2011 International Symposium on Ubiquitous Virtual Reality

ARWand: Phone-based 3D Object Manipulation in Augmented Reality Environment*

Taejin Ha and Woontack Woo
GIST U-VR Lab.
500-712, S. Korea
tha@gist.ac.kr and wwoo@gist.ac.kr

Abstract— **In this paper, we suggest a mobile phone-based indirect 3D object manipulation method that uses sensor information in an augmented reality environment. Specifically, we propose 1) a method that exploits a 2D touch screen, a 3DOF accelerometer, and compass sensors information to manipulate 3D objects in 3D space, 2) design transfer functions to map the control space of mobile phones to an augmented reality (AR) display space, and 3) confirm the feasibility of the transfer functions by implementation. Our work could be applicable to the design and implementation of a future mobile phone-based 3D user interface for AR application in normal indoor and outdoor environments without any special tracking installations.**

Keywords-component; Augmented reality; HMD-based wearable computing; 3D object manipulation; sensor based interaction; mobile phone.

I. INTRODUCTION

At present, mobile phones are often used as multipurpose portable computers. Various sensors (e.g., proximity, touch screen, accelerometer/gyro, compass, and GPS) and wireless communication modules (e.g., NFC, Bluetooth, Wi-Fi, and 3G) have been embedded in recent models of mobile phones. These features can be utilized for novel phone-based interaction methods [1] in an augmented reality (AR) environment. Moreover, within an outdoor AR experience, portable head-mounted display (HMD) devices[1] that can be connected to the mobile phone have been commercialized.

Considering the use of mobile phones' sensor information for 3D manipulation, the 2D input from a touch screen sensor and rotation from an accelerometer sensor have been predominantly used. This, however, limits intuitive 3D interaction (e.g., a 3D translation) using the 2D and rotation inputs. Moreover, 3D interaction is impossible in normal indoor and outdoor spaces without phone tracking installation (e.g., ultrasonic or infrared tracking devices), as is possible in a laboratory.

To enable intuitive, phone-based 3D object manipulation in the normal AR environment (Figure 1), we propose a sensor fusion method that utilizes sensor information from the phone. Then, we adopt transfer functions to map the control space of the phone into the AR display space. Finally, we present its implementations to confirm feasibility.

Figure 1. A user is wearing a HMD and grasping a mobile phone to manipulate 3D object in AR environment.

This paper includes the following content: Related work is reviewed, and then the technical details of the 3D object manipulation method is explained. The implementation is then presented and a conclusion wraps up the information.

II. RELATED WORK

In virtual reality environments including AR, the 3D manipulation method is generally categorized into the direct manipulation method in that control space and display space correspond to one another. Such is in opposition to the indirect manipulation method, in which the spaces do not correspond to one another [2].

Specifically, the direct 3D object manipulation method is also divided into the virtual hand technique [3] for arm-reach 3D object manipulation and the virtual ray technique for far-distance 3D object manipulation. These techniques definitely require indoor tracking installations in order to determine the position and orientation of the manipulation prop (e.g., VR wand).

On the other hand, the external tracking installations are not mandatory for or the indirect 3D object manipulation method, because the internal sensor information of the manipulation prop (e.g., trackball) is sufficient for 3D object manipulation. This type of manipulation method is

---

* This work was supported by the Global Frontier R&D Program on <Human-centered Interaction for Coexistence> funded by the National Research Foundation of Korea grant funded by the Korean Government (MEST) (NRF-M1AXA003-20100029751).

[1] VUZIX. http://www.vuzix.com

978-0-7695-4420-5/11 $26.00 © 2011 IEEE
DOI 10.1109/ISUVR.2011.14

appropriate for our intention, which uses mobile phones' internal sensor information for 3D object manipulation without the incorporation of an external tracking component.

Considering the use of sensor information that is embedded in a mobile phone, tilt, accelerometer, gyro, GPS, compass sensors are used for immersive game control and remote desktop UI control. The sensor information was also exploited to 2D translation of 3D object [4], scroll [5], menu selection [6], navigation [7], and text input [8]. However, the 3D translation manipulation of 3D objects has not yet been reported.

We also need to design a control for display functions (e.g., transfer function) for new types of 3D manipulation methods. This function maps the manipulation quantities on control devices to visualization quantities on display devices. The control function is divided to the position and the rate control method [9].

The position control method inputs a magnitude of control motion vector and maps a magnitude of visual motion vector. Linear, Sigmoid, and Quadratic functions are generally used for them [6]. On the other hand, the rate control method inputs the control motion speed and maps the visual motion speed.

III. PHONE-BASED 3D OBJECT MANIPULATION

A. 3D motion vector generation from a phone

We suggest a phone-based 3D object manipulation method that exploits the phone's internal sensor information. This approach enables the user to manipulate 3D objects in normal indoor or outdoor environments, and tracking devices (e.g., infrared or ultrasonic-based tracking modules) are not installed.

The mobile phones' sensor information is exploited to compose the 3D motion vector, as shown Figure 2. The vector consists of a 2D motion vector V_t on the touch screen and a 3D rotation matrix $\mathbf{R}$ on the mobile phone. The multiplication of the vector V_t and the matrix $\mathbf{R}$ generates the 3D motion vector V_C, as shown Equation 1.

Specifically, the V_t refers to the 2D motion vector of a fingertip on a touch screen if its magnitude is over a specific distance t. (e.g., the user touches the screen and swipes over some distance). For the 3D rotation matrix $\mathbf{R}$, the pitch and roll orientation of a mobile phone is used to determine the x- and y-axes' acceleration of gravity ($\theta_{g_x}, \theta_{g_y}$). The compass sensor information is used to set the mobile phone's yaw orientation (θ_C).

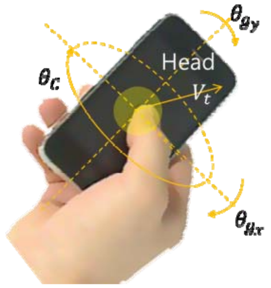


Figure 2. 2D motion vector V_t on the touch screen and rotation matrix $\mathbf{R}$ of the mobile phone make 3D motion vector V_C.

$$V_C = V_t\,\mathbf{R},\ if\,\|V_t\| > t \tag{1}$$

Then, the 3D motion vector V_C on the control space is transferred to the 3D motion vector V_D on the AR display space, as shown in Figure 3. In order to successfully accomplish this objective, we need an appropriate transfer function that should meet accurate and rapid 3D manipulation with a small number of clutching.

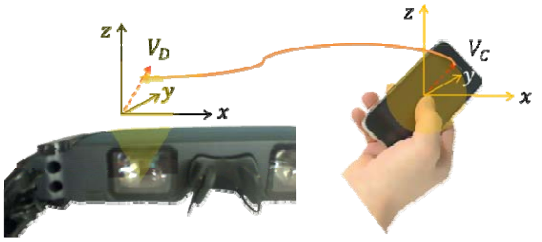


Figure 3. The transfer function maps control motion vector V_C to visual motion vector V_D in the AR display space.

In practice, the 3D motion vector V_D in the AR display space is continuously accumulated from the initial position, as shown in Figure 4, where the bounding spheres reveal magnitudes of the V_D on current time t from initial time 0.

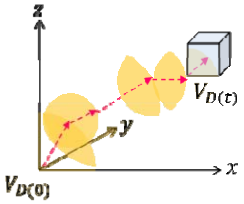


Figure 4. The V_D is accumulated on previous position.

B. Transfer Function Design

Considering the limited size of the touch screen, an appropriate transfer function is necessary in order to map the V_C to the V_D. In this paper, we adopt the position and rate transfer control functions and tune them for the phone-based 3D translation manipulation.

The position control function inputs the magnitude of motion vector M_C as an argument (Equation 2). We consider the linear (Equation 3), sigmoid (Equation 4), and quadratic functions (Equation 5) for the position control function.

$$f_{Control} = f(M_C),\ where\ M_C = ||V_C|| \tag{2}$$

$$f(M_C) = \alpha M_C \tag{3}$$

$$f(M_C) = \gamma\left(\left(\frac{1}{1+e^{-(\alpha M_C-\beta)}}\right) - 0.5\right) \tag{4}$$

$$f(M_C) = (\alpha M_C)^2$$

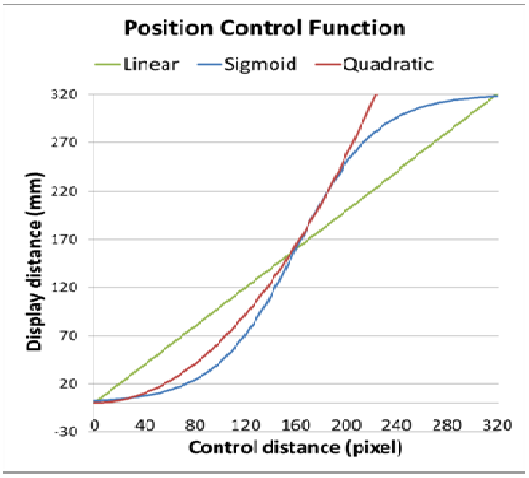


Figure 5. Design of the position control functions.

In tune, we set the maximum input value of the control functions to 320, because the pixel resolution of the x- axis of the touch screen is 320 pixels. Specifically for the linear function, we set 1 to the constant parameter α. For the sigmoid function, we set 1/32 to α, 5 to β, and 320 to γ. For the quadratic function, we set 0.08 to α. Figure 5 shows the effects of the position control functions.

For the rate control function, we refer the mouse cursor acceleration graph of the commercial operating system. The graph defines relationships between the movement speed of the cursor on display devices and the movement speed of mouse input devices.

The rate control function inputs the speed of the fingertip motion vector on the touch screen at every interval time, as shown in Equation 6. To exploit the characteristic of the mouse cursor acceleration graph[2], we simply scale the graph considering the ratio of cursor speed $S_{coursor}$ and mouse speed S_{Mouse}, and the ratio of AR object speed $S_{AR\ object}$ and touch speed S_{Touch} , as shown Equation 7.

The S_D (inch/sec) is calculated through Equation 8. The screen update rate is the camera frame rate (e.g., 30Hz) and the screen resolution is 25.4DPI, considering mm unit in our AR implementation.

Calculating the S_C (Equation 9), the sensing and network communication rate is 60Hz, considering the real-time wireless communication in the local network and the update rate of the mobile phone's accelerometer sensor. The touch screen resolution is 160DPI, according to the hardware specification of the tested mobile phone. Figure 6 shows the effects of the rate control functions.

$$f_{Control} = f(S_C), \text{where } S_C = ||V_C||/dt \tag{6}$$

$$\frac{S_{Coursor}}{S_{Mouse}} = \alpha \frac{S_D : S_{AR\ object}}{S_C : S_{Touch}} \tag{7}$$

$$S_D = \frac{Screen\ Update\ Rate}{Screen\ Resolution} \tag{8}$$

$$S_C = \frac{Sensing\ and\ Network\ Rate}{Touch\ Screen\ Resolution} \tag{9}$$

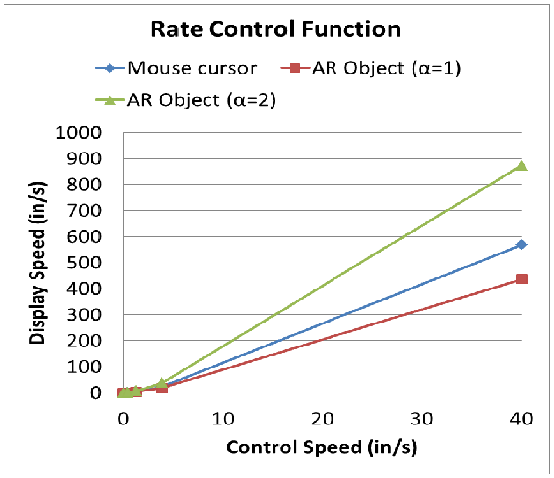
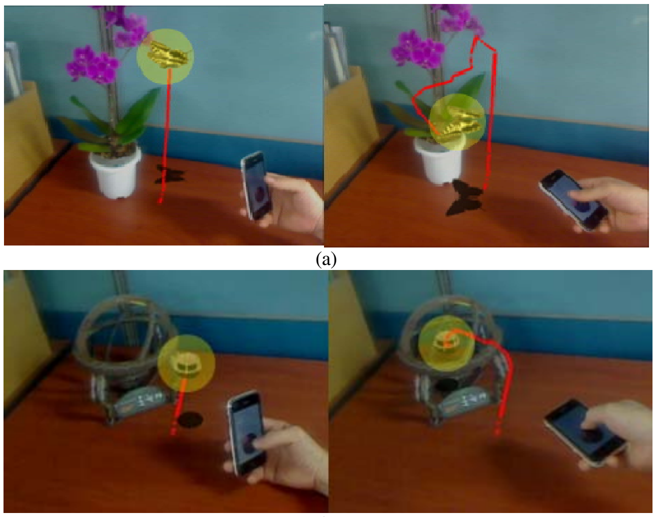


Figure 6. Design of the rate control function.

IV. IMPLEMENTATION

The implementation was conducted in a normal AR environment without any external tracking installations. The notebook computer (CPU duo core, 4GB memory) in the user's backpack executes computer vision-based feature tracking and calculates the HMD camera pose (i.e., the user's viewpoint) for proper 3D object registration within the real environment.


(a)

(b)

Figure 7. Implementation results of 3D object manipulation using a sensor-embedded mobile phone: (a) 3D object movement; (b) 3D object composition in 3D space.

---

[2] Pointer ballistics for Windows XP/Vista, http://www.microsoft.com/whdc/archive/pointer-bal.mspx

V. EVALUATION PLAN

We propose a testing plan to measure optimal parameters of the transfer function that minimize manipulation time, number of clutching, and hand/finger fatigue. The parameterizations can be conducted through empirical evaluation to find optimal parameter values for each function on near or far distance.

In this evaluation, a user can see an AR scene (display space) through the video see-through HMD with a camera aimed in the user's viewing direction and moves his/her hand in 3D space (control space) while grasping a phone, as shown Figure 8. For each trial, a user is required to translate and rotate the cursor object to match the target object (e.g., 6 DOF docking test).

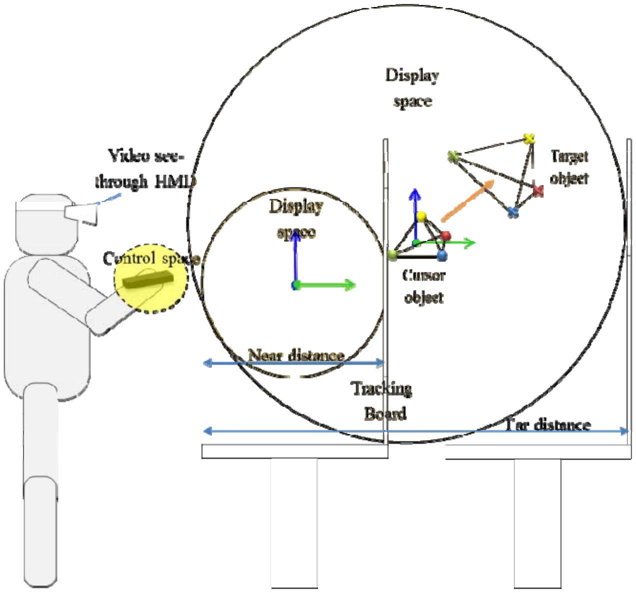


Figure 8. Evaluation environment: a user manipulates a phone in the control space and a virtual 3D object is augmented in the display space.

Through the previous evaluation result, we can calculate a linear model on optimal parameter value and distance, and then we can possibly estimate a parameter value at random distance. Therefore a user can do effective 6 DOF manipulation at any distance in ideal cases. We can also conduct an evaluation to verify how the estimated parameter is appropriate.

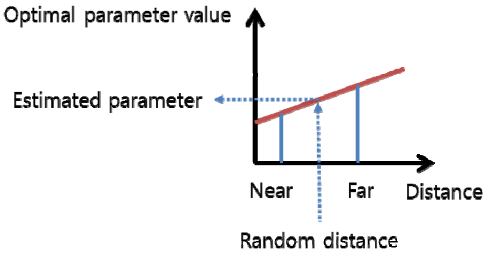


Figure 9. Estimated prarameter at random distance from the linear model.

VI. CONCLUSION AND FUTURE WORK

In this paper, we proposed a phone-based 3D manipulation method to translate a 3D object in an AR environment. The use of the phone's internal sensor information enables the phone to be used as a 3D user interaction interface in normal indoor and outdoor environments without the incorporation of any special tracking installations.

Considering the control function, if low control-to-display gain is applied, then a sophisticated translation could be possible but this requires a significant amount of clutching in order to translate the 3D object to a far distance. On the other hand, a high gain could reduce the frequent clutching, but accurate manipulation could be difficult. Therefore, we need to consider an optimal control function that satisfies both fast and accurate manipulation [10].

Thus, our future studies will attempt to compare the control functions that have been covered in this paper, and we will find optimal function parameters that minimize the task completion time, error, and number of clutching.

REFERENCES

[1] Ballagas, R., Borchers, J., Rohs, M., Sheridan, J.G., The Smart Phone: A Ubiquitous Input Phone, Pervasive Computing, 5, 1, pp. 70-77, 2006.

[2] Bowman, D.A., Kruijff, E., LaViola, J.J., and Poupyrev, I., "3D User Interfaces Theory and Practice," Addison-Wesley, 2004.

[3] Ha, T. and Woo, W., An Empirical Evaluation of Virtual Hand Techniques for 3D Object Manipulation in a Tangible Augmented Reality Environment, 3DUI, pp. 91-98, 2010.

[4] Boring, S., Jurmu, M., Butz, A., "Scroll, tilt or move it: using mobile phones to continuously control pointers on large public displays," OZCHI, pp. 161-168, 2009.

[5] Bartlett, J. F., "Rock 'n Scroll is Here to Stay," IEEE Computer Graphics and Applications, vol. 20, no. 3, 2000, pp. 40–45.

[6] Rahman, M., Gustafson, S., Irani, P., and Subramanian, S., Tilt Techniques: Investigating the Dexterity of Wrist-based Input, CHI, pp. 1943-1952, 2009.

[7] Jeon, S., Hwang, J., Kim, G. J., Billinghurst, M., "Interaction with Large Ubiquitous Displays Using Camera-Equipped Mobile Phones," Personal and Ubiquitous Computing, vol. 12, no. 2, pp. 83-94, 2010.

[8] Wigdor, D., Balakrishnan, R., "TiltText: using tilt for text input to mobile phones," ACM UIST, 2003, pp.81-90.

[9] Casiez, G., Vogel D., Balakrishnan, R., and Cockburn, A., The Impact of Control-Display Gain on User Performance in Pointing Tasks, Human-Computer Interaction, pp. 215-250, 2008.

[10] MacKenzie, I. S., Input devices and interaction techniques for advanced computing, Virtual environments and advanced interface design, pp. 437-470, 1995.