Hybrid Cursor Control for Precise and Fast Positioning without Clutching

Markus Schlattmann
University of Bonn, Germany

Reinhard Klein*
University of Bonn, Germany

1 Introduction

In virtual environments, selection is typically solved by moving a cursor above a virtual item/object and issuing a selection command. In the context of hand-tracking, the cursor movement is controlled by a certain mapping of the hand pose to the virtual cursor position, allowing the cursor to reach any place in the virtual working space. If the virtual working space is bounded, a linear mapping can be used. This is called a proportional control.

While proportional cursor control is very intuitive and easy to use, unfortunately, it often lacks accuracy. Selection of small virtual buttons or objects becomes unfeasible. Therefore often a relative control is employed, where only the directions of real and virtual movement must correspond. This way the accuracy does not depend on the ratio between real and virtual working volume sizes/resolutions. To allow for relative cursor movements in the entire virtual working space usually some kind of clutching mechanism is employed (i.e. a certain action couples/de-couples the virtual to/from real movements). However, the lack of force feedback and physical buttons leads to severe problems of clutching in bare-handed interaction. Therefore, recent approaches use thresholding schemes on the velocities and/or accelerations of hand movements.

For instance, PRISM [Frees et al. 2007] uses thresholds on the current hand velocity to determine the control/display (C/D) ratio. The reciprocal C/D ratio D/C defines the mapping of real movement $D_{controller}$ to virtual cursor movement D_{cursor} as follows:

$$D_{cursor} = D/C \cdot D_{controller}. \quad (1)$$

D/C is adaptively refined such that slow movement is scaled down to gain precision. In this case, the hand/controller moves faster than the cursor, which results in cursor/controller displacement. This displacement is recovered when the hand movement is directed to the cursor position (cursor movement is inhibited). Note that this way some kind of de-clutching mechanism is realized.

Originally, PRISM was designed to aid in direct manipulation of virtual objects. In this case, not only a cursor is visualized but additionally a hand model, which facilitates the understanding of the relation between real and virtual movement. In the case of selection only a cursor is visualized. This complicates the understanding of PRISM. Especially, the de-clutching mechanism leads to user irritations. Therefore, a detailed introduction and quite long familiarization phase is needed, which cannot always be ensured in practice. To this end, we introduce a novel technique not incorporating any clutching mechanism. It is a hybrid between proportional and relative cursor control and is related to PRISM.

2 Hybrid Cursor Control

The basic idea is to additionally make the C/D ratio dependent on the relative positions of cursor and controller. In our case, the controller position is derived from different hand parameters (e.g. hand position in 3D or pointing direction in 2D). To allow for reasonably using the positions of cursor and controller, their coordinates have to be in a suitable relation. Furthermore, as we want to use an adaptive C/D ratio but avoid any clutching mechanism, we have to ensure that the cursor is always located at the edge of the cursor space when the controller is at the limit of its effective range. Therefore, in our setting, the virtual cursor ranges in $[-0.5; 0.5]^n$ and the controller has an effective range of $[-1.5; 1.5]^n$. Note that the controller range need to be larger to allow for gaining precision. Under these conditions we have to ensure that the cursor/controller displacement does not exceed 1.

To allow for precise and fast cursor control we use the same adaptive C/D ratio as in PRISM. But additionally the C/D ratio is modulated depending on the previous cursor position c and the previous and current controller positions p and p' by adding f to Eq. 1 with

$$D_{cursor} = f(D/C) \cdot D_{controller}. \quad (2)$$

The function f accelerates or decelerates the cursor movement depending on the following two cursor/controller distances: the l^2-norm of the cursor/controller displacement vector $v = \frac{1}{2}(p+p') - c$ with $d_2 = \|v\|$ and a signed distance d_1 which also depends on the movement direction. d_1 is defined as the projection of v on the direction of controller movement with $d_1 = < v, \frac{p'-p}{\|p'-p\|} >$.

The sign of d_1 distinguishes whether the controller moves to ($d_1 < 0$) or from ($d_1 > 0$) the cursor position. Either a deceleration (if $d_1 < 0$) or an acceleration (if $d_1 > 0$) can reduce the cursor/controller displacement or the increase of cursor/controller displacement. The value of d_1 describes the size of the cursor/controller displacement dependent on the controller movement direction. We use it for indicating the amount of cursor acceleration/deceleration. Following this, we define f as

$$f(D/C) = \begin{cases} D/C \cdot (1 - |d_1|) & \textbf{for} \quad d_1 < 0 \\ D/C + (1 - d_2)d_1 + d_2 d_2 & \textbf{for} \quad d_1 \geq 0 \end{cases}. \quad (3)$$

In the first case the cursor movement is decelerated depending on the value of d_1. Note that in this case $|d_1|$ never becomes greater 1. In the second case the cursor movement is accelerated depending on the values of d_1 and d_2. d_2 is used to weight the influence of d_1 and itself. This formulation is needed to ensure $|d_1|, d_2 \leq 1$, because otherwise (if simply $D/C + d_1$ would be used) the displacement could become larger than 1 if the controller is reciprocated orthogonal to the displacement vector (if $d_1 = 0$).

In general, precision is increased by moving the cursor slower than the controller. This changes the cursor/controller displacement. Using PRISM the displacement could linearly grow until the edge of the controller space is reached without reaching the edge of the cursor space. Then a backward de-clutched movement would be needed to further move the cursor. Using our technique, with increasing displacement the cursor is increasingly accelerated such that the entire cursor space can be reached without any special maneuvers. Precision can be increased without reducing rapidness or including clutching. Our method leads to superior results in menu navigation and selection tasks. No introduction or practicing is needed for an effective handling.

References

FREES, S., KESSLER, G. D., AND KAY, E. 2007. Prism interaction for enhancing control in immersive virtual environments. *ACM Trans. Comput.-Hum. Interact. 14*, 1, 2.

*e-mail: {markus, rk}@cs.uni-bonn.de