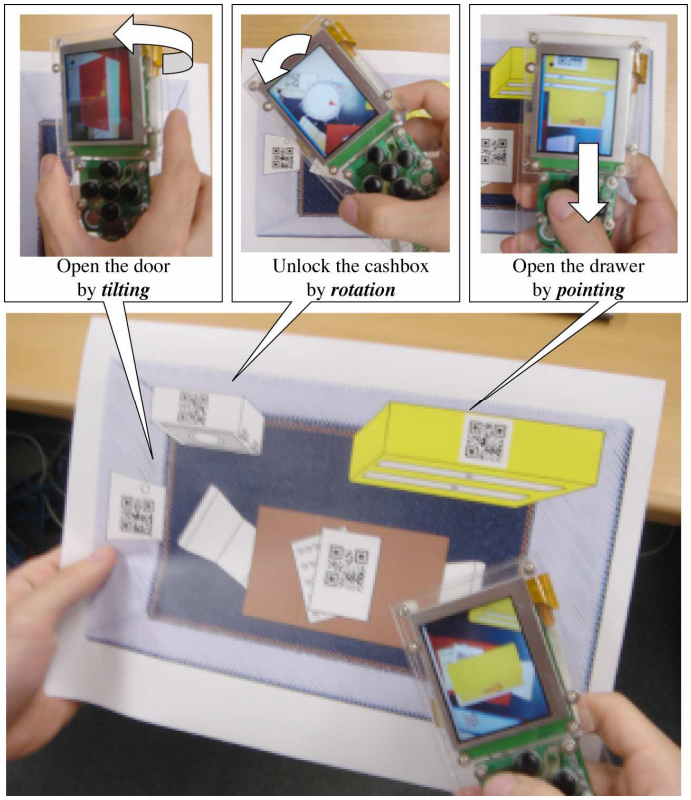


Fig. 10. Adventure game paper

6 Conclusion

Our proposal, “Mobile Magic Hand,” calculates the mobile phone’s position and pose by analyzing the optical flow if the visual code can’t be detected. We described a preliminary evaluation of the performance of its basic operations. The results confirm that “Mobile Magic Hand” well realizes the key functions of ***rotation***, ***tilting*** and ***distance***.

We intend to develop a new method that can more accurately calculate the mobile phone’s translation in order to improve the usability of ***pointing***, and evaluate the performance of practical applications of “Mobile Magic Hand”.

References

1. Hansen, T., R., Eriksson, E. and Lykke-Olesen, A.: Mixed Interaction Space - Expanding the Interaction Space with Mobile Devices. In People and Computers XIX - The Bigger Picture, Proceedings of British HCI, Springer. (2005)
2. Rohs, M. and Zweifel, P., A.: Conceptual Framework for Camera Phone-based Interaction Techniques. Pervasive Computing: Third International Conference, PERVASIVE2005, Lecture Notes in Computer Science (LNCS) No. 3468. (2005)
3. Yoshida, Yo., Satou, T., Miyaoku, K. and Higashino, S.: Reacher Interface for Intuitive Information Navigation. INTERACT2005. (2005)1091–1095
4. Rekimoto, J. and Nagao, K.: The world through the computer:Computer augmented interaction with real world environments. Proceedings of the 8th annual ACM symposium on User interface software and technology. (1995)29–36
5. Lucas, B., D. and Kanade, T.: An Iterative Image Registration Technique with an Application to Stereo Vision. International Joint Conference on Artificial Intelligence. (1981)674–679
6. Wigdor, D. and Balakrishnan, R.: TiltText : using tilt for text input to mobile phones. Proceedings of the 16th annual ACM symposium on User interface software and technology. (2003)81–90
7. Siio, I., Masui, T. and Fukuchi, K.: Real-world interaction using the FieldMouse. In Proceedings of the ACM Symposium on User Interface Software and Technology, (1999)113–119
8. Wang, J. and Canny, J.: TinyMotion : Camera Phone Based Interaction Methods. Conference on Human Factors in Computing Systems CHI2006 extended abstracts on Human factors in computing systems. (2006)339–344
9. QRCode, http://www.qrcode.com/