Amplifying Spatial Rotations in 3D Interfaces

Ivan Poupyrev [1, 3], Suzanne Weghorst [2], Takahiro Otsuka [1], Tadao Ichikawa [3]

[1] ATR MIC Research Laboratories
2-2 Hikaridai, Seika, Souraku-gun
Kyoto 619-02, Japan
(774) 95-1432
{poup, otsuka}@mic.atr.co.jp

[2] HIT Lab, University of Washington
Box 352142
Seattle, WA 98195, USA
(206) 685-3215
weghorst@hitl.washington.edu

[3] IS Lab, Hiroshima University
1-4-1 Kagamiyama
Higashi-Hiroshima 739, Japan
(824) 24-7669
ichikawa@isl.hiroshima-u.ac.jp

ABSTRACT

We have derived the generic equations for the zero-order control-display gain that allow for linear and non-linear amplification of spatial rotations in 3D user interfaces. Sample 3D interaction techniques have been implemented for 3D viewpoint control and object manipulation.

Keywords

3D interaction, input devices, manual control, C-D gain

INTRODUCTION

The concept of control-display (C-D) gain is fundamental in human-machine interfaces requiring continuous control by an operator. Regardless of whether humans steer a car, use a computer mouse, or manipulate objects in virtual reality (VR), C-D gain mappings work beyond the scenes, transforming the user inputs captured by input devices into the movements of controlled elements. Consequently, there is a vast body of research that has been investigating C-D mappings and their implications on user performance [1].

In this paper, we extend previous research on manual control by deriving generic zero-order C-D gain equations for 3D rotations and implementing them as interaction techniques. Although C-D gain mappings have been routinely used with multiple degree-of-freedom (DOF) input devices, their use has been limited only to position and navigation tasks. When it comes to 3D rotations, most researchers as well as producers of commercial devices and software have used only the most basic one-to-one mapping. In fact, we are not aware of any previous attempts to develop interaction techniques that use other forms of C-D mappings for multiple DOF input devices in 3D rotation task.

C-D MAPPINGS FOR SPATIAL ROTATION

Rotations in 3D space are significantly more confusing then they appear, since they do not follow familiar laws of Euclidean geometry. The standard mathematical representation for rotations that we will be using here involves *quaternions*. For our purposes, we need only few basic facts about quaternions; a detailed discussion can be found in [2]. 1) A quaternion q is a four-dimensional vector often represented as a pair ($\mathbf{v}$, w), where w is a real number and $\mathbf{v}$ is a 3D vector. 2) Given quaternions q and q', we can calculate their multiplication qq', length $|q|$, and inverse q^{-1}. 3) The rotation about unit axis $\hat{\mathbf{u}}$ by angle ϑ can be represented as a unit quaternion in two equal forms:

$$q = (\sin\frac{\vartheta}{2}\hat{\mathbf{u}}, \cos\frac{\vartheta}{2}) = e^{\frac{\vartheta}{2}\hat{\mathbf{u}}} \qquad (1)$$

4) The rotation of a vector $\mathbf{v}$ about axis $\hat{\mathbf{u}}$ by angle ϑ can be computed as $\mathbf{v}' = q\mathbf{v}q^{-1}$ and the sequence of rotations q_1, q_2 can be computed as their multiplication q_2q_1.

Let q_c be the orientation of a multiple DOF input device:

$$q_c = (\sin\frac{\vartheta_c}{2}\hat{\mathbf{u}}_c, \cos\frac{\vartheta_c}{2} = e^{\frac{\vartheta_c}{2}\hat{\mathbf{u}}_c},$$

where $\hat{\mathbf{u}}_c$ is the momentary axis of rotation and ϑ_c is the angle. The zero-order C-D gain should amplify the angle of rotation ϑ_c by C-D ratio coefficient k leaving axis $\hat{\mathbf{u}}_c$ intact:

$$q_d = (\sin\frac{k\vartheta_c}{2}\hat{\mathbf{u}}_c, \cos\frac{k\vartheta_c}{2}) = e^{k\frac{\vartheta_c}{2}\hat{\mathbf{u}}_c} = q_c^k$$

Therefore, the basic equation of the zero-order C-D gain for spatial rotations is a power function of the form:

$$q_d = q_c^k, \qquad (2)$$

where q_c is the device rotation, q_d is the displayed rotation, and k is the C-D gain coefficient. Compare this with a zero-order C-D gain equation for positioning [1]:

$$D_d = kD_c,$$

where D_c is the controller displacement and D_d is the displacement of the display element. Instead of multiplying by k, we take the device rotations in power of k, because unlike translations they are combined by multiplication. For example, to double the amount of rotation defined by q_c we have to *multiply* q_c by itself, i.e., take it in the power of two.

Quaternion q_c defines the orientation relative to the initial orientation of the device. Sometimes, however, it might be useful to calculate the C-D gain relative to *any* desired orientation q_0. The following equation allows this to be done:

$$q_d = (q_c q_0^{-1})^k q_0. \qquad (3)$$

Notice that Eq. 3 is identical to Shoemake's *slerp* function used for quaternion interpolation [2]. Indeed, while Shoemake *interpolates* quaternions using a great arc on a quaternion 3-sphere, we *extrapolate* orientation q_c using the great arc connecting q_0 and q_c.

Equations 2 and 3 "scale" device rotations uniformly. In some applications, however, it might be very useful to apply non-linear mappings that maintain a small C-D ratio (and, therefore, a better accuracy) close to the initial orientation q_0, and that increase this ratio (and the speed of rotation) as the user rotates the device further from q_0.

To introduce non-linear C-D mappings, we define the distance between rotations q_c and q_0 as the angle of the smallest rotation connecting q_c and q_0. This angle can be calculated as $\omega = 2\text{arcosine}\,(q_c \cdot q_0)$, where $q_c \cdot q_0$ denotes their dot product [2]. Now, to develop a non-linear C-D mapping for spatial rotations, we simply replace the coefficient k in Eqs. 2 and 3 with non-linear function $F(\omega)$:

$$k = F(\omega) = \begin{cases} 1 & \text{if } \omega < \omega_0 \\ f(\omega) = 1 + c(\omega - \omega_0)^2 & \textit{otherwise} \end{cases}, \quad (4)$$

where ω0 is the threshold angle and c is a coefficient. This equation has a very simple interpretation. As long as the device orientation qc is in a close vicinity of the zero orientation q0, i.e., the distance ω between them is less than ω0, the C-D gain has a constant ratio of 1. Therefore, the rotation is a one-to-one mapping. When the user rotates the device further than ω0, the C-D gain becomes a non-linear function *f* of the distance ω: the further the user rotates the device, the larger the C-D ratio becomes. To insure a smooth transition between the two parts in Eq. 4, *f* should be C1 continuous in ω0 and *f*(ω0) = 1. It is easy to show that the function in Eq. 5 satisfies these requirements.

INTERACTION TECHNIQUES

The equations developed in the previous section define generic forms of linear and non-linear C-D mappings for spatial rotation. Using these equations, it is easy to develop a wide range of interaction techniques with desired properties for various input devices and applications.

For example, we have used these equations to develop a simple 3D interaction technique for viewpoint control in a desktop VR, where the orientation of the viewpoint was controlled by the user's head tracked by a camera. An important problem of using head rotations for interaction in desktop environments is that the range of rotations that can be tracked and used for interaction is severely limited. Indeed, even small head rotations would make viewing the desktop screen uncomfortable and, after a certain angle, impossible. In addition, in the case of computer vision, excessive head rotations increase the tracking errors because fewer facial features (used for tracking) become visible to the camera.

Using non-linear C-D mappings (Eqs. 3 and 4) we amplified user's head rotations and allowed him to control a 3D kabuki mask whose orientation corresponded to the orientation of the virtual viewpoint (Figure 1). The mask was registered with the user's face on a live video stream from the camera and its orientation was updated in real time as the user rotated his head. Therefore, the user could relate both the orientation of his own face and the orientation of the virtual viewpoint, represented as a kabuki mask.

Another technique that we developed allowed to manipulate 3D objects using the Polhemus 6DOF tracker. Virtual rotations were amplified using the linear C-D gain (Eqs. 2 and 3). The technique allowed a large range of object rotations to be achieved with single hand movements. If C-D gain was not used, the user would have had to either rotate the device with his fingers or use "clutching"; though an appropriate device design can make this easier [3], using C-D gain provides an additional way to solve this problem.

Both techniques were preliminarily evaluated during their demonstrations for simple "toy" tasks. These initial evaluations showed that the mappings felt quite intuitive and that most of the users did not have any problems in using them. Moreover, in the object rotation task, some of the users did not even notice the difference between the one-to-one mapping and a mapping that doubled the amount of rotations.

CONCLUSIONS

In this paper, we derived generic zero-order C-D gain mapping functions for spatial rotation tasks. The equations are generic in the sense that they can be used with any device, task, or application requiring continuos spatial rotations. More work is required to develop 3D techniques for practical, rather then "toy" tasks, understand their properties, and to conduct formal human factors evaluations.

ACKNOWLEDGMENTS

We are thankful to Jock Mackinlay whose idea of using non-linear C-D gain for 3D rotation initiated this work.

REFERENCES

1. Knight, J., *Manual control and tracking*, in *Handbook of human factors*, G. Salvendy, Editor. 1987, John Wiley and Sons. pp. 182-218.
2. Shoemake, K. *Animating rotations with quaternion curves*. in *SIGGRAPH'85*. 1985: ACM., pp. 245-254
3. Zhai, S., P. Milgram, and W. Buxton. *The influence of muscle groups on performance of multiple degree-of-freedom input*. in *CHI'96*. 1996: ACM. pp. 320-327

Figure 1: The rotation of a head mapped into mask rotations. Top row: the C-D gain was not used; the mask and face rotations are the same. Bottom row: the C-D gain was used, the mask rotates (non-linearly) further than the user's face.