Designing Gaze-based User Interfaces for Steering in Virtual Environments

Sophie Stellmach[*] and Raimund Dachselt[†]
User Interface & Software Engineering Group
Faculty of Computer Science
University of Magdeburg, Germany

Abstract

Since eye gaze may serve as an efficient and natural input for steering in virtual 3D scenes, we investigate the design of eye gaze steering user interfaces (UIs) in this paper. We discuss design considerations and propose design alternatives based on two selected steering approaches differing in input condition (*discrete* vs. *continuous*) and velocity selection (*constant* vs. *gradient-based*). The proposed UIs have been iteratively advanced based on two user studies with twelve participants each. In particular, the combination of *continuous* and *gradient-based* input shows a high potential, because it allows for gradually changing the moving speed and direction depending on a user's point-of-regard. This has the advantage of reducing overshooting problems and dwell-time activations. We also investigate *discrete constant* input for which virtual buttons are toggled using gaze dwelling. As an alternative, we propose the *Sticky Gaze Pointer* as a more flexible way of discrete input.

CR Categories: H.5.2 [Information Interfaces and Presentation]: User Interfaces - Evaluation/methodology—Input devices and strategies;

Keywords: Gaze input, 3D interaction, traveling, steering, virtual environments

1 Introduction

Three-dimensional (3D) virtual environments (VEs) play an increasingly important role in more and more application areas, such as medicine, 3D gaming, social networking environments (e.g., Second Life), as well as virtual interactive training. Eye movements may serve as an effective and efficient modality for interacting with such environments [Tanriverdi and Jacob 2000; Istance et al. 2008]. This is especially the case if other input modalities are not available due to constrained movement (e.g., motor impaired users) or occupied hands (e.g., machine operators or surgeons).

So far, most of the work on gaze control has focused on the interaction with two-dimensional (2D) interfaces, such as gaze typing (e.g., [Majaranta and Räihä 2002; Ashtiani and MacKenzie 2010]) and the interaction with other 2D interface elements (e.g., [Špakov and Miniotas 2005]). Only few research is devoted to the investigation of more natural gaze-based interaction techniques for 3D VEs until now (e.g., [Tanriverdi and Jacob 2000; Smith and Graham 2006; Istance et al. 2008]). This is the case even though the human gaze may serve as a fast and natural input channel for

working with digital systems and may actually be faster than traditional selection devices [Ware and Mikaelian 1987; Jacob 1990]. However, to take advantage of these benefits, particular challenges for gaze-based user interfaces (UI) have to be taken into account, such as inherent inaccuracies and the *Midas Touch* problem (i.e., the involuntary activation of events with our gaze [Jacob 1993]). This is related to the common attempt to substitute point-and-click mouse interaction with gaze input. If the *click* event is not handled effectively, an inappropriate activation is performed causing the Midas Touch effect. With respect to gaze-only interaction, typical ways for counteracting this problem are *blinking* (e.g., [Ashtiani and MacKenzie 2010]) or *dwelling* (e.g., [Jacob 1990]) to confirm a command. Although dwell-based activation is perceived as more natural than blinking [Jacob 1993], it slows down how fast a user can issue commands. Finally, additional design considerations include how to map various functionalities appropriately to a limited set of gaze-based controls, how to cope with inaccuracies, and how to design suitable eye gaze interfaces [Vickers et al. 2008].

This motivated a deeper investigation of eye gaze interfaces for 3D VEs. In this paper, we deliberately focus on how our gaze can be used for steering in such environments, as exploration is a fundamental interaction task in 3D VEs. However, we envision that the developed techniques should later be combinable with additional interaction tasks, for example, for object selection and manipulation, either via gaze-based controls or by integrating additional input modalities such as a handheld for confirming actions (e.g., [Stellmach and Dachselt 2012b]). In this paper, we investigate superimposed 2D gaze UIs to steer through a virtual scene. We discuss design considerations and the iterative design process of gaze-based UIs for two selected steering approaches differing in velocity selection and input condition. For one technique, a steering command (e.g., move forward) is issued if fixating a virtual button (similar for example to [Vickers et al. 2008]). The other technique allows for continuously and gradually changing the movement direction and speed based on the user's point-of-regard. This may ease gaze-based steering in 3D VEs greatly by reducing overshooting problems (as for example reported by Vickers et al. [2008]), and the need for dwell-time-based activations. These are often described as unnatural (e.g., [Hansen et al. 2008]), because users are forced to deliberately wait for an action to be issued. Our developed UIs have been iteratively advanced based on quantitative and especially insightful user feedback from two user studies with 12 participants each that are reported in this paper. In a nutshell, this paper contributes to an in-depth investigation of gaze-based steering tasks in virtual 3D scenes using superimposed 2D eye gaze UIs. This includes a careful and iterative development of different gaze steering approaches including their thorough evaluation.

The remaining paper is structured as follows. First, related work is described on how eye gaze has been previously used to steer in 3D VEs. In Section 3, we discuss design considerations for 2D gaze steering UIs on the basis of two proposed UI designs. Subsequently, a user study comparing these UIs is described in Section 4. Based on the obtained results, further advancements of the UIs are described in Section 5. These techniques have been evaluated in a follow-up user study described in Section 6. We conclude this paper with a discussion of the developed gaze-based steering UIs.

---
[*]e-mail:stellmach@acm.org
[†]e-mail:dachselt@acm.org

2 Related Work

In context of gaze-based selection of graphical UI (GUI) elements, gaze-directed pointing has been investigated extensively over the years aiming for a fast, natural, and effortless input modality (e.g., [Jacob 1990; Zhai et al. 1999; Cournia et al. 2003; Kumar et al. 2007]). Pointing can be efficiently accomplished via gaze, since gaze often precedes a manual action [Jacob 1990]. However, when developing gaze-based interaction techniques, the inherent lack of precision of gaze tracking (e.g., caused by involuntary eye movements and peripheral vision) has to be considered. As a result, interface elements usually need to be larger for comfortable gaze-based selections compared to the mouse. This may, however, be compensated with adapted zooming or magnification techniques (e.g., [Stellmach and Dachselt 2012b]) or by using additional input devices for more precise selections (e.g., [Zhai et al. 1999]).

Different gaze-only techniques for interacting with 3D VEs via 2D interface elements have been presented in recent years (e.g., [Istance et al. 2008; Vickers et al. 2008; Istance et al. 2009; Istance et al. 2010]. A locomotion mode for rotating and moving an avatar in a virtual 3D world (e.g., World of Warcraft) is described using discrete *active regions* for issuing a sequence of keystroke events [Vickers et al. 2008]. However, these regions are not visible during use, which may increase the cognitive workload for remembering the location and associated tasks of the different regions. In addition, unintentional movements were performed, if accidentally passing over an active region. Nevertheless, Vickers et al. [2008] report that with a suitable speed of movement, gaze-based locomotion showed promising results compared to keyboard and mouse input. In addition, they report the use of pie menus for which a first dwell represents a right button click and a second dwell issues a left button click at the viewed position. However, hitting the small control objects proved to be highly error-prone.

Snap Clutch [Istance et al. 2008; Vickers et al. 2008] is a gaze-based mouse emulator which provides different modes of operation, such as 'Locomotion' to steer in a 3D VE. Visual feedback about the current mode is indicated by the cursor's shape and color. While this approach reduces the *Midas Touch* problem, moving through a 3D VE was error-prone due to the difficulty to quickly change between modes which caused overshooting problems. To counteract this, gaze gestures (i.e., off-screen glances) are used to quickly switch between additional modes. Among others, this includes a normal 'Dwell click' mode and a 'Park it here' mode, which allows leaving the cursor at a position on the UI (e.g., a control panel) after a dwell. They indicate a high potential for the 'Park it here' technique referring to positive user feedback and performance results from a user study with, however, only two participants.

Mine [1995] describes *gaze-directed flying* for which the user flies in whatever direction she is currently looking. Similarly, Smith and Graham [2006] use eye gaze input to substitute mouse control in a First-Person Shooter game. Users could control the orientation of the camera view with their eyes, e.g., by looking at the upper region of the screen, the view rotated up. The rotation speed for both conditions was fixed. Users could move the camera by using the 'W' (forward), 'A' (left), 'S' (backward), and 'D' (right) buttons on the keyboard. Both Mine [1995] and Smith and Graham [2006] report a *Midas Touch* effect, because the camera always moved towards viewed objects, although users may have wanted to proceed in another direction. Thus, users are not able to freely look around while moving toward a target. The *Midas Touch* problem can, however, be counteracted by a dwell-time activation. Nacke et al. [2009] present a user study for a similar gaze-based steering approach in a First-Person Shooter resulting in very positive game experiences with particular high immersion levels. This indicates that adapting

the camera orientation with the gaze may serve as a natural interaction technique.

Finally, with respect to steering in a 3D VE, gaze-based pan and zoom are also of interest. For example, Adams et al. [2008] and Hansen et al. [2008] use a distinct central zoom region surrounded by a pan region towards the screen border. A continuous movement is performed when looking in the respective region. Another approach is to use the entire screen space for panning for which the panning speed depends on the distance between screen center and current point-of-regard (PoR) (e.g., [Zhu et al. 2011; Stellmach and Dachselt 2012a]).

In a nutshell, gaze-based interaction is still error-prone and requires a deeper investigation for the design of eye gaze interfaces. The *Midas Touch* issue poses one of the major problems of gaze-based interaction, which can be overcome by several means, for example, by a dwell-based activation. Although users may be slower issuing commands with dwell activations, these are perceived as more natural than for example eye blinks [Jacob 1993]. Other solutions include gaze gestures [Istance et al. 2010] and multimodal combinations [Castellina and Corno 2008; Stellmach et al. 2011; Stellmach and Dachselt 2012a]. Furthermore, using the gaze to steer in a 3D VE may cause overshooting problems or performing wrong movements. It may, however, contribute to a higher immersion and positive user experiences.

3 Design of Eye Gaze User Interfaces for Steering in a Virtual Environment

With respect to the design of eye gaze steering UIs, we first discuss basic aspects for steering in a virtual 3D scene and address these for our development. Secondly, our proposed steering approaches are discussed along with specific design decisions. For steering or traveling in a virtual 3D scene, we distinguish two main types of movement: *translation* (i.e., moving forward, backward or step to the left or right) and *rotation* (i.e., turn view up, down, left, or right).

3.1 Basic Design Decisions

To define a user's movement, Mine [1995] points out that two key parameters must be specified: *direction* and *velocity of motion*. Bowman et al. [1997] extend these and additionally consider a third high-level component: *input condition*. These three components encompass a range of attributes, from which we chose a subset for further investigation that we have addressed in the following way:

Direction: For our development, we decided to control the direction of motion by gazing at 2D UIs superimposed over a 3D VE. This is inspired by work from Istance et al. [2008; 2009], describing gaze-only steering techniques in World of Warcraft and Second Life using superimposed 2D gaze UIs and *active regions*.

Velocity: For the rate of movement, we concentrate on two types: *constant* and *rate-based*. A rate-based velocity allows a user to gradually adapt the speed of movement, for example, depending on the gaze position within a specific region. Since different gradients can be used to describe the velocity distribution, we refer to this type of technique as 'gradient-based' throughout this paper.

Input Condition: In this paper, we focus on *continuous* and *discrete* input. For the *continuous input*, as soon as an interface element (e.g., an *active region*) is looked at, an associated event is (continuously) issued. Once an area is not viewed anymore, the movement stops, whereas with *discrete input*, an action can be switched on or off by issuing a discrete event (e.g., *start moving forward*), for example, after gaze dwelling on a button-like UI.
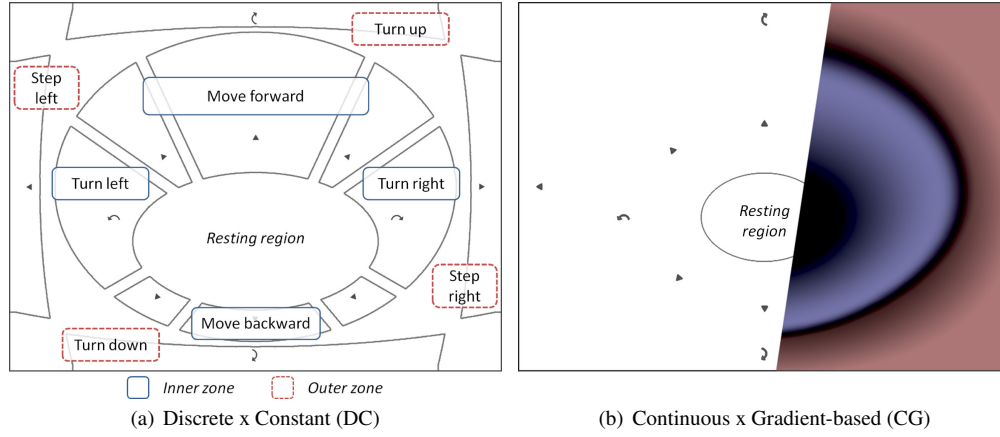
(a) Discrete x Constant (DC) (b) Continuous x Gradient-based (CG)

Figure 1: *Overview of the proposed gaze UIs and textural annotations for explanation purposes. The underlying (invisible) interaction mask for CG is shown (right).*

Based on these specifications, we investigated two particular combinations of input conditions and velocity types: **Discrete and Constant (DC)** and **Continuous and Gradient-based (CG)**. First, the combination of *discrete* and *constant* input is interesting, because a user can freely observe a 3D VE after a certain movement has been started. Thus, this technique can help to decrease the distraction of a user's gaze from an actual point-of-interest to a UI element that needs to be fixated to issue a certain action. Secondly, the *continuous* and *gradient-based* gaze input is interesting, as it may ease performing small movements by gradually adapting the velocity. So, users can quickly change the rate and direction of movement based on their gaze positions on the screen.

Furthermore, in contrast to the work from Vickers et al. [2008] and Istance et al. [2009], who only use four steering tasks (i.e., move forward, backward, turn left and right), we aim for more freedom of movement by supporting four **rotation tasks** (i.e., turn left, right, up, and down) and four **translation tasks** (i.e., move forward, backward, left, and right).

3.2 Design of the DC and CG User Interfaces

We iteratively elaborated several gaze UI designs for the specified input types DC and CG (see Figure 1). First, specific design considerations and decisions for *both* are discussed, before the individual steering UIs are described in more detail.

With respect to the **functionality mapping**, we assigned the most frequently used steering commands (i.e., move forward and backward, turn left and right) close to the center of the screen. The remaining tasks (i.e., step left and right, turn up and down) were mapped to UI elements close to the screen border. Thus, we distinguish two interaction sets: an *inner* and an *outer zone* (see Figure 1). In addition, similar to Istance et al. [2009], we *combine actions* in the *inner zone* to move diagonally by issuing, for example, 'move forward' and 'turn left' at the same time. Furthermore, to reduce the risk of overwhelming users with the gaze input, a *resting region* is integrated in which users can reside with their gaze without issuing any commands. We define a resting region at the screen center that is deliberately larger than necessary, because we considered a possible integration of additional gaze-based selection and manipulation techniques for future work. Figure 1 shows an overview of how we distributed the steering commands to particular regions for both DC and CG.

Concerning the **position** and **size** of the gaze GUIs, inherent eye tracking inaccuracies have to be taken into account. Therefore, our UI extends over the entire screen space (*space-filling*) to allow for bigger interface elements to help counteracting imprecise gaze estimations. However, this may lead to a cluttered view. Thus, we decided to use an unobtrusive semi-transparent design. Small icons are displayed to indicate which tasks are associated with the respective UI regions (see Figure 1) and to act as visual anchors inspired by the focus points described by Kumar et al. [2007]. Finally, the designs are border-oriented and the gaze cursor cannot move beyond the screen borders. Thus, we assume an indefinite virtual extension and decided to make the outer regions smaller. The sizes of individual interface elements relate to the importance of an associated task (i.e., frequency of use). Therefore, a relatively large active region is used for forward movement and a comparatively small one for moving backwards (similar to [Vickers et al. 2008; Istance et al. 2009]). In the following, we go into detail about specific design decisions for the proposed DC and CG approaches.

Discrete and Constant UI (DC). The first design type uses a *discrete* input with a *constant* rate of movement. Distinct UI regions are associated with a particular steering direction and (constant) speed, as illustrated in Figure 1(a). Users can perform a movement in the virtual scene by gaze dwelling at the associated interface element (dwell-based clicking). An activated region is displayed slightly more opaque to provide visual feedback. The movement can be stopped by dwelling at the activated UI again or by directly activating a new motion by dwelling at a different UI element. Additional active regions have been included for *combined* tasks (e.g., move forward and turn left). Furthermore, the distinct active regions need to have sufficient distances between each other to prevent issuing commands unintentionally.

Continuous and Gradient-based UI (CG). The second design uses a *continuous* input and a *gradient-based* velocity selection (see Figure 1 (b)). While no movement is executed if looking within the resting region, after dwelling at the (invisible) inner or outer zones the associated steering commands are continuously issued. Technically, the speed of movement depends on an underlying gradient mask (not shown to the user) that defines the velocity based on a brightness distribution (see Figure 1(b)). For example, a slow forward movement will be executed, if a user looks only slightly above the resting region. For the gradient-based design, the active regions need to be bigger than for the discrete input to provide a better interaction with the underlying gradient mask. In order to reduce time-
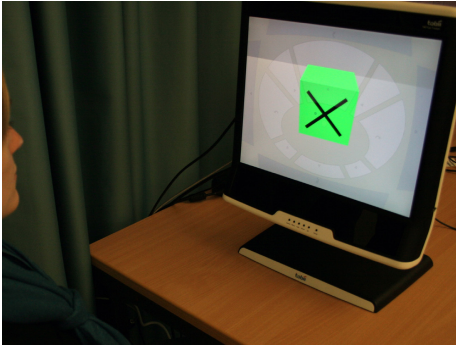
Figure 2: *Test setup with a virtual 3D scene overlayed with DC.*

Figure 3: *The scene overviews as presented to the participants.*

consuming dwell-time activations, we propose to gradually merge the discrete steering commands within a zone (inner and outer). For example, a smooth transition between 'moving forward' and 'turning to the side' can be achieved in the inner zone, which is similar to work from [Smith and Graham 2006] and [Nacke et al. 2009]. Thus, the dwell-time-based activation is only required to 'activate' the inner or outer zone (only one zone can be active at a time). This way, the gaze can move from the resting region to the outer zone (crossing over the inner zone) without unintentionally issuing any actions. Once a zone is activated, movement commands can be continuously issued within this zone without any dwell-time interruptions (e.g., first moving forward, turning a bit to the left while still moving forward). When a currently activated zone is not looked at anymore, the zone has to be activated via dwell-time again. As illustrated in Figure 1(b), we use an unobtrusive design only illustrating the resting region and displaying additional icons to indicate the mapping of steering commands to the invisible active regions.

4 User Study

To find out about the practicality, advantages and disadvantages of the developed gaze steering UIs, they are evaluated in a user study. Especially the qualitative user feedback can help to further explore and refine the proposed designs. In this respect, we assume that the gradient-based velocity selection is assessed as more natural, because users can adapt the rate of movement gradually with a lower risk of overshooting. On the other hand, we expect that a major advantage of the discrete input is to allow users to freely observe a scene while moving through it (decreased visual distraction). However, we assume it to be less natural and task efficient (with respect to the task completion times).

Participants. Twelve able-bodied university members volunteered for the study (9 male, 3 female; from 24 to 35 years of age, with normal or corrected-to-normal vision). All were familiar with 3D VEs. Only two people have used an eye tracker before.

Design. The two eye gaze interfaces with their underlying input conditions (DC and CG) act as an independent variable and were tested by all participants (within-subjects/repeated measures design) in a counterbalanced order. We did not test a mouse baseline condition, since our main goal was not to beat mouse input, but rather to investigate how eye gaze steering UIs can be enhanced.

Apparatus. For testing the proposed gaze steering UIs, Microsoft's XNA Game Studio 3.0 (based on C#) was used to implement the presented gaze interfaces and to create virtual 3D test scenes. Users could freely explore the scenes by moving their camera viewpoints with one of the described gaze steering UIs. The Tobii T60 eye tracker system, which is integrated into a 17-inch TFT color screen
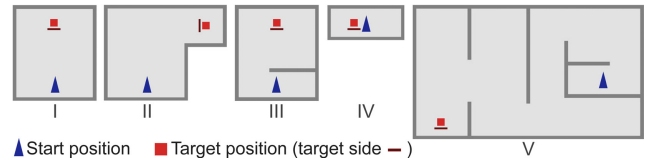
(with a resolution of 1280x1024 pixels), was used to determine participants' gaze positions on the monitor. The current gaze position is represented by a small semi-transparent circle. The gaze interaction is based on pointing at active regions by looking at them and on clicking via dwell-time activation (250 ms). The value for the constant velocity for DC is the medium velocity for CG.

Procedure. After a brief introduction, a demographic questionnaire was handed out. Each participant was seated approximately 60 cm from the eye tracker screen and was instructed to sit fairly still, but without restricting his/her movement. Before testing each design, a calibration was conducted, and participants could test the respective gaze UI until they felt sufficiently acquainted with it (usually less than two minutes). Each participant had the task to get from a given start position to a predefined target - a box with a target face indicated with an X (see Figure 2). The target cube turned green as soon as a participant reached an acceptable distance and viewing angle. Thus, the interaction included 3D translation (i.e., move forward, backward, left, and right) and rotation tasks (i.e., turning left, right, up, and down), as illustrated in Figure 1. However, a fixed height position was used to prevent users from flying through the ground or up in the air. To investigate how each gaze UI is used for different problems (e.g., to reach a goal directly in front of the user in contrast to navigating through a maze-like scene), five different combinations of start and goal positions were tested in individual scenes (see Figure 3). The scenes were tested in a fixed order with increasing difficulty. This included simply moving forward and turning the camera view downwards to look at a target on the floor (scene I). More complex tasks required, for example, small movements around a nearby object (scene IV) or steering around several corners (scene V) to reach the target. The scene overviews as presented in Figure 3 were shown to the participants one at a time. After finishing the five test scenes for a given eye gaze UI, an **intermediate questionnaire** was handed out (described below) before continuing with the next UI design.

Measures. Our quantitative measures included logged task completion times, camera positions, and gaze data. To better assess strengths and weaknesses of our gaze steering approaches, we put a particular emphasis on substantial user feedback gained from several *intermediate* and a *final questionnaires*. The **intermediate questionnaire** contained 19 quantitative and 2 qualitative questions. For the quantitative part, participants should rate different statements about eight usability aspects (summarized in Figure 4) based on 5-point Likert scales from *1 - Do not agree at all* to *5 - Completely agree*. For this, on the one hand, we considered aspects promoting an effective traveling technique [Bowman et al. 1997]: 1. *Speed*, 2. *Accuracy* of movement, 3. *Ease of learning*, 4. *Ease of use*, and 5. *Spatial awareness*. On the other hand, we took the following additional issues into account: *Intuitiveness*, *Visual distraction*, and *Task-driven use*. At the end of the study, a **final questionnaire** with one qualitative and 16 quantitative questions was handed out, which asked participants to assess the individual designs and how the interaction felt altogether. On average, each session took about 70 minutes for the described procedure.
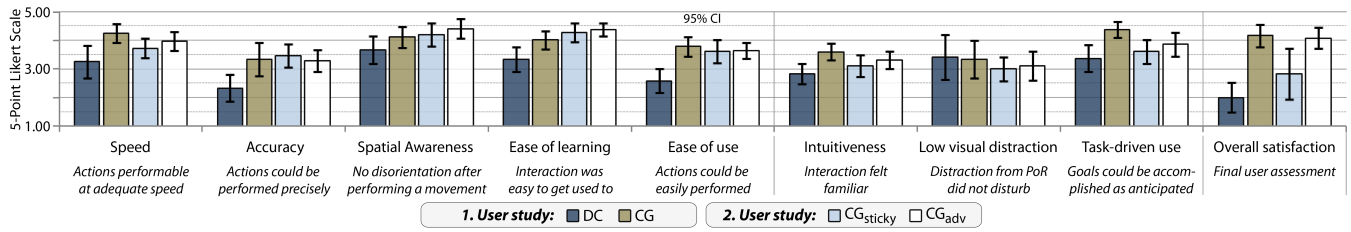
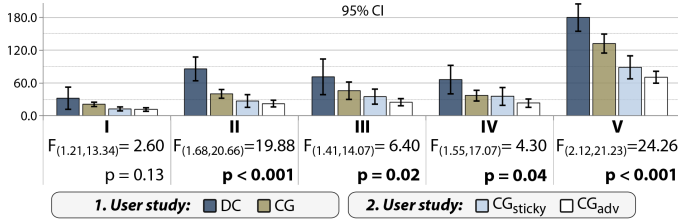**Figure 4:** *Overview of the quantitative results from the questionnaires with ANOVA and significance values.*

Figure 5: *The mean task completion times (in seconds) for reaching the target in the five test scenes.*

4.1 Results

In the following, results are reported for the comparison of the two eye gaze steering UIs: Discrete and Constant (DC) and Continuous and Gradient-based (CG). Paired samples t-tests have been used to compare task completion times and quantitative questionnaire results for the two conditions. We put a particular emphasis on qualitative user feedback to further improve the proposed design alternatives.

Task Performance. All participants were able to reach the target in the five scenes with each UI. Due to the complexity of degrees of freedom for moving through a 3D scene, we refrain from assessing the number and type of errors quantitatively. However, problems that participants encountered are discussed in Section 4.1.1. The task performance was investigated based on the mean task completion times for the five tested scenes. Users were faster with CG than with DC in all scenes, as illustrated in Figure 5 (two left bars of each diagram). Participants needed significantly longer to reach a target with DC in scene II (t(11)=3.75, p=0.003), scene III (t(11)=2.42, p=0.034), and scene V (t(11)=3.42, p=0.006). No significant performance differences could be identified for scene I and IV. Furthermore, moving around a nearby target was generally rated as difficult in the questionnaire (Mean value (M) = 3.92, Standard Deviation (SD) = 0.64). This is also reflected in longer task completion times, as shown for scene IV (see Figure 5), which required several small movements to reach the target. This was particularly challenging with the DC design because of overshooting problems. While this problem was reduced for the CG input, users explained that they kept looking back at the target more frequently the closer they got, which made movement control difficult for the continuous input. In contrast, participants disagreed that moving around or towards a distant object was difficult (M=1.75, SD=0.72).

User-ranked Usability. Participants could rate eight usability aspects for each design based on different statements in the *intermediate questionnaire* (see section *Measures*). Figure 4 provides an overview of these assessments and the overall satisfaction from the *final questionnaire* with their corresponding mean values and 95% confidence intervals (CI). The CG design was assessed very positively and was ranked better than DC for all aspects, except for *visual distraction*, for which, however, both only received moder-

ate ratings (DC: M=3.42, SD=1.38; CG: M=3.33, SD=1.18). Thus, this shows that the assumed major advantage of DC concerning a decreased visual distraction could not be confirmed. In particular, the CG design was assessed as *easy to learn* (M=4.02, SD=0.59) and that goals could be achieved as anticipated (*task-driven use*) (M=4.38, SD= 0.53). All in all, gaze-based interaction in this experiment was rated as rather slow (*Speed*) and imprecise (*Accuracy*). Especially for the DC design, participants had difficulties to perform actions precisely (M=1.33, SD=0.89). One reason given by participants were time-consuming dwell activations. In the final questionnaire, participants stated that they liked CG (M=4.17, SD=0.69) significantly better than DC (M=2.00, SD= 0.91).

4.1.1 Qualitative User Feedback and Discussion

A comprehensive overview of frequently reported advantages and disadvantages for the eye gaze steering UIs are summarized in Figure 6. In general, the designs were conceived as clearly structured, unobtrusive, and simple. The visual feedback for the currently active regions for the DC design have been positively mentioned. On the other hand, it was not found particularly useful to vary the size of interface elements according to the importance of an associated steering command. All in all, participants preferred the CG UI over DC (see also Figure 4 - 'overall satisfaction'). Participants explained their assessment with the reduced need for dwell-time-based activations for the CG UI. Please remember that dwell-time activations were only required for switching between the inner and outer zone in the CG UI. In addition, five participants particularly liked that different moving directions could be seamlessly controlled in the inner zone. Some participants asked for a more adaptive user interface for which they could change the size of interface elements and assign interaction tasks differently. In the following, participants' comments about the individual eye gaze designs are discussed in more detail.

Discrete & Constant. The *constant* speed showed to be too inflexible and was assessed as not particularly useful (M=3.00, SD=0.91). For long distances the speed was described as too slow and for small movements as too fast. The *discrete* input was generally criticized (M=2.17, SD=0.99). This was mainly due to eye tracking inaccuracies which caused involuntary activations or the inability to stop in time. Thus, participants wished that the movement should stop as soon as leaving an active region (*continuous input*). Nevertheless, one participant remarked: *Actually I find it quite relaxing, because I don't need to constantly look somewhere else to move.* Besides overshooting problems, dwell-time activation was negatively mentioned as it was described as time-consuming and inefficient. Several participants had problems dwelling at a certain location without *wandering off* with their gaze, because they became distracted by the underlying moving 3D scene. With respect to the DC design participants suggested the possibility to activate different regions at the same time and to integrate gradient-based speed adjustments.

Continuous & Gradient-based. The CG design was assessed very positively by participants. On the one hand, the continuous

Figure 6: *An overview of the most frequently reported advantages and disadvantages for the DC and CG designs.*

gradient-based design made it possible to decrease *overshooting* and *wandering off* problems. On the other hand, users described the interaction with the CG design as more natural and comfortable. While in particular the gradient-based velocity control was perceived very positively (M=4.25, SD=0.75), it was complained that precise speed selections were difficult to achieve. This could be improved with a better overview about how the speed values are distributed in the gradient-based design. Even though the use of dwell-time was reduced in this design, it was still described as disturbing, especially after leaving a zone accidentally.

5 Revised Designs

Based on the results from the user study, we discuss implications for revising the presented designs in this section. First, we describe an enhanced CG design. Secondly, we propose an alternative DC input which we call *Sticky Gaze Pointer* that works in concert with the advanced CG design.

5.1 Advanced Continuous & Gradient-based UI

While the CG UI has already been assessed positively, its design can be further improved. To facilitate more precise movements, it should be clearer how different steering directions and velocity values are distributed. For this purpose, we revised the CG UI to show additional visual cues such as a subtle velocity scale indicated by a few additional icons (see Figure 7). In addition, the layout of the inner and outer zones are marked more clearly. Figure 7 shows the refined design which we refer to as *advanced CG* (CG_{adv}) in the following. Furthermore, the CG UI has been revised with respect to the position, size, and distances of interface elements. The border-oriented layout did not prove useful, because eye tracking accuracy degraded towards the screen borders. Thus, despite our anticipation that interface elements at the screen borders can be made smaller, this was impeded by the decreasing accuracy. As a result, the *size* of interface elements in the outer zone is now increased.

5.2 Sticky Gaze Pointer

As an alternative for discrete constant input, we propose the **Sticky Gaze Pointer** (CG_{sticky}). This is inspired by the 'Park it here' technique from Istance et al. [2008] (see Section 2), which allows leaving the cursor at a position on the UI after a dwell. Instead of applying this to a detached steering control panel, we decided to investigate this approach in context with the gradient-based steering UI to allow for a more flexible way of selecting different steering directions and velocities. Thus, with the Sticky Gaze Pointer a user can *stick* (lock) the gaze pointer to a fixated location anywhere on the screen. This has the advantage that no large distinct virtual buttons are required anymore. In addition, a user can freely look around while a movement is performed that is associated with a previously fixated location in the steering UI.
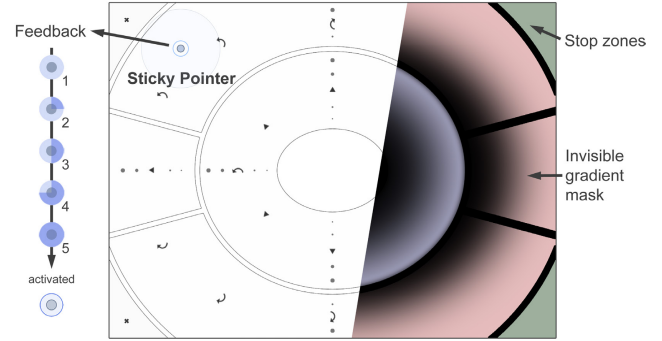

Figure 7: *Revised CG design with the Sticky Gaze Pointer.*

The procedure for activating the Sticky Gaze Pointer is as follows: after a short fixation, an icon appears at the point-of-regard and a short animation is shown until finally activating the Sticky Gaze Pointer at the respective location (see Figure 7, left row). This additional visual feedback indicates the current state of the Sticky Gaze Pointer and acts as a visual anchor to reduce *wandering off* problems. Please remember that these problems impeded the dwell-time-based activation (and deactivation) of commands for the DC UI, because users became easily distracted by the underlying moving 3D scene. A certain margin (e.g., 50 pixels) around the initiated pointer icon is used to allow for a tolerance to imprecise eye tracking data. If looking beyond this margin, the activation of the Sticky Gaze Pointer is canceled during the feedback animation.

As we have learned from the initial DC UI, more efficient means for stopping a movement are required otherwise causing overshooting problems. Therefore, we decided that users can stop a discretely activated movement (i.e., deactivate the Sticky Pointer) by different means. First, additional 'stop' zones at the screen corners are used which discontinue the movement as soon as being looked at. This means that no dwell-based confirmations are required for stopping. Secondly, a saccadic gaze movement of a certain minimal length could also stop a movement. However, the minimal length value has to be carefully chosen, as a distance too short may lead to involuntary stops, whereas a value too large may prevent stopping in time. As proposed by Istance et al. [2008], a third alternative is to dwell at the Sticky Gaze Pointer again to pick the cursor up again. All in all, the Sticky Gaze Pointer can be combined with CG_{adv} to allow for a more flexible selection of steering directions and velocities compared to the initial distinct virtual buttons.

6 Follow-up User Study

Based on the proposed improvements, we tested the two refined gaze steering variants in a follow-up user study: the advanced CG design without (CG_{adv}) and with the Sticky Pointer (CG_{sticky}) (see

Figure 7). The same *design*, *apparatus*, and *procedure* as described in Section 4 were used again. Again, twelve able-bodied university members volunteered for the study (9 male, 3 female; from 21 to 31 years of age, with normal or corrected-to-normal vision). Five participants had already taken part in the first study.

6.1 Results & Discussion

In the following, results are reported for the comparison of all discussed eye gaze designs in this paper: Discrete and Constant (DC), Continuous and Gradient-based (CG), and the advanced designs with and without the Sticky Pointer (CG_{sticky}, CG_{adv}). A repeated-measures ANOVA (Greenhouse-Geisser corrected) was used to investigate differences in task completion times and usability rankings. Post-hoc paired samples t-tests with a Bonferroni correction were used to further investigate which particular values differed significantly (pairwise comparisons). While quantitative results may indicate performance improvements, we lay an emphasis on qualitative feedback again, because this helps in better assessing the usability and further improvements for eye gaze steering UIs.

Task Performance. All participants could successfully complete all scenes with CG_{adv} and CG_{sticky}. Figure 5 shows an overview of the mean task completion times for the original and refined designs. Participants were faster with the refined designs compared to DC and CG, whereby CG_{adv} was fastest. Based on post-hoc tests, the refined designs helped to improve task completion times significantly, especially for scene V. There, the mean task completion times were significantly longer for DC compared to CG_{sticky} (p=0.02) and CG_{adv} (p<0.001) and also longer for CG compared to CG_{sticky} (p=0.04) and CG_{adv} (p=0.002). Only for scene I the four tested designs (from both studies) did not significantly differ in task completion time.

User-ranked Usability. Figure 4 provides an overview of the user-ranked usability aspects for the four UIs with the corresponding mean values and 95% confidence intervals (CI). In general, CG_{adv} was assessed better than CG_{sticky}, except with respect to the *accuracy* of interaction for which CG_{sticky} received slighty better ratings (CG_{sticky}: M=3.46, SD=0.72; CG_{adv}: M=3.29, SD=0.66). While CG_{sticky} was assessed in many aspects similar to CG_{adv}, all in all participants disliked this technique as shown by the moderate rating for the *overall satisfaction* (M=2.83, SD=1.57), although opinions highly diverged. With respect to DC, the refined designs received better ratings except for *visual distraction*. Thus, again the advantage of the discrete input for a decreased visual distraction could not be confirmed. Participants found DC particularly more difficult to learn, to use and to achieve anticipated actions with it than with CG_{sticky} or CG_{adv}. With respect to CG, participants rated CG_{adv} similarly.

6.1.1 Qualitative User Feedback and Discussion

Based on the quantitative results, we can conclude that even though participants assessed both refined designs similarly positive to the initial CG UI, the task performances could be significantly improved especially for larger scenes with the enhanced designs. Four participants emphasized the well-designed UI with respect to its transparency, clear icons, and neutral colors. In addition, the smooth transition between moving forward and turning sideways in the inner zone was positively mentioned by six participants again.

Sticky Pointer. Although participants generally liked the idea of the Sticky Pointer, the tested basic scenes were too small to benefit from it (mentioned by four participants). However, they could imagine to use it to steer in large 3D VEs. In addition, three participants explained that the activation and deactivation of the Sticky

Pointer was too inflexible. For some occasions, it took too long to activate, and for others it was too fast and with that became disturbing. In this context, participants mentioned that CG_{adv} gave them a higher feeling of control. Furthermore, general problems for gaze-based interaction were again encountered, including unreliable tracking and inaccurate gaze estimations.

Advanced CG UI. In general, the *continuous gradient-based* input allowed for a more precise movement (less overshooting) and was assessed as easy to learn and to use. However, a remaining problem for the CG UIs was the higher degree of visual distraction, because the user had to look away from an object of interest to steer in the 3D VE. Although this was not striking for the tested basic scenes, it could affect the task performance for visually rich 3D virtual worlds and should be further investigated in future studies. In this context, a revised Sticky Gaze Pointer may aid in counteracting this problem.

Improvements. Participants proposed several improvements concerning the designs. First, the distances between inner and outer zones should be further increased. Moreover, it was proposed to reduce the gaze-based control to one zone, while the other one should be controlled via an additional device, for example, a handheld device (see also [Stellmach and Dachselt 2012a]). This would also allow for easily switching to an additional free (undisturbed) exploration mode which could then be used for additional interaction tasks, such as object selection and manipulation. Quick mode switches could also be performed using off-screen glances, for example, as proposed by Istance et al. [Istance et al. 2008]. However, this may be less appropriate for more complex systems with a large number of different modes.

Outlook. As an outlook, alternative activation modalities such as speech or a (physical) button click could be used to confirm an action more quickly (e.g., Stiefelhagen and Yang [1997]; Castellina and Corno [2008]; Stellmach and Dachselt [2012a; 2012b]), as several participants complained about dwell-based activations. Furthermore, users should be enabled to customize the gaze-based UIs, for example, by changing the thresholds for the *dwell-time activation* (e.g., [Hansen et al. 2008]) and *speed of movement*. While users who have little experience with 3D VEs may want to proceed slowly, experts can quickly move through the environment.

Lessons learned. For future research about (superimposed) eye gaze UI design, some lessons learned include the following:

- **'Wandering off' problems**: This is the easy distraction from the underlying moving scene when trying to dwell on a semi-transparent UI element and was especially a problem for the DC UI. This can be compensated, for example, by locally decreasing transparency or adding a larger radius for performing a dwell activation.

- **Benefits of gradient-based input** for steering: It allows for higher control, works well with not overly precise gaze positions, and may reduce dwell activations for gaze-only input. Furthermore, users were able to steer to a target significantly faster especially in larger scenes compared to discrete and constant input.

- **Importance of well thought out UI designs**: The improved task performance for CG_{adv} compared to CG showed that although the same interaction principle was applied (i.e., continuous and gradient-based), a well thought out design for eye gaze interfaces is important. For particular design advice please refer to Figure 6 and Section 5 where we discussed the advancements for CG_{adv}.

7 Conclusion

This paper presented an investigation of the design of eye gaze interfaces for steering in 3D virtual environments. For this, we described the development of 2D eye gaze steering user interfaces (UIs) that mainly differ in *input condition* (discrete vs. continuous) and *velocity selection* (constant vs. gradient-based). In this context, the *continuous gradient-based* (CG) input allowed for a smooth continuous gaze-based steering by combining different motion directions and velocities within one UI, which reduced the need for time-consuming dwell activations. In an iterative design process, we evaluated different eye gaze UI designs for *discrete constant* (DC) and for CG in two user studies providing valuable insights for further improvements. After all, the CG UIs were preferred by participants and were assessed as natural, fast, and easy to use and learn. In addition, the revised CG UI resulted in the fastest task completion time among the tested designs. Based on this, our investigations suggest that eye gaze steering can benefit from a continuous gradient-based interface design. In this context, while gaze UIs can be visually unobtrusive and yet effective to use, it is important to consider special design issues, such as *wandering off* problems for dwell-based activations. Further investigations are necessary for the suitability of the proposed gaze-based steering techniques for visually rich 3D VEs and in combination with additional tasks (e.g., a search task). With the presented design considerations and the thorough evaluation of the tested designs, we provide a solid foundation for further developments of novel gaze-based steering UIs. Thereby, other application areas may also benefit from such techniques, such as using the eyes to pan and zoom through image collections (e.g., [Hansen et al. 2008; Stellmach et al. 2011]) or for remotely controlling machines (e.g., [Tall et al. 2009]).

Acknowledgements

References

ADAMS, N., WITKOWSKI, M., AND SPENCE, R. 2008. The inspection of very large images by eye-gaze control. In *Proc. AVI'08*, ACM, 111–118.

ASHTIANI, B., AND MACKENZIE, I. S. 2010. BlinkWrite2: an improved text entry method using eye blinks. In *Proc. ETRA'10*, ACM, 339–345.

BOWMAN, D. A., KOLLER, D., AND HODGES, L. F. 1997. Travel in immersive virtual environments: An evaluation of viewpoint motion control techniques. In *Proc. VRAIS '97*, IEEE Computer Society, 45.

CASTELLINA, E., AND CORNO, F. 2008. Multimodal gaze interaction in 3D virtual environments. In *Proc. COGAIN '08*, 33–37.

COURNIA, N., SMITH, J. D., AND DUCHOWSKI, A. T. 2003. Gaze-vs. handbased pointing in virtual environments. In *Proc. CHI '03*, ACM, 772–773.

HANSEN, D. W., SKOVSGAARD, H. H. T., HANSEN, J. P., AND MØLLENBACH, E. 2008. Noise tolerant selection by gaze-controlled pan and zoom in 3D. In *Proc. ETRA'08*, ACM, 205–212.

ISTANCE, H., BATES, R., HYRSKYKARI, A., AND VICKERS, S. 2008. Snap clutch, a moded approach to solving the midas touch problem. In *Proc. ETRA'08*, ACM, 221–228.

ISTANCE, H., VICKERS, S., AND HYRSKYKARI, A. 2009. Gaze-based interaction with massively multiplayer on-line games. In *Proc. CHI'09*, ACM, 4381–4386.

ISTANCE, H., HYRSKYKARI, A., IMMONEN, L., MANSIKKAMAA, S., AND VICKERS, S. 2010. Designing gaze gestures for gaming: an investigation of performance. In *Proc. ETRA'10*, ACM, 323–330.

JACOB, R. J. K. 1990. What you look at is what you get: eye movement-based interaction techniques. In *Proc. CHI '90*, ACM, 11–18.

JACOB, R. 1993. *Eye Movement-Based Human-Computer Interaction Techniques: Toward Non-Command Interfaces*. Ablex Publishing Co., ch. Advances in Human-Computer Interaction, 151–190.

KUMAR, M., PAEPCKE, A., AND WINOGRAD, T. 2007. Eyepoint: practical pointing and selection using gaze and keyboard. In *Proc. CHI'07*, ACM, 421–430.

MAJARANTA, P., AND RÄIHÄ, K.-J. 2002. Twenty years of eye typing: systems and design issues. In *Proc. ETRA'02*, ACM, 15–22.

MINE, M. 1995. Virtual environment interaction techniques. Tech. rep., Technical Report, UNC Chapel Hill CS Dept.

NACKE, L., STELLMACH, S., SASSE, D., AND LINDLEY, C. A. 2009. Gameplay experience in a gaze interaction game. In *Proc. COGAIN'09*, COGAIN Association, 49–54.

SMITH, J. D., AND GRAHAM, T. C. N. 2006. Use of eye movements for video game control. In *Proc. ACE'06*, ACM, 20.

STELLMACH, S., AND DACHSELT, R. 2012. Investigating gaze-supported multimodal pan and zoom. In *Proc. ETRA'12*, ACM.

STELLMACH, S., AND DACHSELT, R. 2012. Look & touch: Gaze-supported target acquisition. In *Proc. CHI'12*, ACM.

STELLMACH, S., STOBER, S., NÜRNBERGER, A., AND DACHSELT, R. 2011. Designing gaze-supported multimodal interactions for the exploration of large image collections. In *Proc. NGCA'11*, ACM, 1–8.

STIEFELHAGEN, R., AND YANG, J. 1997. Gaze tracking for multimodal human-computer interaction. *Acoustics, Speech, and Signal Processing, IEEE International Conference on 4*, 2617.

TALL, M., ALAPETITE, A., SAN AGUSTIN, J., SKOVSGAARD, H. H., HANSEN, J. P., HANSEN, D. W., AND MØLLENBACH, E. 2009. Gaze-controlled driving. In *Proc. CHI EA'09*, ACM, 4387–4392.

TANRIVERDI, V., AND JACOB, R. J. K. 2000. Interacting with eye movements in virtual environments. In *Proc. CHI '00*, ACM, 265–272.

VICKERS, S., ISTANCE, H., HYRSKYKARI, A., ALI, N., AND BATES, R. 2008. Keeping an eye on the game: eye gaze interaction with massively multiplayer online games and virtual communities for motor impaired users. In *Proc. ICDVRAT'08*, 159–166.

ŠPAKOV, O., AND MINIOTAS, D. 2005. Gaze-based selection of standard-size menu items. In *Proc. ICMI '05*, ACM, 124–128.

WARE, C., AND MIKAELIAN, H. H. 1987. An evaluation of an eye tracker as a device for computer input. In *Proc. SIGCHI+GI '87*, 183–188.

ZHAI, S., MORIMOTO, C., AND IHDE, S. 1999. Manual and gaze input cascaded (MAGIC) pointing. In *Proc. CHI '99*, ACM, 246–253.

ZHU, D., GEDEON, T., AND TAYLOR, K. 2011. Moving to the centre: A gaze-driven remote camera control for teleoperation. *Interact. Comput. 23* (January), 85–95.