Facilitating System Control in Ray-based Interaction Tasks

André Kunert* Alexander Kulik† Christopher Lux‡ Bernd Fröhlich§

Virtual Reality Systems Group, Faculty of Media
Bauhaus-Universität Weimar

Abstract

This paper investigates the usability of tracked wands equipped with additional input sensors for system control tasks in 3D interaction scenarios. We integrated a thumb-operated circular touchpad into a hand-held wand and compared the performance of our input device to common ray-based interaction in a menu selection and parameter adjustment task. The results show that both interfaces can be highly efficient, but ray-based interaction is only competitive if large-sized graphical interface representations are provided. In contrast, touchpad input performs well independent of the size of the graphical elements due to proprioceptive and tactile feedback.

CR Categories: H.5.2 [User Interfaces]: Graphical user interfaces (GUI)—Interaction Styles;

Keywords: 3D Pointing, 3D Input Device, System Control

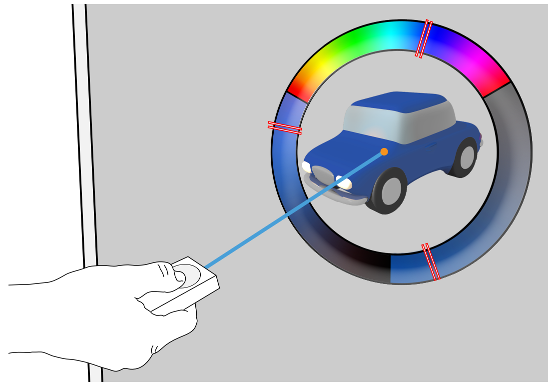


Figure 1: *The Pie Slider interface for color adjustments.*

1 Introduction

Task sequences in immersive virtual environments often consist of selecting an object followed by various manipulations of it. In between these manipulations, system control actions are used for the modification of non-spatial object attributes or for changing the context or constraints of the interaction tool. Designing sophisticated support for such interaction workflows is challenging due to the differing requirements of spatial interaction and menu control tasks. Spatial interaction may involve the operation of up to six degrees of freedom (DOF) to position and orient objects in 3D space. Menu interaction on the other hand does not necessarily benefit from 3D input, but rather from precise 2D manipulations. Nevertheless tracked wands for ray-based interaction can be found in almost all virtual reality installations, wherein the wand is used for 3D selection, spatial manipulation and also system control tasks.

Our idea was to equip the wand with an additional input controller, which is well suited for the operation of menus, buttons and sliders. Such a design allows the user to perform spatial tasks and system control tasks with the appropriate input modality, switch rapidly between both tasks, and avoid moving the ray off the selected object during 2D interaction. While wands with additional controllers such as a joystick exist, our suggestion of strictly assigning the subtasks to the appropriate input has not yet been explored. Within this paper we performed and illustrated the first step in this direction by evaluating whether a 2D controller integrated into a tracked wand is even able to compete with the omnipresent ray-based pointing technique when operating menus and sliders.

As proof of concept, we integrated a thumb-operated circular touchpad into a simple tracked wand (fig 1). With the touchpad we used the Pie Slider interface [Kulik et al. 2009] for adjusting the color of an object to a given color in the HSV color model. Our user study compared this type of menu and slider control to a common ray-based manipulation of linear sliders. The results indicate that the touchpad-extended wand benefits from tactile interaction feedback. The visual focus of the user can remain on the manipulated object for evaluating the adjustment effects rather than on the input device or the interface visualization. As a result, task performance with the Pie Slider remained unaffected by the size of its graphical representation. Ray-based interaction, however, showed a strong dependency on the size of the visual interface.

2 Related Work

3D pointing is a powerful gesture to select targeted objects and locations for the issuing of commands. In particular, in combination with speech input [Bolt 1980], such selection and command interfaces can be very effective. More common, however, is to choose commands and adapt the properties of selected objects using graphical menus. This is partly due to technical reasons but also applies the multiple choice concept, thereby relieving the user from memorizing applicable commands.

Interacting with menu systems in 3D space has been subject to many previous research projects. A comprehensive overview of the various concepts and implementations is given by Dachselt and Hübner [Dachselt and Hübner 2007].

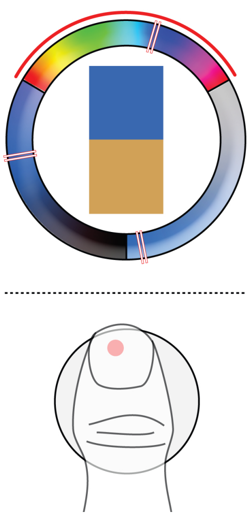
McMahan and Bowman [McMahan and Bowman 2007] studied the impact of subtask sequences on the efficiency of system control operations. They identified *Object*, *Action* and *Parameter* specifications as generic subtasks during typical operations when dealing with selection and command interfaces. Their results demonstrate the superiority of task sequences that begin with the selection of a targeted *Object*. More specifically, *Object-Action+Parameter* task sequences were found to be quite efficient.

*e-mail: andre.kunert@uni-weimar.de
†e-mail: alexander.kulik@uni-weimar.de
‡e-mail: christopher.lux@uni-weimar.de
§e-mail: bernd.froehlich@uni-weimar.de

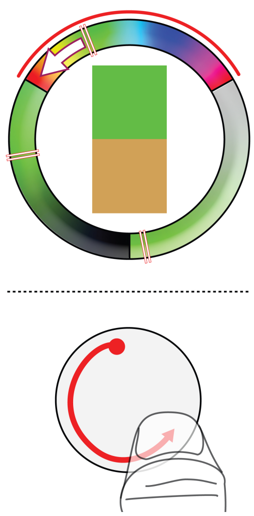
The tasks in the study of McMahan and Bowman were all operated by ray selection. The proposed sequences of subtasks, however, can also be realized with other or additional interfaces. With the Pie Slider [Kulik et al. 2009] we used for the present study, the selection of *Action* and the adjustment of *Parameter* are operated in a fluent gesture. In combination with ray-based object selection, the interface thus promotes an *Object-Action+Parameter* sequence.

3 Pie Slider Interaction

The Pie Slider [Kulik et al. 2009] is a novel parameter control interface that combines advantages of physical tangibility and adaptable graphical representations. The interface consists of a circular touchpad and a correspondingly shaped on-screen menu to choose from several adjustable parameters. The parameter set is arranged as segments of a circle framing the selected object (fig 1). Tapping into the corresponding touchpad zone activates the respective parameter for adjustments (fig 2a). Values can be directly manipulated through subsequent circular motion along the rim of the touchpad (fig 2b). During the adjustment of a selected parameter, the touchpad areas corresponding to other elements of the graphical representation can be passed by the finger without activating another parameter. Thus the parameter range can be mapped to 360 degrees or even to multiple physical rotations if more precision is required.



(a) *Parameter activation.*



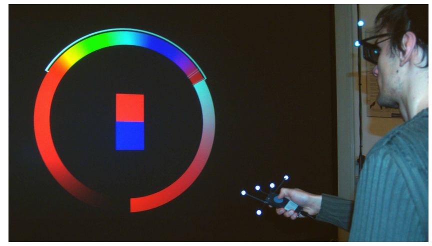
(b) *Parameter adjustment.*

Figure 2: *The Pie Slider for specifying a color in HSV color space.*

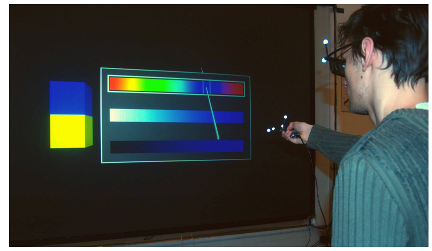
4 User Study on a Color Adjustment Task

We implemented an HSV (Hue, Saturation, Value) color adjustment task, similar to the one presented by [Kulik et al. 2009], to analyze the usability and performance of the Pie Slider (Pie Slider condition) in comparison to ray-based interaction with linear sliders (3D pointing condition). Users were asked to match the color of a displayed cube to the color of a subjacent one (fig 3).

In the Pie Slider condition, the screen displayed the three HSV controls as equally distributed ring segments around both cubes with *hue* to the top, *saturation* to the lower right and *value* assigned to the lower left sector (fig 3a). For the 3D pointing condition, the controls were vertically stacked above each other right beside the color cubes with *hue* to the top, *saturation* in the middle and *value* to the bottom (fig 3b). Each slider indicated the current setting of the respective parameter using a wiper.



(a) *The Pie Slider interface.*



(b) *Linear sliders manipulated through 3D pointing.*

Figure 3: *Slider menus for HSV color adjustment.*

Only one HSV parameter had to be adjusted per trial. The respective slider was highlighted by a surrounding white frame. The other two sliders remained operational, but accidental modifications were reset automatically after completing the action. Once a parameter had been set correctly within the given tolerance, the task was completed and the next trial automatically started.

The task structure can be subdivided into two main phases: the parameter activation and the adjustment phase. The execution times for both phases were recorded separately. Activation and adjustment times of incorrectly selected parameters were also logged.

Finger contact inside the corresponding circle sector of the touchpad activated the associated parameter in the Pie Slider condition. The selected parameter could then be adjusted through circular motion along the touchpad's rim. Lifting the finger from the touchpad completed an adjustment action. For the 3D pointing condition, a three-dimensional pick ray was used to select a slider on the screen and to adjust the corresponding parameter by dragging its wiper horizontally. The wand-mounted touchpad could be pressed down completely like an ordinary button to trigger selections. While holding this button down, the selected slider could not be lost. Horizontal movement of the ray intersection point was mapped to the position of the wiper, while vertical movement was disregarded.

In contrast to the Pie Slider condition, the 3D pointing condition provided a shortcut modality. Implicit to the selection of a slider, one could point directly to a certain parameter value on the slider. The respective wiper would then jump directly to that position. Keeping the button pressed enabled further fine-adjustments. This feature can be useful if the target value is known beforehand, as was the case in our test scenario. In practice, however, adjusting parameters is often an exploratory task considering that the target value is not known beforehand but explored through its manipulation.

The performance of the ray-based technique certainly depends on the size of the graphical interface. Thus, we decided to maximize its size in order to provide the best possible condition for 3D pointing as a reference. System control interfaces that cover almost the entire display are rarely appropriate since they would occlude the more relevant application content. We therefore included *size* as an independent variable to the study setup. *Parameter* and the color offset (*distance*) were included as further independent variables.

4.1 Experimental Setup

The study was conducted on a 95" stereoscopic rear projection setup. Users were standing about 2 m in front of the screen center operating the input device. To gain the correct perception of the pointing ray, the users had to wear tracked polarization glasses. We decided to use this immersive display setup to provide the best possible implementation of 3D pointing. The visual stimuli were displayed at a stereo depth corresponding to the screen surface.

In both conditions the input device was a 6 DOF tracked wand with a circular touchpad (28 mm diameter) mounted on top. For the 3D pointing condition, a ray was emanating from the wand. In the Pie Slider condition the wand-mounted circular touchpad was operated by the thumb. A non-linear transfer function, as known from mouse pointer acceleration in operating systems, was applied for relative motion input to adjust parameter values with the Pie Slider.

4.2 Participants

Seven female and fifteen male participants aged between 20 and 33 volunteered for this study. All but one of them were students in engineering, fine arts or humanities. None of them reported to have issues with color perception or stereoscopic vision.

4.3 Design and Procedure

After a short training to familiarize participants with the stereoscopic setup and the involved interfaces, 81 color adjustment tasks were recorded with each *technique* condition. The order was balanced between users. The 81 trials per user and *technique* sum up from 3 repetitions of 3 *parameter* × 3 *distance* × 3 *size* conditions. Three different *distance* conditions between the initial and the target values were defined to match an index of difficulty of 4, 4.5 or 5, as defined by Fitts' Law [Fitts 1954].

To ensure best training for the most difficult conditions, we tested the *size* condition in decreasing order. The linear sliders had an active width of 120 cm × 10 cm in height in the *large* condition, 60 cm × 5 cm in the *medium* condition and finally 30 cm × 2.5 cm for the *small* setup. The Pie Slider segment length always corresponded to the length of its linear counterpart through all size variations. The diameter of the on-screen representation of the Pie Slider interface thus ranged from 115 cm (*large*) to 57cm (*medium*) to 29cm (*small*). Since we maintained the size of the colored cubes, the circular sliders could not be displayed surrounding them in the *small* condition but only to their side like the linear sliders in all *size* conditions.

Each *size* condition was tested in three blocks of nine successive trials. A block consisted of 3 *distance* × 3 *parameter* conditions. The participants were encouraged to take breaks between blocks to minimize fatigue. Based on the color space of the projection setup, we predefined a set of start and target values for each parameter *hue*, *saturation* and *value* that could be easily identified by the user. These sets were randomly presented to the participants ensuring that no specific color adjustment task had to be repeated again. The tolerance level for adjustments was set to 5% of the slider length.

4.4 Hypothesis

Different ergonomic requirements of the actual study setup required some modifications of the Pie Slider technique compared to the desktop situation in [Kulik et al. 2009]. Mainly, the active touch area was reduced in size and operated by the thumb instead of the index finger. This may lead to accuracy drawbacks. Thus, we estimated slightly longer times spent on activation and adjustment compared to those obtained from the desktop study (1 s and 3-4 s).

For the 3D pointing condition, we based our hypothesis on the previous research of [Steed and Parker 2005] and [Poupyrev et al. 1998]. They recorded selection times for 3D pointing tasks ranging from 1.1 s to 3.5 s depending on target size, position in depth and the employed display technology. Comparing our setup characteristics with these findings, we could assume that slider activation might take about 2.5 s. However, pilot experiments indicated that adjusting the linear sliders through ray manipulation was faster in our case. In contrast to the cited studies, our selection targets were not appearing at random positions in the virtual environment, but at a fixed position at the center of the screen. This considerably facilitated successive trials. Proprioceptive cues allowed for quasi blind pointing and thus to keep the visual focus on the object of interest even though the graphical interface was placed next to it. Compared to relative motion input for slider adjustments with the Pie Slider, the proprioceptive cues of absolute and direct pointing should also speed up the parameter manipulation. However, in the case of the 3D pointing technique, performance also depends largely on the target size. From these considerations we derived two hypotheses:

- H1: Ray-based interaction will be faster than the Pie Slider in the large-sized GUI condition.
- H2: The Pie Slider will be faster than 3D pointing in the small-sized GUI condition.

4.5 Results and Discussion

Data was collapsed and entered into a 2 (technique) x 3 (parameter) x 3 (distance) x 3 (size) analysis of variance with order of *technique* as between-subjects factor. Bonferroni adjustment of α was used for post-hoc comparisons. Order of *technique* produced neither a main nor an interaction effect. We found significant main effects on task completion times (TCT) for *parameter* ($F(2, 40) = 20.55$, $p < .01$), *distance* ($F(2, 40) = 14.24$, $p < .01$) and *size* ($F(2, 40) = 27.3$, $p < .01$) as well as interaction effects for *technique* × *size* ($F(1, 20) = 35.9$, $p < .01$) and *technique* × *distance* ($F(2, 40) = 6.8$, $p < .01$).

The overall task completion times were 5.64 s in the Pie Slider condition and 5.68 s for 3D pointing. A closer look at the task phases (fig 4) showed that with the Pie Slider, substantial more time was spent on the adjustment operation (4 s) than on the parameter activation (1.2 s). The activation time using 3D pointing took almost twice as long (2.07 s) as its Pie Slider counterpart. But for parameter adjustment, 3D pointing demonstrated advantages (2.77 s). Time spent on mistakenly activated sliders was comparable between both conditions (*Pie Slider*: 0.85 s, *3D pointing*: 0.88 s).

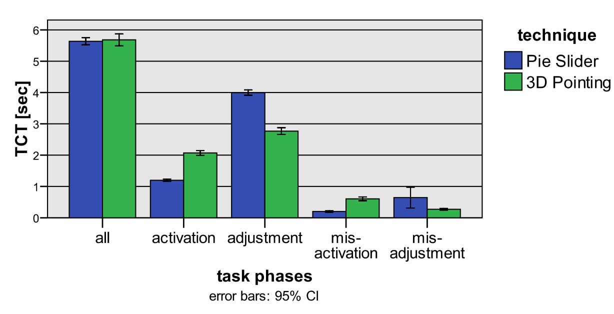


Figure 4: *Task phases per menu technique.*

We found significant differences regarding color parameters. Adjusting *hue* required the longest time (6.22 s), followed by *value* (5.43 s) and *saturation* (5.32 s). Post-hoc comparisons showed that only *hue* differs significantly from both other parameters.

Size had the expected effect on task completion times. From the *large* to the *small* condition, TCTs increased from 5.21 s over 5.28 s to 6.5 s. As expected, we found an interaction between *technique* and *size*. The results indicated that task performance with the Pie Slider is less dependent on *size* than with 3D pointing (fig 5). TCTs in the 3D pointing condition followed the overall trend. Post-hoc comparisons with Tukey revealed significant differences between all *size* conditions (*small*: 7.44 s *medium*: 5.1 s and *large*: 4.5 s, all $p < .05$). In the Pie Slider condition, instead we see an inverse relationship between performance and size. Here the longest operation time (5.92 s) was found for the *large* sized visual interface, while not much difference can be observed between the *medium* (5.45 s) and the *small* condition (5.55 s). The difference between the *large* and both other conditions was found to be significant in a post-hoc Tukey test($p < .05$). This odd effect can be explained by learning. Recall that all participants started with the *large* size condition.

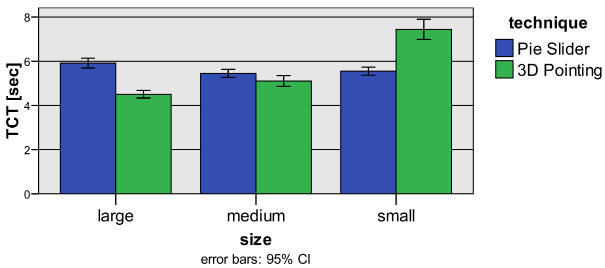


Figure 5: *Menu - Size interaction.*

We also independently compared *technique* for the three *size* conditions. The differences between both techniques in the *small* as well as in the *large* condition were significant (both $p < .01$), which confirms H1 and H2. Subjective ratings of the participants of our study validated this result. Thus, not only is tangible interaction with the Pie Slider very efficient, but direct pointing in 3D space on 2D interface elements can be as well. If the spatial relation to the pointing target remains fixed and the targets are relatively large, the users can benefit from proprioceptive cues. Unfortunately, such an ideal situation rarely occurs when using common 2D GUI elements in immersive large screen environments. For 3D pointing-based interfaces, we suggest to place virtual control interfaces in large screen setups always in the same relative position to the user to get proprioceptive support for selection and manipulation.

Significant main effects on *distance* show an overall dependency of TCT on the distance between start and target color values. The interaction of *distance* with *technique* reveals that the overall effect is only based on the *distance* dependency of the Pie Slider. While the TCTs in the 3D pointing condition did not vary much from the *short* (5.73 s) over the *medium* (5.48 s) to the *large* (5.84 s) distance, the TCTs in the Pie Slider condition increased with larger distances (5.12 s, 5.52 s and 6.27 s). This indicates that the Pie Slider interface is more strongly affected by adjustment distances than 3D pointing. We see two potential reasons: first, the possibility to shortcut the adjustment distance by direct pointing to the intended color value and second, the general rapidity of 3D pointing. Minimal tilting of the pointing device results in rapid motion of the ray intersection point on the graphical slider controls. With the Pie Slider, instead, quite long finger movements have to be performed to cover large adjustment distances. Note that this also enables higher precision.

5 Conclusions and Future Work

Common workflows for immersive virtual environments often involve spatial manipulation of scene content and the modification of object related attributes in a concerted fashion. 3D pointing enables sophisticated target selection as well as basic spatial manipulation, but lacks precision as required for the operation of graphical menu controls. Furthermore, the cognitive model of spatial manipulation operating multiple DOF and 2D interaction with menus and widgets differs quite a bit. We suggest using an appropriate controller for each interaction category.

Our user study investigated how a graphical interface controlled by a thumb-operated touchpad integrated in a tracked wand performs in comparison to ray-based system control. The results show that 3D pointing delivers excellent interaction performance and outperformed the touchpad controller for very large graphical interface representations. 3D pointing benefits from proprioception, which facilitates the coarse selection of items placed in a fixed relative position to the user. On the other hand, our results also demonstrate that direct ray selection is very sensitive to the size of the visual representation. Using the touchpad on the wand, the tangible and visual representation of the interface are separated from each other and thus reducing the size of the graphical interface has no negative impact on its performance.

Our results verify that it is worthwhile to equip 3D wands with additional 2D controllers for system control tasks. Now we have to look at task sequences in various application areas (e. g. 3D modeling) wherein spatial and menu interaction alternate reasonably often. We expect that the benefit of our concept will be even more visible in such real-world situations, considering that users do not need to move the ray off the manipulated object during menu tasks.

The possibility to assign commands to selected objects is a major advantage of computer applications. Therefore, we argue that adequate 3D computer interfaces should facilitate the concerted operation of 3D selection and system control tasks. We have shown that the integration of dedicated 2D controllers into 3D pointing devices is a promising approach in that direction.

References

BOLT, R. A. 1980. "put-that-there": Voice and gesture at the graphics interface. In *SIGGRAPH '80: Proceedings of the 7th annual conference on Computer graphics and interactive techniques*, ACM, New York, NY, USA, 262–270.

DACHSELT, R., AND HÜBNER, A. 2007. Three-dimensional menus: A survey and taxonomy. *Computers & Graphics 31*, 1, 53–65.

FITTS, P. 1954. The Information Capacity of the Human Motor System in Controlling the Amplitude of Movement. In *Journal of Experimental Psychology 47*, 381–391.

KULIK, A., KUNERT, A., LUX, C., AND FRÖHLICH, B. 2009. The pie slider: Combining advantages of the real and the virtual space. In *Proceedings of the 9th International Symposium on Smart Graphics*, Springer Berlin / Heidelberg, 93–104.

MCMAHAN, R., AND BOWMAN, D. 2007. An empirical comparison of task sequences for immersive virtual environments. In *Proceedings of the IEEE Symposium on 3D User Interfaces*, 25–32.

POUPYREV, I., WEGHORST, S., BILLINGHURST, M., AND ICHIKAWA, T. 1998. Egocentric object manipulation in virtual environments: Empirical evaluation of interaction techniques. *Computer Graphics Forum 17*, 3, 41–52.

STEED, A., AND PARKER, C. 2005. Evaluating effectiveness of interaction techniques across immersive virtual environmental systems. *Presence: Teleoper. Virtual Environ. 14*, 5, 511–527.