An Investigation of Multimodal Kinematic Template Matching for Ray Pointing Prediction for Target Selection in VR

MARCELLO GIORDANO, Reality Labs Research, Meta Inc, Toronto, ON, Canada
TOVI GROSSMAN, Department of Computer Science, University of Toronto, Toronto, ON, Canada
AAKAR GUPTA, DANIEL CLARKE, RORIK HENRIKSON, SEAN TROWBRIDGE, STEPHANIE SANTOSA, MICHAEL GLUECK, TANYA JONKER, HRVOJE BENKO, and DANIEL WIGDOR, Reality Labs Research, Meta Inc, Toronto, ON, Canada

We explore the use of multimodal input to predict the landing position of a ray pointer while selecting targets in a Virtual Reality (VR) environment. We first extend a prior 2D Kinematic Template Matching technique to include head movements. This new technique, Head-Coupled Kinematic Template Matching, was found to improve upon the existing 2D approach, with an angular error of 10.0° when a user was 40% of the way through their movement. We then investigate two additional models that incorporated eye gaze, which were both found to further improve the predicted landing positions. The first model, Gaze-Coupled Kinematic Template Matching, resulted in angular error of 6.8° for reciprocal target layouts and 9.1° for random target layouts, when a user was 40% of the way through their movement. The second model, Hybrid Kinematic Template Matching, resulted in angular error of 5.2° for reciprocal target layouts and 7.2° for random target layouts when a user was 40% of the way through their movement. We also found that using just the current gaze location resulted in sufficient predictions in many conditions. We reflect on our results by discussing the broader implications of utilizing multimodal input to inform selection predictions in VR.

CCS Concepts: • **Human-centered computing** → *Pointing*; *Virtual reality*;

Additional Key Words and Phrases: Virtual Reality, Ray Casting, Gaze, Endpoint Prediction, Pointing, Selection

This article is an extension of a published research paper [52] and its accompanying nonarchival interactivity demo [51]. This prior work is presented predominantly in Sections 3 to 5, whereas the new work is presented predominantly in Sections 6 to 9.

Authors' Contact Information: Marcello Giordano, Reality Labs Research, Meta Inc, Toronto, ON, Canada; e-mail: marcello@marcellogiordano.ca; Tovi Grossman (corresponding author), Department of Computer Science, University of Toronto, Toronto, ON, Canada; e-mail: tovi@dgp.toronto.edu; Aakar Gupta, Reality Labs Research, Meta Inc, Toronto, ON, Canada; e-mail: aakar.hci@gmail.com; Daniel Clarke, Reality Labs Research, Meta Inc, Toronto, ON, Canada; e-mail: danclarke@meta.com; Rorik Henrikson, Reality Labs Research, Meta Inc, Toronto, ON, Canada; e-mail: rorik@henrikson.ca; Sean Trowbridge, Reality Labs Research, Meta Inc, Toronto, ON, Canada; e-mail: sean@trowbridges.net; Stephanie Santosa, Reality Labs Research, Meta Inc, Toronto, ON, Canada; e-mail: ssantosa@meta.com; Michael Glueck, Reality Labs Research, Meta Inc, Toronto, ON, Canada; e-mail: mglueck@meta.com; Tanya Jonker, Reality Labs Research, Meta Inc, Toronto, ON, Canada; e-mail: tanya.jonker@fb.com; Hrvoje Benko, Reality Labs Research, Meta Inc, Toronto, ON, Canada; e-mail: benko@meta.com; Daniel Wigdor, Reality Labs Research, Meta Inc, Toronto, ON, Canada; e-mail: daniel@dgp.toronto.edu.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.
© 2025 Copyright held by the owner/author(s). Publication rights licensed to ACM.

ACM 1557-7325/2025/4-ART3
https://doi.org/10.1145/3702319

ACM Reference format:
Marcello Giordano, Tovi Grossman, Aakar Gupta, Daniel Clarke, Rorik Henrikson, Sean Trowbridge, Stephanie Santosa, Michael Glueck, Tanya Jonker, Hrvoje Benko, and Daniel Wigdor. 2025. An Investigation of Multimodal Kinematic Template Matching for Ray Pointing Prediction for Target Selection in VR. *ACM Trans. Comput.-Hum. Interact.* 32, 1, Article 3 (April 2025), 37 pages.
https://doi.org/10.1145/3702319

1 Introduction

The usage and popularity of **Virtual Reality (VR)** technologies has increased greatly over the last decade. Ray pointing is typically used within VR environments to facilitate target selection [26]. However, controlling a virtual ray with a handheld controller can be difficult, especially for small and distant targets [69]. Selection could potentially be improved if VR systems could predict a user's intended target, while a selection action is in its initial stages.

In 2D environments, many endpoint prediction models have been developed to facilitate target selection tasks [66, 87, 94, 114]. With such models, a cursor's trajectory is continuously analyzed as it moves toward an intended target, so that the model can predict where the final endpoint of the trajectory will be. One promising technique, **Kinematic Template Matching (KTM)**, matched cursor velocity profiles to a library of templates from known movements to predict a trajectory's endpoint in 2D space [87]. Despite the promise of such 2D techniques, they have yet to be applied to 3D VR environments. Thus, within this research, we extended KTM into 3D space by harnessing the additional multimodal channels that are inherent to VR hardware configurations: 6 df tracking of the handheld controller and **Head-Mounted Display (HMD)**, and real-time eye gaze[1] that emerging HMDs can measure [1, 3]. By leveraging these additional multimodal input streams, we sought to predict a user's selection intention.

In this article, we explore enhanced endpoint prediction models for ray pointing during target selection in VR, that build upon KTM. Most notably, KTM leverages only the 2D cursor movements to predict a landing position. Our key insight is that additional multimodal channels can be integrated into the template matching procedure to improve prediction results. Our first model, **Head-Coupled Kinematic Template Matching (HeadKTM$_7$)**, compared the velocity profiles of the controller and head (via their HMD) to past historical data (i.e., templates) to predict the final landing position of the ray pointer. A data collection study found that the model's predictions were within 10.0° of the true landing position 40% of the way through a target selection movement, and within 3.4° 90% of the way through a movement. At the 40% mark (302 ms, on average), this improved prediction accuracy by 44.1% compared to a direct extension of KTM that only used hand movement. This model and the associated findings have been previously published in our own prior work [52].

An important observation from this study was that one's head provided an early indication of their intended movement direction, but it was rarely oriented directly toward the final target position. As such, we hypothesized that the eye gaze might be able to provide a more accurate indication of the final landing position of the cursor, since users are more likely to look directly at targets prior to selection [101]. While past research has explored the coordination of gaze and head movements in mixed reality [98, 108], no known research has looked at such patterns within the context of ray pointing.

[1]The term gaze has been used to refer to movements of the eye and the head. As with prior research, we use the term gaze for eye movements [99], and refer to movements of the head as head movements.

Thus, we then build upon our prior work [52], and investigate the coordination patterns between gaze, head, and controller movements, during target acquisition in VR, through a follow-up study. This study also varied the layout of the next target location to understand how task context impacted model performance. A detailed analysis of the movement metrics reiterates findings from prior work; gaze can provide a stronger indication of the final landing position than the **Head Angle (HA)**. We then explore two model variations: **Gaze-Coupled Kinematic Template Matching (GazeKTM$_7$)** which integrated gaze into the templates, and **Hybrid-Gaze Kinematic Template Matching (HybridKTM$_7$)**, which alternated between using (a) KTM predictions prior to a saccade occurring and (b) using the raw gaze location after a saccade occurred. Compared to HeadKTM$_7$, GazeKTM$_7$ enhanced predicted landing positions by 8.9% for reciprocal target layouts and by 19.8% for random target layouts, with angular errors of 6.8° and 9.1°, respectively, when participants were 40% of the way through their selection movements (356 ms, on average). Compared to HeadKTM$_7$, the second model, HybridKTM$_7$, demonstrated a 29.3% improvement for reciprocal target layouts and a 37.2% improvement for random target layouts, with angular errors of 5.2° and 7.2°, respectively, at 40% progress. Furthermore, we found that simply using the raw gaze position (GazeOnly) outperformed both models with an average angular error of 4.1° for random target layouts and 3.8° for reciprocal target layouts at 40% progress; however, it was less accurate prior to 40% progress. For reciprocal target layouts, the hybrid model (HybridKTM$_7$) was best overall, while for random target layouts, the raw gaze position (GazeOnly) provided the best estimate throughout the entire selection process.

In Section 8, we discuss considerations related to the practical applications of our work. In particular, our work, and endpoint prediction more generally, has important implications toward the design of target facilitation techniques. For example, adaptive control gain techniques [23] could bias pointer movements toward predicted landing positions. Alternatively, techniques that leverage target snapping [106, 109], or cursor bending [86, 102], could be enhanced by biasing toward targets within the predicted endpoint region. Endpoint prediction in VR could also be used as a mechanism to reduce perceived activation latency, by inferring next actions before they occur [110]. Finally, in Section 9, we discuss limitations of our work and areas for future research. In summary, the contributions of this work are:

(1) The adaptation of a KTM endpoint prediction technique to VR environments.
(2) A HeadKTM$_7$ technique that integrated velocity data from the controller and head into the predictive model.
(3) Two gaze-based KTM techniques (GazeKTM$_7$, HybridKTM$_7$) that integrated gaze data into the predictive model.
(4) Three controlled empirical studies showing that these models can outperform more direct adaptations of prior approaches while predicting ray pointer landing positions.
(5) A discussion of the practical implications of our work and implementation considerations for endpoint prediction in real-world VR applications.

2 Literature Review

The present research extends prior work on 2D cursor prediction to VR environments, so we first review related research in the areas of VR selection techniques and cursor endpoint prediction models. We then discuss related work that utilizes gaze and head movements for input and interaction.

2.1 VR Selection Techniques

One of the primary interactive operations within VR is object selection, which enables the manipulation of remote objects [26]. In the last several decades, numerous techniques have been developed

to enable and facilitate object selection in VR environments. A full review of such techniques is beyond our scope, and we direct the reader to Argelaguet and Andujar [9] for a comprehensive survey. Most critically, early research [26] identified two main classes of selection techniques, i.e., virtual hand (i.e., 3D cursors, arm-extension) [40, 43, 53, 76, 92, 113] and ray pointing or casting [26, 40, 69, 91]. With virtual hand techniques, a 3D cursor is directly controlled by the position of the user's hand. With ray pointing, a cursor emanates from one's hand or a 6 df handheld controller like a laser pointer. Within each of these classes of techniques, numerous variations and facilitation methods have been developed. For example, 3D cursors can be enhanced with nonlinear mappings to facilitate distant target selection [92], or with dynamically sized activation areas to facilitate the selection of small targets [106]. Similarly, there have been many adaptations developed for ray casting techniques, which are formally classified in the survey by Argelaguet and Andujar [9]. In our work, we focus on ray pointing for several reasons. First, ray-based approaches have been shown to be more efficient for selection across several different studies [9, 45]. Second, ray pointing has become an industry standard for object selection within most commercial VR systems. Finally, ray pointing better enables distant-target selection, whereas virtual hand techniques are typically limited to the interaction of objects within arm's reach [9, 26]. In fact, ray pointing has become not only a common technique in VR, but is also used when selecting distant targets on large displays [58, 62, 79].

One drawback of ray pointing is that it can be difficult to select small and distant targets due to the angular accuracy needed. Several techniques have been developed to improve ray pointing by decreasing the required level of pointing precision. For example, early work used a conical area for the ray [69]. Several other approaches have been considered in the literature, such as snapping to the closest target [102, 109], bending around obstacle targets [86], or increasing the ray's activation area using bubble selection mechanisms [71]. A drawback of these techniques is that they can cause ambiguity if multiple targets are in the selection region. As such, techniques have been developed to disambiguate selection intent. Examples include: providing a controllable depth cursor along the ray [15, 45], leveraging the user's gaze to infer the target depth [30], or supporting progressive refinement techniques to break down the selection into multiple stages [45, 63]. However, the literature lacks techniques to predict the landing position of a ray pointer while it's still in motion, which could further enhance the selection process.

2.2 Cursor Endpoint Prediction and Distribution Models

Existing research has explored *endpoint prediction* techniques, which predict where a cursor will land while a pointing movement is in progress. The motivation for such techniques is to enable pointing facilitation, such as target expansion [73], which depend on the ability to predict the endpoint of an input gesture. Several approaches have been proposed to perform endpoint prediction, as described below.

With regression-based extrapolation, existing models of cursor movements were used to predict the location of a distant target based on partial movements [10]. Most successful was the motion kinematics approach by Lank et al. [66], which they subsequently improved to consider the stability of the prediction [94]. An alternative approach is to use target classification, which integrates knowledge about targets in the environment to identify the most probable candidate target [78]. Recent work used neural networks and Kalman filters to predict user intent based on the kinematics of the cursor [11, 21].

Related to these approaches are models based on **Optimal Feedback Control (OFC)** from motor control theory [105]. In such models, the human is represented as a dynamic system which is continuously observing sensory movements and determining optimal control actions such as reaching or pointing. Such models are related to our work as they can provide predictions of motion trajectories and endpoints during early stages of a motion [68]. Indeed, the use of such

models has become an active area in **Human–Computer Interaction (HCI)** recently [36, 39, 42]. For example, Fischer et al. examine how OFC can be used to understand 2D mouse pointing behaviors [39]. Their work demonstrates that the OFC models can be used to predict and explain the entire movement trajectory of a mouse pointer. Bachynskyi and Müller further showed the application of OFC to model mid-air movements used in VR and AR interfaces [12]; however, the model requires knowledge of the end target position. That said, OFC models can be used to reverse-engineer the human's objective function (i.e., desired endpoint) using inverse optimal control. For example, Ziebart et al. assigned probabilities to targets using inverse optimal control and Bayes' rule for target prediction [114]. While such techniques are promising, they are complex and require knowledge of the target locations. A formal comparison to OFC approaches was out of scope for our work, but we discuss possible implications to ray pointing prediction in Section 9.5.

A final approach is *KTM* [87]. With this technique, the velocity profile of a partial pointing movement is compared to a library of known "template" movements to predict the final cursor position. The technique compares a user's movement to the user's own template library. This allows results to be personalized to each individual's pointing behaviors, at the cost of requiring individual data collection. To predict the candidate movement's final endpoint, the technique uses the travel distance associated with the best matched template and applies that distance to the current direction of the candidate's movement from the original start point. It was found that on average, KTM predicts the endpoint location within 83 pixels of the true endpoint when 50% of the movement has been completed, 48 pixels at 75%, and 39 pixels at 90%. This performance was better than prior kinematic endpoint prediction methods [66]. Although untested, the authors suggested that this prediction could then be combined with target selection techniques such as target expansion [73] or gravity wells [23]. Template matching offers several advantages over the other techniques, i.e., it is target-agnostic, user-adaptable, and easy to implement [87]. As such, this work builds upon this approach, and we refer the reader to this prior work for its implementation details [87].

Target prediction has also been leveraged beyond desktop configurations. In touch-based interfaces, it has been used to reduce perceived latency [28, 81, 82]. For example, Xia et al. leveraged hover information to predict when and where a touch would occur [110]. Ahmad et al. applied similar predictive models for in-car, mid-air selection [5]. Despite the promise of these techniques, we are unaware of work that applies prediction models to VR ray pointing, to predict selection intent.

In addition to endpoint *prediction* models, there have been several efforts to model endpoint *distributions* during 2D [17, 18, 19, 61] and 3D [108, 111] target acquisition tasks. An early example for touch devices includes the work by Bi and Zhai which modeled touch points as a dual Gaussian distribution and considered the target with the shortest "Bayesian touch Distance" from a touch point as the desired target [18]. Follow-up work has provided similar Gaussian assumption-based models to improve selection of 2D moving targets [46]. In the realm of 3D environments, Yu et al. offered a model (*EDModel*) of endpoint distributions of a ray pointer during 3D target acquisition in VR environments [111]. Similarly, Wei et al. modeled the endpoint distributions of the head and gaze vectors during gaze-pointing in AR environments [108]. These models have all been shown to improve pointing accuracy by correcting the final endpoint once targeting motion has been completed. In contrast, our work (and endpoint prediction more broadly) seeks to predict the endpoint of a target acquisition movement while the movement is still in progress.

2.3 Understanding and Utilizing Gaze and Head Movements for Input and Interaction

The use of eye movements for input in VR environments was proposed by Tanriverdi and Jacob [104] and a design space for such interactions has been recently developed [54]. For goal-directed movements, it is understood that hand movements are preceded by eye movements that guide the

hands toward the object of interest [65, 101]. Within the domain of physical interaction, Helsen et al. investigated the coordination of the hand and point of gaze while aiming [48–50] and found that the point of gaze arrived on a target prior to the hand at approximately 50% of the response time. The coordination of eye and pointer movements has also been investigated during desktop-based tasks such as tracing [35], visual search and selection [20], Web search [31, 56], and real-world PC use [70], and while using multitouch surfaces [88, 89]. Researchers have begun to explore the coordination of eye, pointer, and head movements in VR environments [96, 98, 100], but have not yet explored gaze behaviors during ray pointing selection in VR.

Gaze has often been used as an input modality for target acquisition [24, 37, 38, 80, 103, 107, 112]. For example, Zhai et al. proposed Manual and Gaze Input Cascaded Pointing [112], where a cursor jumped to a user's gaze location and could then be refined with mouse movements while using a desktop. Most relevant to our own work is the use of gaze for pointing in VR environments. Early work demonstrated mixed results when using gaze-based pointing rather than hand-based pointing [33, 84, 104]. In fact, a study by Qian and Teather showed that utilizing head movements instead of eye movements provided better selection performance [93]. Recent work has proposed to, instead, combine gaze with hand and head-based input in VR [57, 64, 90]. For example, Sidenmark and Gellersen explored a set of selection techniques that utilized the relationship between eye and head movements, resulting in improved control and flexibility [97]. This past work reveals that even though multimodal channels, such as gaze, may be less accurate or more noisy than handheld controllers, they can still be leveraged to improve selection. In our work, we utilize gaze and head movements to predict where a user intends to point, in contrast to the above techniques, which used such modalities to facilitate final selection operations.

There have been prior efforts to utilize gaze and head movements to predict a user's intent. Example uses include: intent recognition during object manipulation on tabletop displays [13] and target prediction for desktop selection [22]. In the realm of VR HMDs, Casallas et al. used relative head-target and hand-target features to predict intended moving targets [27], Arabadzhiyska et al. used gaze patterns to predict saccade endpoints to aid with foveated rendering [8], Mardenbegi et al. investigated the use of gaze, coupled with head movements to predict the depth of targets to resolve targeting ambiguities [72], and David-John et al. used gaze dynamics to predict *when* a user intends to interact in VR [34].

Most relevant to our own work, Cheng et al. used gaze and hand movements to predict intended targets in VR [32]; however, their work focused on direct hand manipulations. Their algorithm used the point of gaze as the prediction at the instant that a hand movement began. It is unknown how reliable this method is, especially given that fixations do not always occur while reaching [96]. Furthermore, this approach was developed for direct manipulation, whereas we wish to apply such a technique to remote target acquisition using ray pointing. Finally, recent work by Wei et al. examined eye and head endpoint distributions during gaze-based pointing [108]. Their analysis was used to build predictive models and improved pointing accuracy of gaze-based selection. We aim to use a complementary approach to improve the prediction accuracy for ray-based selection.

In summary, prior predictive techniques have demonstrated the use of naturally occurring gaze and head movement data. However, they did not focus on detailed behaviors of gaze and head movements during ray pointing in VR. Our work contributes findings that provide such behaviors and inform our multimodal ray pointing prediction models.

3 HeadKTM$_7$

In this section, we describe our HeadKTM$_7$ model, which leverages the head movements of the user to predict the final controller location and direction while the pointer movement is still in progress. To adapt the KTM approach [87] for ray pointing in VR environments, we considered the traditional

ray pointer to be a virtual laser pointer that had 5 df, i.e., the origin (*X, Y, Z*) and direction (θ_h, θ_v) of the ray. Our head-coupled model adapted and extended the KTM technique as follows:

(1) The KTM method was built for 2D cursor pointing, predicting the (*X, Y*) coordinates of the movement's endpoint. To adapt the technique for 3D ray pointing, we are not predicting an "endpoint" *per se*, rather the final landing position of a ray. Thus, estimates of not only the 3D coordinates of the handheld controller but also the angle at which the ray is being emitted are needed.
(2) The KTM method only considers the velocity profiles of the cursor in the template matching procedure. Thus, a cursor's velocity profile across targets with different distances may not be distinguishable in the first part of its movement [41, 60]. We extend this method to also consider the user's head movement, hypothesizing that this additional channel may increase prediction accuracy. In total, four velocity profiles are considered: the positional and angular velocities of the controller and head. Their relative importance are weighted by parameters (*a, b, c,* and *d*). We call this head-coupled variation *HeadKTM*.
(3) The KTM method selects one, best matching template, to estimate the endpoint distance. We extend the method to a *top-n* approach, where the weighted average of multiple matching templates may be used. This may compensate for when the top match may not be as accurate. We call this *top-n* variation KTM_n.

These enhancements were the backbone of our $HeadKTM_n$ technique, where *n* represents the number of matching templates which are used. The technique is target-agnostic and can be personalized to individual users. The approach follows the same general approach as KTM, described below. The tuning process for the model parameters (*a*, *b*, *c*, *d*, and *n*) is described later in Section 4.6.

3.1 Step 1. Building the Template Library

The template library is built by capturing selection movements for known targets, considering both the motion of the handheld controller and the head during selection. Because these are spatial input channels, both the location and the angle of the controller and head (via HMD) are considered (Figure 1(a)). As such, each template consists of four velocity profiles (Figure 1(b)):

—**Controller Positional (CP)** velocity—the change in the controller's (*X, Y, Z*) coordinates in millimeters per second.
—**Controller Angular (CA)** velocity—the change in angle of the controller's forward vector in degrees per second.
—**Head Positional (HP)** velocity—the change in the head's (*X, Y, Z*) coordinates in millimeters per second.
—**HA** velocity—the change in angle of the head's forward vector in degrees per second.

In the previous KTM technique, the template library cropped any backtracking movements from a template (e.g., for movements that overshoot their intended target) [87]. Our initial testing found that adequate results were achieved without this step. Unlike KTM, it was necessary to smooth the templates due to the noise introduced by the controllers. A Gaussian smoothing operation was performed on each velocity using a 5-point window ($\sigma = 1.66$). The profiles were then resampled to 20 Hz in preparation for comparison to subsequent candidate movements.

3.2 Step 2. Preprocessing Candidate Pointing Movements

As a new candidate movement is captured, the position and angle values of the handheld controller and head were used to create the four partial velocity profiles. They were smoothed using a 5-point Gaussian window ($\sigma = 1.66$) and resampled to 20 Hz as each new point was collected. As with

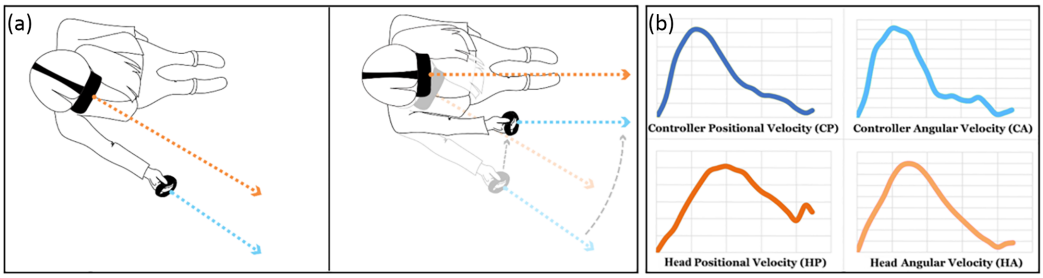


Fig. 1. (a) During a ray pointing movement, both the head and controller change in position and angle. (b) Each template consists of four velocity profiles: CP velocity, CA velocity, HP velocity, and HA velocity.

KTM, each velocity profile in the template library that had a longer duration than the candidate movement was truncated to the candidate movement's length.

3.3 Step 3. Matching Candidate Movements

The candidate movement, C, was then compared to each template, T_i, at the arrival of each new data point using the cumulative scoring function from KTM [87], which averages the difference between the velocity values at each timestamp. The scoring calculation was repeated for each of the four velocity profiles, resulting in four scores (i.e., S_{CP}, S_{CA}, S_{HP}, and S_{HA}). The final cumulative scoring function, $S(T_i)$, was a weighted sum of the four individual scores, where a, b, c, and d were tuning parameters (Equation (1)). Note that by setting a, c, and d to 0, the model would be analogous to the KTM model and only use the velocity profile of the CA:

$$S(T_i) = aS_{CP}(T_i) + bS_{CA}(T_i) + cS_{HP}(T_i) + dS_{HA}(T_i). \tag{1}$$

3.4 Step 4. Calculating the Expected Landing Position

To predict the landing position, the n-best template matches are considered, which are ranked by the minimum values of $S(T_i)$. To calculate the expected final *movement angle distance* of the ray, a weighted average of the movement angle distances of the *top-n* templates is computed. The template's weight, w_i, is defined as the reciprocal of $S(T_i)$, and the template's movement angle distance is d_i. Using these values, the weighted average angular distance is calculated by

$$\mu = \frac{\sum_{i=1}^{n}(w_i * d_i)}{\sum_{i=1}^{n} w_i}. \tag{2}$$

The controller's initial *angle* is rotated by μ, along the current angle of motion (as defined by the angle between the current controller's forward vector and the initial controller's forward vector). The same approach was used to calculate the expected CP. Using the weighted average of the *top-n* template's controller distances, the magnitude of this average is added to the initial controller's position along the current direction of movement. By combining the expected *angle* and *position*, the final ray pointer landing position is calculated (Figure 2).

4 Experiment 1: Understanding the Coordination of Head and Controller Movement

Our evaluation of HeadKTM was divided into two separate experiments. This first experiment gathered initial data to better understand human head and controller behaviors during ray pointing using controlled target sizes and distances. The experiment consisted of a pointing task in a VR environment, using a ray pointer, without prediction enabled. This first experiment was also used

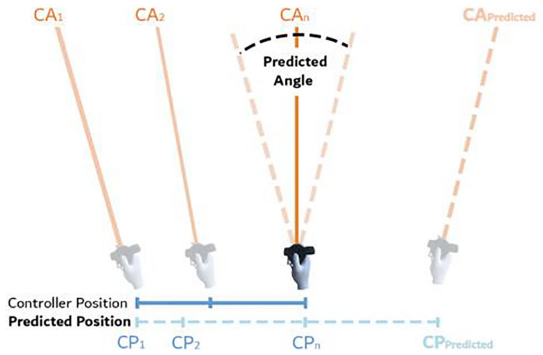


Fig. 2. To predict the final landing position, the predictions for the final angle and position of the controller were combined.

to determine the parameters for the model, while the second experiment, presented in Section 5, evaluated the effectiveness of the model compared to baseline approaches.

4.1 Participants

Seventeen participants (11 female), with no major motor impairments and normal or corrected-to-normal vision (only contact lenses were allowed) were recruited. They ranged in age from 18 to 26 ($M = 21$ years, SD = 2 years). Participants were compensated $30 CAD for their time. A Randot Stereo Optical Test was administered prior to the experiment to ensure adequate stereo vision. All participants were right-handed and operated the controller with their right hand. Informed consent was obtained prior to the study.

4.2 Apparatus

The experiment was conducted using an Oculus Rift CV1 HMD, with a resolution of 2,160 × 1,200, using a single Oculus Touch handheld controller for input. The Index Trigger button was used for selection. The position and angle of the HMD and controller were tracked using Oculus Constellation Sensors. The system ran on a 3.7 GHz Intel Core i7-8700k desktop computer with an NVIDIA GeForce RTX2080 graphics card and was developed in Unity3D. The HMD display updated at a frequency of 90 Hz, and both the HMD and CPs and CAs were updated at a rate of 90 Hz. The handheld controller manipulated a ray pointer using an absolute mapping, with the ray originating from the tip of the controller, aligned with the z-axis of the local handheld controller coordinate system.

4.3 Procedure

The task was a reciprocal 3D pointing task, wherein participants pointed back and forth, in succession, between a start and end target (Figure 3). No distractor targets were included. The target to be selected was a yellow sphere, and the other target was a semi-transparent gray sphere. The background of the scene was a gray gradient. In the VR environment, participants stood on an elevated platform above an infinite grid ground plane. The target to be selected turned green when it was intersected by the ray to indicate that the target could be selected. Upon successful selection, the targets swapped colors. If the ray did not intersect the target to be selected when a button-click occurred, the trial was counted as an error, and the participant kept trying to select it until they were successful.

Participants were asked to complete the task as quickly as possible, without exceeding an error rate of 4%. The error rate was displayed after each block of trials. During the study, participants stood on a marked floor position and were told not to move their feet. The software and the experimenter

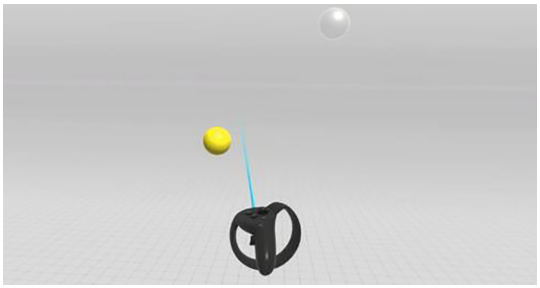


Fig. 3. First-person view of the study environment.

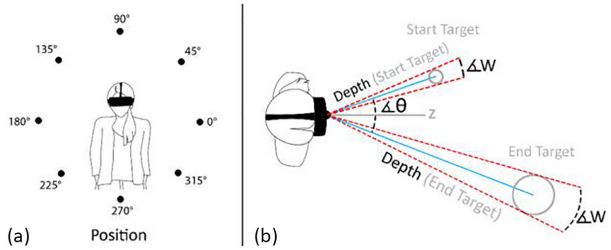


Fig. 4. The target layout. (a) Reciprocal targets were on opposite sides of the z-axis at varying depths. (b) Targets appeared at one of eight angles with equal angular widths.

ensured their feet were in the proper position prior to each trial. The coordinate system was calibrated after the participant stood on a marked spot prior to the study. The experimenter could recalibrate at any time during the study. The point between the eyes was the origin, with the positive axes being left to right (X), bottom to top (Y), and back to front (Z). Participants performed approximately 2 minutes of practice trials to become familiar with the task.

4.4 Design

A repeated-measures, within-participant design was used. The position of the goal target varied based on three independent variables—*Depth* (3 m, 6 m, 9 m), *Theta* (25°, 50°, 75°), and *Position* (0°–315° at 45° increments) (Figure 4). *Depth* manipulated the distance between the target center and the origin. *Theta* changed the magnitude of the angle between the vectors that was generated by connecting each target to the origin, with the center vector of these two vectors laying along the Z-axis. For each combination of *Theta* and *Depth,* there was a ring of target locations (i.e., *Position*) evenly distributed at 45° increments (Figure 4(a)). During each reciprocal task, targets were placed in opposite locations of the ring, but their depth values could vary (Figure 4(b)). A target's size was determined by its angular width, W (4.5°, 9.0°), relative to the origin. With W fixed, the further the targets were, the larger their radius; however, the angle needed to place the ray within its boundaries remained fixed. During each reciprocal task, the angular width of both targets was equal (Figure 4(b)).

The study had 54 blocks for each of the 54 possible combinations of *Depth* (start target), *Depth* (end target), *Theta*, and W in random order. For each block, four sets of reciprocal trials were performed for the eight *Positions* (i.e., four pairs), and consisted of nine selections (i.e., eight reciprocal selections between the two targets at opposite positions). This resulted in $54 \times 4 \times 8 = 1{,}728$ trials per participant. The experiment was completed in 60 minutes and participants were encouraged to take breaks between blocks to prevent fatigue.

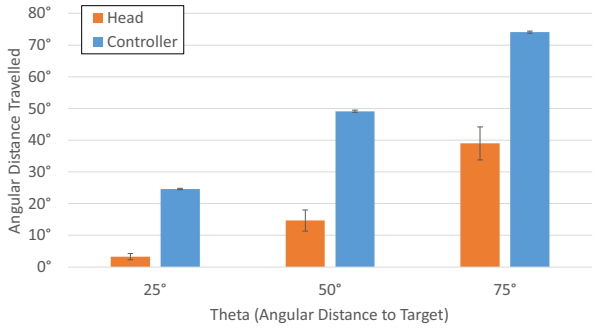


Fig. 5. The HA and CA distance traveled, with respect to *Theta*.

4.5 Results

Prior to the analysis, outliers were removed (i.e., trials where the duration was longer than 2 SDs of trials with the same *Theta* and *W*; 6% of the data). Trials where errors occurred (1.7%) were only analyzed up to the end of the first selection attempt. We focus our analysis on the angular movements of the head and controller: the angular distance traveled, and their velocity profiles throughout the movement, since ray pointing is best described as a function of the angular amplitude of movement [62]. An **repeated measures analysis of variance (RM-ANOVA)** was used for statistical analysis. Error bars in all graphs in this article represent 95% CIs.

4.5.1 HA and CA Distance Traveled. We first look at the main effect that each independent variable had on the angular distance traveled of both the head and controller (Figure 5). This is calculated by looking at the cumulative angular distance between each sample within a trial. The angle between targets, *Theta*, had a significant effect on the angular distance traveled for both the head ($F_{2,\ 32} = 272.4$, $p < .0001$) and controller ($F_{2,\ 32} = 116{,}673$, $p < .0001$). For the controller, this is intuitive, as the angular distance must be traversed to reach the target. The *Width* was also found to have a significant effect for both the head ($F_{1,\ 16} = 79.1$, $p < .0001$) and controller ($F_{1,\ 16} = 18.2$, $p < .005$). Interestingly, the distance traveled by the head increased as the *Width* decreased, likely because participants tried to see the smaller targets better. The *Position* also had a significant effect for both head ($F_{7,\ 112} = 110.1$, $p < .0001$) and controller ($F_{7,\ 112} = 7.6$, $p < .0001$). *Depth* had a significant effect for controller ($F_{2,\ 32} = 17.9$, $p < .0001$), but not for head ($p = .07$). Overall, the results show that the main factors of targeting will influence the trajectories of both the controller and head movements, indicating promise for using both channels in a template matching model.

To perform a statistical comparison of angular distance between head and controller, an additional variable, *Modality*, was introduced in our analysis. The head moved only a fraction of the angle that the controller moved ($F_{1,\ 16} = 422.0$, $p < .0001$), and there was significant interaction between *Modality* and *Theta* ($F_{2,\ 32} = 48.9$, $p < .0001$). This is intuitive because the CA must move within the bounds of the target, while the head must only move so that the target is in the participant's field of view. Figure 6 illustrates a scatterplot of the final focal point of the Head and Controller, when each target was acquired. The controller focus (blue) was within the bounds of the target, while the head (orange) only traveled a fraction of the distance, with observable variation between trials.

4.5.2 Head and Controller Velocity Profiles. We also evaluated how the velocity profile templates differed across movement angles. To generate "representative" velocity profiles, the velocity profiles were resampled to 20 Hz, and the average velocity at each interval was computed (Figure 7). Consistent with prior findings [41, 60], the velocity profiles for the controller were not distinguishable

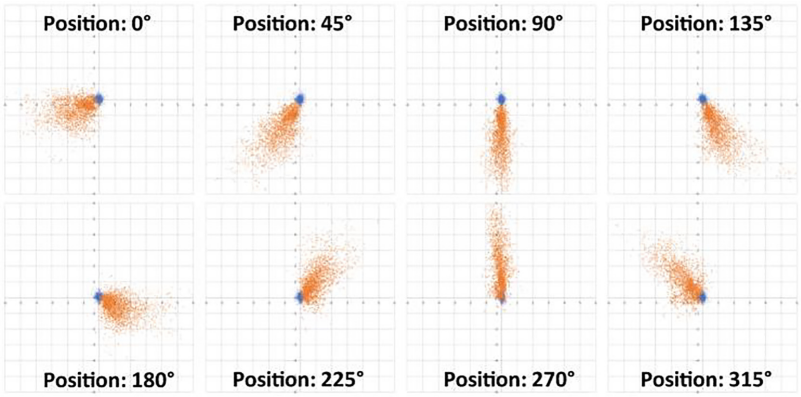


Fig. 6. A scatterplot of the final controller (blue) and head (orange) locations, for each position. Points are the intersection of each ray with a vertical plane centered at the target location. The controller focus (blue) was within the target's bounds, while the head only traveled a fraction of the required distance, with observable variation between trials.

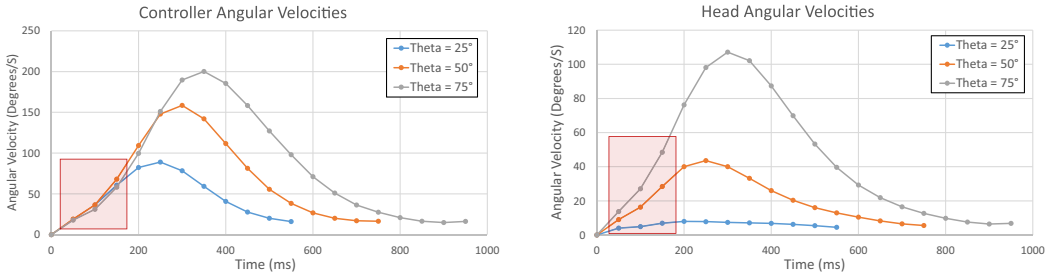


Fig. 7. Representative angular velocity profiles for the controller (left) and head (right) by movement angle. The highlighted regions illustrate that in the first 150 ms of movement, profiles only diverge for the head, not the controller.

during the initial stage of movement across the three movement angles. However, the velocity profiles for the head diverged immediately. This suggested incorporating head movements within a predictive model may allow for earlier predictions.

4.6 Model Parameters and Performance

To analyze the performance of our model and tune its parameters, intra-participant template libraries were created from each participant's trials. Each movement was compared against the other templates in the participant's template library. Angular error was used to calculate the accuracy of individual predictions, defined as the angular distance between the predicted ray and the ray from the predicted controller origin through the center of the intended target. This angular error informed the precision of the model, and was chosen, as it provides a single representative metric of accuracy, analogous to the pixel distance used to evaluate 2D endpoint prediction models. We discuss additional metrics of precision in our future work section.

The model had two main factors to tune: the weights in the scoring algorithm for the four velocity profiles (Equation (1)) and how many templates to use in the *top-n* matching algorithm (Equation (2)). To determine n for the *top-n* matching templates, four equally weighted components were used (i.e., a, b, c, $d = 1$), and the cumulative accuracies of all trials for all participants and different

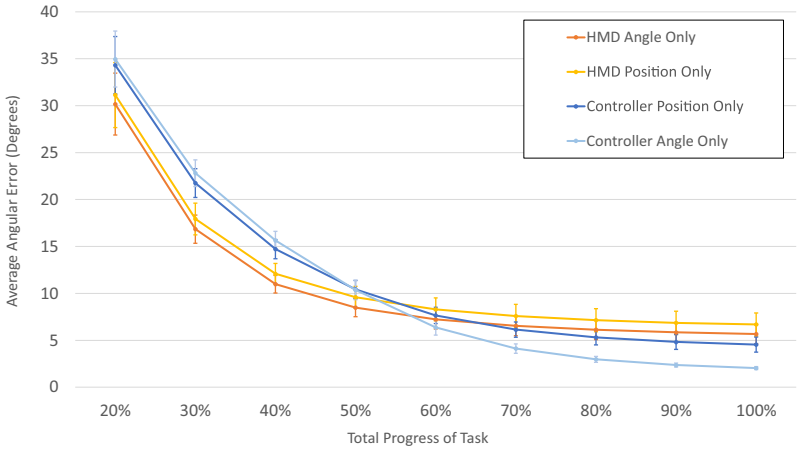


Fig. 8. The angular error of the prediction derived from each input channel at various stages of the movement.

values of n were calculated. Results improved gradually from $n = 1$ to 4, whereas $n = 4$ to 10 were similar, with $n = 7$ providing the best result, which is the value used for our implementation.

To determine the best weighting values for the scoring function (i.e., a, b, c, and d in Equation (1)), various values were used to try and optimize the model's accuracy at 40% of the movement progress (thus prioritizing early prediction). For context, the mean movement time across all trials was 754 ms, so a prediction at 40% progress would occur roughly 302 ms after a movement begins on average; however, this would be dependent on *Theta*. An RM-ANOVA was performed to compare the angular error of the four models, that exclusively used one of the four respective velocity profiles, at 10% intervals of progress points (Figure 8). The model had a significant effect on the angular error ($F_{3,\ 48} = 4.5$, $p < .01$), as did the progress point ($F_{9,\ 144} = 92.1$, $p < .001$). There was also a significant interaction effect between model and progress point ($F_{27,\ 432} = 120.7$, $p < .001$). The data in Figure 8 suggest that the HA may provide a better indicator for the first half of the movement, while the CA may be a more accurate predictor toward the end of a movement. This may be due to the velocity profiles of the head being more distinguishable in the first phase of movement (Figure 7). By considering the relative performance of each input channel, setting the best at 1 and worst at .5, and interpolating the remaining two values, we derived values of $a = .95$ (CP), $b = .5$ (CA), $c = .86$ (HP), and $d = 1$ (HA) and average accuracies of 6.4°, 3.3°, and 2.3° at 50%, 70%, and 90% of the way through the task, respectively. We call this model HeadKTM$_7$, for "Head-Coupled KTM" with $n = 7$.

5 Experiment 2: Validation of HeadKTM$_7$

This second experiment was conducted to further validate the model and compare it to baseline approaches. Experiment 1 only examined three movement angles, whereas this second experiment evaluated the robustness of the model against a continuous range of angles, depths, positions, and widths.

5.1 Participants

A total of 12 participants (9 female) completed Experiment 2 and were recruited from the pool of 17 participants from Experiment 1. Participants were compensated $30 CAD. Informed consent was obtained prior to the study.

5.2 Apparatus and Procedure

The same apparatus and procedure used in Experiment 1 was used in Experiment 2.

5.3 Design

An repeated-measures within-participant design was used. As with the first experiment, the task was a reciprocal 3D pointing task with no distractor targets. The only controlled variable was *Theta*, which used all angles from 15° to 85°, at 1° intervals. All other task variables were randomized. The *Depth* of both targets ranged continuously from 3 m to 9 m, *Position* ranged from 0° to 359°, and *Width* ranged from 4.5° to 9.0°. The experiment took approximately 60 minutes, with each angle of *Theta* repeated seven times in random order. To prevent fatigue and to allow breaks, the experiment was divided into 50 blocks, with each block presenting 7 pairs of targets with randomized *Theta* values (thus requiring 6 reciprocal selections). This resulted in $50 \times 7 \times 6 = 2{,}100$ trials per participant. Before each session, participants were given practice trials, lasting about 2 minutes, to familiarize themselves with the task.

5.4 Analysis

All trials were analyzed; however, trials where errors occurred (2.7%) were only analyzed up until the first selection was attempted. HeadKTM$_7$ was compared to a direct adaptation of the KTM model. Because these two models vary in two ways (use of head-coupling, and averaging of top seven templates), we also performed a comparison to KTM$_7$ (KTM, $n = 7$) and HeadKTM (KTM, with head coupling), to understand the benefit of these two variations. As in prior endpoint prediction work [114], the results were also compared to a baseline that utilized the current position of the ray without any prediction applied. An RM-ANOVA was performed to compare the angular error of the five models at 10% intervals of task progress points. A Bonferroni correction was applied to all *post-hoc* pairwise comparisons, by multiplying the (uncorrected) p-values by the number of comparisons made [2].

5.5 Results

There were overall significant effects of model ($F_{4,\ 44} = 187.1$, $p < .001$) and progress point ($F_{9,\ 99} = 132.7$, $p < .001$) on the angular error. There was also a significant interaction effect between model and task progress ($F_{36,\ 396} = 100.9$, $p < .001$). The results indicated that HeadKTM$_7$ outperformed the KTM, KTM$_7$, and HeadKTM models (all at the $p < .001$ level) (Figure 9). Overall, HeadKTM$_7$ provided promising results, with an angular error of 7.3° at 50% of the way through the task, 4.4° at 70%, and 3.4° at 90%. This was encouraging, as it showed that both proposed enhancements could increase the accuracy of predictions. Compared to the baseline with no prediction, HeadKTM$_7$ improved accuracy by 62.1% at 40% progress. However, near the end of the movement, the baseline was the most accurate. This is consistent with prior findings [114] and suggests that adaptive techniques that can detect when a movement is ending may be useful. It is interesting that HeadKTM outperformed KTM$_7$, indicating that the improvement of our technique primarily stems from using the head-coupled data. The differences in accuracy were most pronounced at 40% progress, with angular errors of 10.0° for HeadKTM$_7$, 11.6° for HeadKTM, 15.0° for KTM$_7$, 17.9° for KTM, and 26.4° for the baseline. The average movement time across all trials was 731 ms, so a prediction at 40% progress would occur 292 ms after a movement began, on average.

5.6 Summary

By modifying the KTM model to include head-coupling and consider the *top-n* matches, the HeadKTM$_7$ predictions were 44.1% and 62.1% more accurate than KTM and the baseline, respectively,

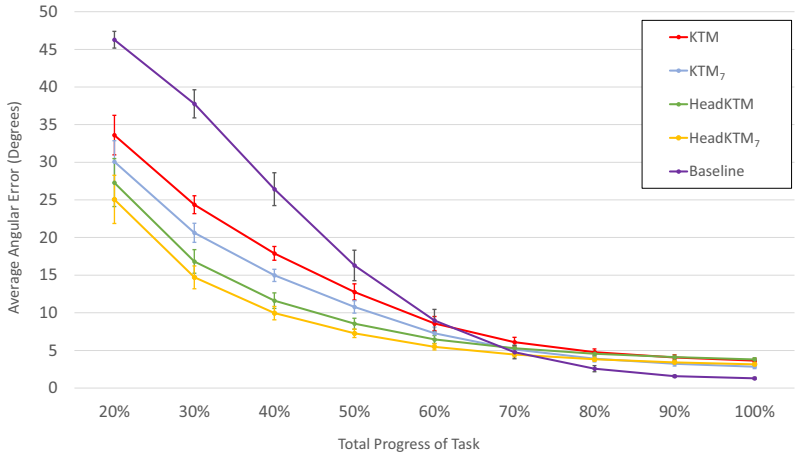


Fig. 9. Angular error for the baseline and four variants of the model, i.e., KTM, KTM_7, HeadKTM, and $HeadKTM_7$.

40% of the way through a participant's movements (292 ms). The most important insight was that head movements provided an earlier indication of the participant's intent (Figure 7); however, the head did not point directly toward the target (Figure 6) because participants only moved their head enough to see the target in their field of view. This may explain why $HeadKTM_7$ provided the most improvement during the initial stages of pointing.

6 Experiment 3: Use of Gaze during Ray Pointing

Experiment 2 demonstrated the promise of utilizing head movements to improve the ray pointer landing position during target selection. In Experiment 3, we seek to understand if leveraging gaze data could further improve the accuracy of landing position predictions. The purpose of this study is two-fold. First, we wanted to better understand how gaze is coordinated with head and controller movements during VR ray pointing selection (Figure 10). Shedding light on these coordination behaviors would aid in the development of our gaze-enhanced models, but also guide future research efforts which aim to leverage gaze in VR environments. Second, in Section 7, we will use the data collected from this study to test our proposed gaze-enhanced models.

6.1 Participants

Twenty-five participants with no major motor impairments and with normal vision (no glasses or contact lenses were allowed to ensure accurate eye tracking) were recruited. Participants were 18 to 65 years old.[2] Participants were compensated $30 for their time. A Randot Stereo Optical Test was administered prior to the experiment to ensure adequate stereo vision. Informed consent was obtained prior to the study.

6.2 Apparatus

The experiment was conducted using an HTC Vive Pro Eye HMD that had a display resolution of 2,800 × 1,600 and an embedded eye tracker. The eye tracker had a reported accuracy of .5°–1.1° and

[2]Due to unintended data storage loss, we are unable to report the specific breakdown of participant age and gender for this study. The participant pool was made up of a range of ages, genders, and experience levels with 3D games and VR systems. They were representative of typical North American adult computer users that would not experience challenges using VR platforms (no visual or motor impairments).

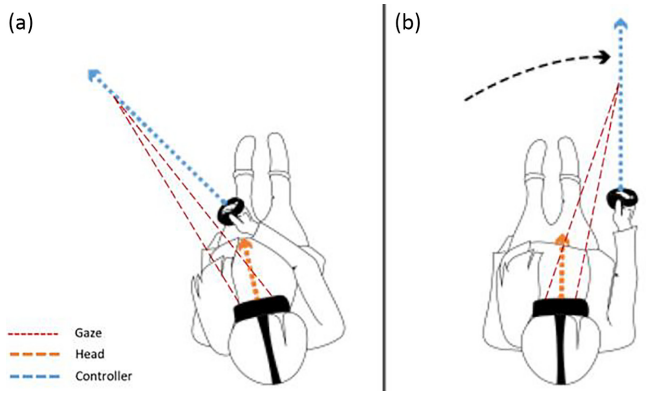


Fig. 10. Experiment 3 explores the coordination patterns of the CP, HP, and gaze position during VR target selection.

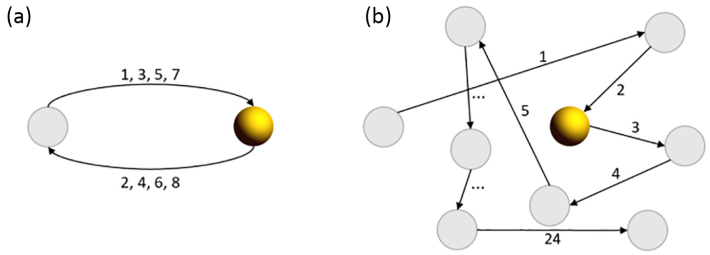


Fig. 11. The study used two target layouts. (a) The reciprocal layout trial set consisted of eight trials between two reciprocal target positions. (b) A random layout trial set consisted of 24 trials across randomly positioned targets. The current target was rendered as a yellow sphere. The next target did not appear until the previous target had been selected.

framerate of 120 Hz. The HMD positions and angles were updated at a rate of 90 Hz. The system ran on a 3.7 GHz Intel Core i7-8700k desktop computer with an NVIDIA GeForce RTX2080 graphics card and was developed in the Unity3D environment.

6.3 Task

The study used a target acquisition task. Participants were placed in a scene with a gray gradient background and an infinite grid ground plane and were asked to select a target rendered as a yellow sphere. The participant selected the target by using a virtual ray that was controlled by a handheld controller. Once the target was intersected by the ray, the participant could click the trigger on the controller and the next goal target would be displayed. If the ray did not intersect the target, the trial would count as an error, and the participant would need to adjust the ray and click again.

The target layout was either reciprocal or random (Figure 11). The reciprocal layout involves a planned back and forth motion, whereas with the random layout, the user must wait until the next target is revealed before initiating their movement. For the reciprocal layout, the participant would switch between selecting one of two targets that were placed equidistant from the center of the participant's viewpoint at controlled distances and angles. Participants alternated between selecting each target 8 times. For the random layout, the target positions were pregenerated and presented in a random order. This task represented situations where a user could not predict where a target would appear until the trial began, which was hypothesized to influence their head

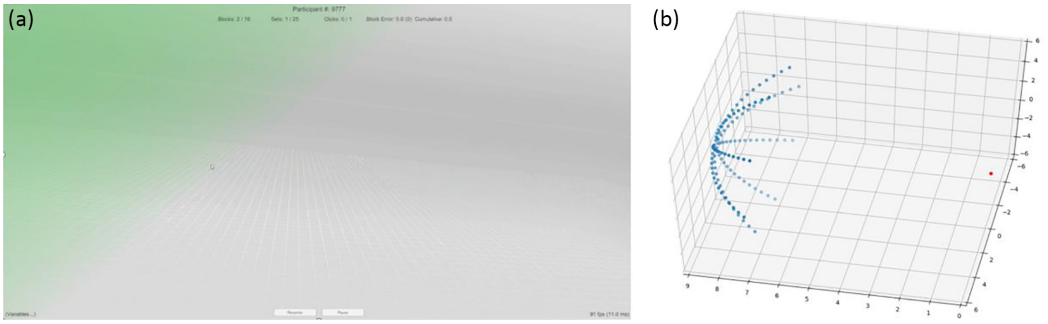


Fig. 12. (a) The green glowing effect that indicated which direction the next target was going to appear. This effect was used to avoid visual search behaviors when the next target fell out of the participant's field of view. (b) The target positions used in the study (blue) were at a fixed depth of 9 m from the participant (red dot).

and gaze coordination patterns [96]. Participants selected eight targets when this target layout was used.

6.4 Procedure

Prior to the study, to calibrate the system, the coordinate system was reset after the participant found a comfortable position in a chair, with the HMD in a resting state. The point between their eyes was set as the origin, with the positive axes being left to right (X), bottom to top (Y), and back to front (Z). The eye tracker was also calibrated for each participant with a short calibration procedure. Participants were given a 2-minute warmup to familiarize themselves with the task and sat in the chair throughout the experiment. During initial testing, we noticed that in the random layout, the target would sometimes fall outside the participant's field of view, which would cause them to engage in a visual search to find the position of the next target. To avoid visual search behaviors, we introduced an off-screen visualization [16] using a green glowing effect that would appear in the direction of the next target when it fell outside the participant's field of view (Figure 12(a)).

6.5 Design

For this study, an repeated-measures within-participant design was used with two factors, *Layout* (i.e., reciprocal, random) and *Amplitude* (i.e., 5°–60° in 5° increments). Each trial consisted of a start position and end position. For the reciprocal *Layout*, the targets were positioned radially around the origin opposite to each other at a fixed depth from the participant (i.e., 9 m). The movement direction between targets ranged from 0° to 135° at 45° intervals. For the random *Layout*, targets would appear randomly within the same bounding sphere determined by the outermost targets in the reciprocal *Layout*. The amplitude between targets varied from 5° to 60° in 5° increments (Figure 12(b)). The angular width of the target (i.e., the angle of the target boundaries relative to the origin) remained fixed at 4.5° throughout the study.

The experiment was completed in a single session lasting about 30 minutes. Within each session, the order of the *Layout* (i.e., reciprocal or random) was randomized. Participants performed 384 trials for each *Layout* for a total of 768 trials; however, the structure differed between *Layout*. For the reciprocal *Layout*, there were 12 blocks of trial sets, one for each of the 12 *Amplitudes*, in random order. Within each block, there was one trial set for each of the four *Directions*, in random order. Within each trial set, the participant moved their cursor back and forth between the two targets 8 times (12 *Amplitudes* × 4 *Directions* × 8 repetitions = 384 trials). For the random *Layout*, there were

16 trial sets, consisting of 24 trials each. Within each trial set, the targets' positions were randomly chosen, with each of the 12 discrete *Amplitudes* (5°–60°, at 5° increments) appearing twice each (16 trial sets × 12 *Amplitudes* × 2 repetitions = 384 trials).

6.6 Gaze Data Processing

The eye tracking data provided timestamped (*x, y, z*) coordinates that represented fixations and saccades. The vector was output directly by the eye tracker as a combination of the individual gaze vectors from the left and right eyes pointing in the direction of the participant's gaze. Due to noise in the gaze data, we needed to process these data. First, we identified short gaps in the data (e.g., one to five consecutive samples) that had no reported position due to tracking loss or blinking and used linear interpolation to fill them. Second, we filtered the gaze data using the 1 Euro Filter with a minimum cutoff of 9 and a beta .7, which biased toward minimizing latency. Finally, the resulting data were segmented into saccades and fixations using the dual threshold implementation of the velocity threshold heuristic [6]. The start of a saccade was defined as the moment when the velocity exceeded 130°/s (V_d) and the end of a saccade as the moment when the velocity dropped below 70°/s (V_f). This data processing procedure was applied to each trial to segment the data into fixations and saccades.

6.7 Data Exclusions

Given the inherent noise in the gaze tracking data and the importance of this information for our modeling efforts, we implemented thorough outlier removal. Data from four participants had to be removed due to calibration errors (P1, P13) and poor gaze tracking data (P7 and P12). We removed trials in which the target was not successfully selected on the first attempt (1,991 trials). We also removed individual trials that were more than 3 SDs from the mean with respect to the gaze location at the time of target selection (156 trials), the number of saccades in a single trial (149 trials), and any trial where the gaze shifted away from the target by more than 20° between samples (which represent a velocity over 1,000° per second [14]; 154 trials). Finally, we removed trial completion time outliers for each participant and amplitude combination (51 trials). Overall, 1,701 (10.55%) of the trials from the 21 participants were removed, leaving a total of 14,427 trials.

6.8 Data Collection Measures

The following metrics were calculated for analysis. Some metrics relate to the overall task, some relate to each of the three individual input modalities (controller, head, and gaze), and some relate specifically to gaze data.

- *Trial Completion Time (Overall)*: Calculated as the total milliseconds between the trial start and successful selection of the goal target.
- *Angular Error (Head, Controller, and Gaze)*: The angular distance measured in degrees between the final directional vector and the target center, for each of the three input modalities.
- *Peak Velocity (Head, Controller, and Gaze)*: The maximum velocity (°/s) that each of the three input channels reach.
- *Peak Velocity Progress (Head, Controller, and Gaze)*: Measures when, in the trial, each input channel reaches its peak velocity, expressed as a percentage of the total trial time.
- *Number of Saccades (Gaze)*: The number of saccades that occurred in each trial.
- *Final Saccade Progress (Gaze)*: Measures when the final saccade completes, expressed as a percentage of the total trial time.

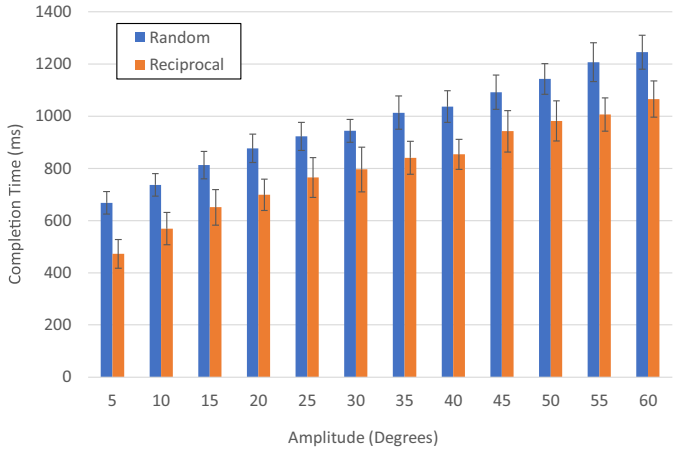


Fig. 13. Completion time by *Layout* and *Amplitude*.

6.9 Data Collection Results

For all dependent variables, an RM-ANOVA was performed, using the same Bonferroni correction procedure for *post-hoc* pairwise comparisons as Experiment 2. To perform a statistical comparison of dependent variables related to the gaze, controller, and head, an additional variable, *Modality,* was introduced.

6.9.1 Trial Completion Time. The grand mean for trial completion time was 889 ms. The RM-ANOVA found main effects of *Layout* ($F_{1,\ 20} = 144.0$, $p < .001$) and *Amplitude* ($F_{11,\ 220} = 279.3$, $p < .001$). There was no interaction between *Layout* and *Amplitude* ($F_{11,\ 220} = .98$, $p = .47$). Overall, reciprocal trials (803 ms) were faster than random trials (975 ms), and time increased with amplitude continuously from the range of 5 (570 ms) to 60 (1,155 ms), with all pairs being significantly different ($p < .05$) except for the pairs [20°, 25°], [25°, 30°], [35°, 40°], [50°, 55°], and [55°, 60°]. Figure 13 shows the impact of *Layout* and *Amplitude* on completion time.

6.9.2 Angular Error. When evaluating the angular error, the RM-ANOVA revealed a significant main effect of *Modality* ($F_{2,\ 40} = 860$, $p < .001$) and *Amplitude* ($F_{11,\ 200} = 193$, $p < .001$), but not *Layout* ($F_{1,\ 20} = 4$, $p = .59$). *Post-hoc* pairwise comparisons indicated that controller was most accurate ($M = .91°$), followed by gaze ($M = 1.40°$), and head ($M = 15.16°$), all at the $p < .001$ level. There was also a significant interaction effect between *Modality* and *Amplitude* ($F_{22,\ 440} = 103$, $p < .001$). What is interesting is that the angular error of gaze and controller remained constant across target amplitudes, while the accuracy of the head varied. For larger amplitudes, the head was less accurate during the reciprocal target layouts, indicating that the head may move less when the target is in a predictable location. Participants tended to move their heads anywhere from within 5°–22° of the final target for reciprocal *Layout*, and within 14°–17° for random *Layout* (Figure 14).

Scatterplots of the final landing positions further illustrate the angular error of each modality (Figure 15). The CPs were densely packed within the target boundaries. Gaze positions mirrored this, but with more divergence. The HPs diverged around the target for random *Layout*; however, for reciprocal *Layout,* banding suggested that the head moved in the direction of the target, like Study 1 (Figure 6). As such, integrating gaze into HeadKTM$_7$ may improve modeling, due to its increased accuracy compared to head movements.

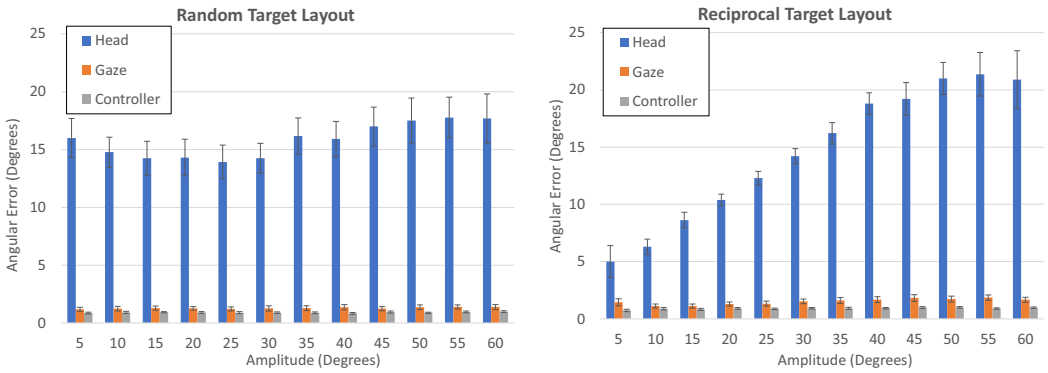


Fig. 14. Accuracy of the head, gaze, and controller for random *Layout* (left) and reciprocal *Layout* (right).

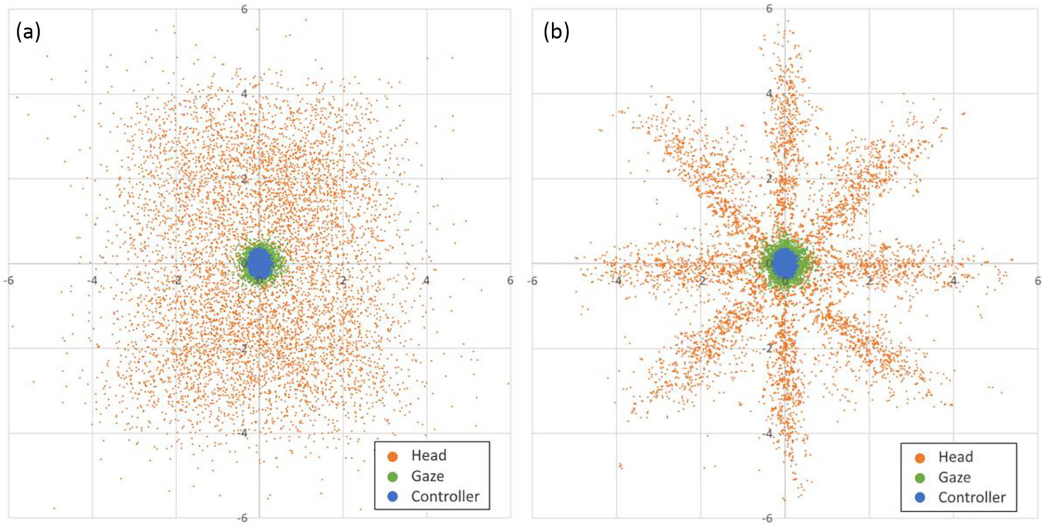


Fig. 15. Scatterplots of the final landing positions of the head, gaze, and controller for (a) random *Layout* and (b) reciprocal *Layout*.

6.9.3 Peak Velocity. The peak velocities are significantly impacted by *Modality* ($F_{2,\ 40} = 871$, $p < .001$), and highest for eye (463°/s), followed by controller (170°/s) and then head (34°/s) (all $p < .001$). Peak velocities are also significantly influenced by *Amplitude* ($F_{11,\ 220} = 728$, $p < .001$) (Figure 16). The values increase with angular amplitude, consistent with prior human factors studies [14]. The *Layout* also had a significant effect ($F_{1,\ 20} = 46.8$, $p < .001$) and there was significant interaction between *Modality* and *Layout* ($F_{2,\ 40} = 3.76$, $p < .05$) and between *Modality* and *Amplitude* ($F_{22,\ 440} = 210$, $p < .001$).

6.9.4 Peak Velocity Progress. In terms of *when* the peak velocity occurred, the RM-ANOVA found a significant main effect of *Modality* ($F_{2,\ 40} = 69.3$, $p < .001$), *Layout* ($F_{1,\ 20} = 14.9$, $p < .001$), and *Amplitude* ($F_{11,\ 220} = 46.7$, $p < .001$). There was also a significant interaction between *Modality* and *Layout* ($F_{2,\ 40} = 11.68$, $p < .001$) (Figure 17) and between *Modality* and *Amplitude* ($F_{22,\ 440} = 12.41$, $p < .001$). On average, the peak velocity is achieved first by the gaze, at 26.2% of the trial, then by the controller, at 33.9% of the trial, and finally by the head, at 36.5% of the trial. The pairwise

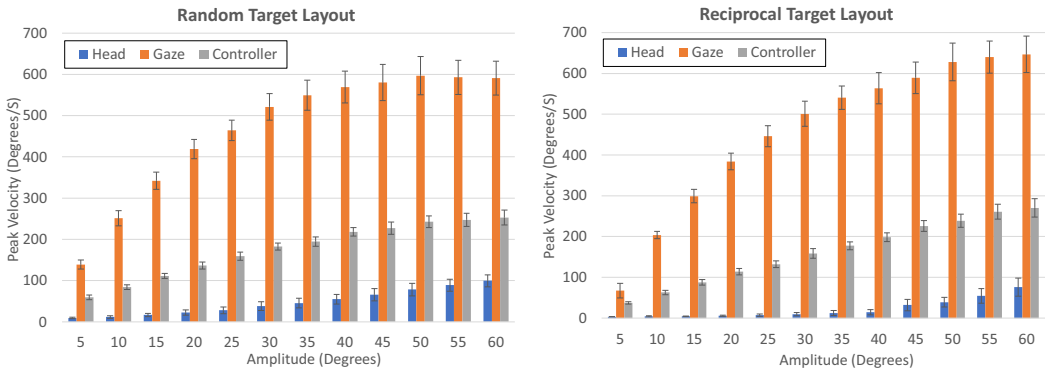


Fig. 16. The peak velocity for the head, gaze, and controller when a (left) random *Layout* and (right) reciprocal *Layout* was used.

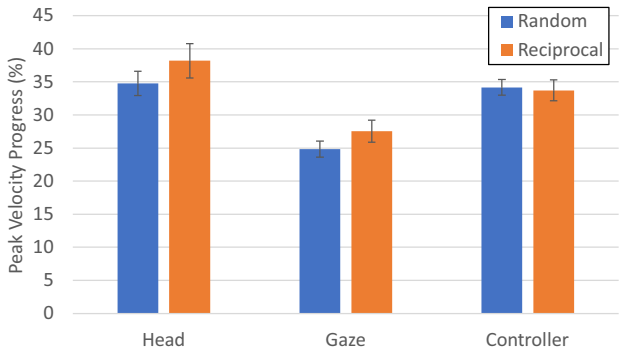


Fig. 17. The peak velocity progresses for each *Modality* and *Layout*.

differences between gaze, with both head and controller, were significant (both $p < .001$), while the pairwise difference between controller and head was not significant ($p = .13$). The finding shows that the gaze achieves its peak velocity sooner than both the head and the controller, at roughly one quarter of the total trial time, suggesting its potential use to provide earlier indications of final landing position within a predictive model.

6.9.5 Number of Saccades. The average number of saccades was 1.23 (SD = .82). In the majority of trials (71.7%) a single saccade occurred (Figure 18(a)). Both *Layout* ($F_{1,\ 20} = 65.8$, $p < .001$) and *Amplitude* ($F_{11,\ 220} = 199.5$, $p < .001$) had a significant effect on the number of saccades, and the interaction between *Layout* and *Amplitude* was significant ($F_{11,\ 220} = 25.1$, $p < .001$) (Figure 18(b)). The number of saccades increases with *Amplitude*, a trend more prominent with random trials.

6.9.6 Final Saccade Progress. On average the final saccade ended at 37.7% (SD = 10.5%) of the total trial time (Figure 19(a)). This result aligns with prior research on the coordination of the hand and point of gaze during aiming [48–50]. This is additional evidence that the gaze position could be used early, during a targeting action, to estimate a user's intended target location. The RM-ANOVA found that *Amplitude* had a significant effect on the final saccade progress ($F_{11,\ 220} = 29.5$, $p < .001$; Figure 19(b)). The *Layout* did not have a significant effect ($p = .26$); however there was a significant interaction between *Layout* and *Amplitude* ($F_{11,\ 220} = 13.4$, $p < .001$). For the reciprocal *Layout*, the

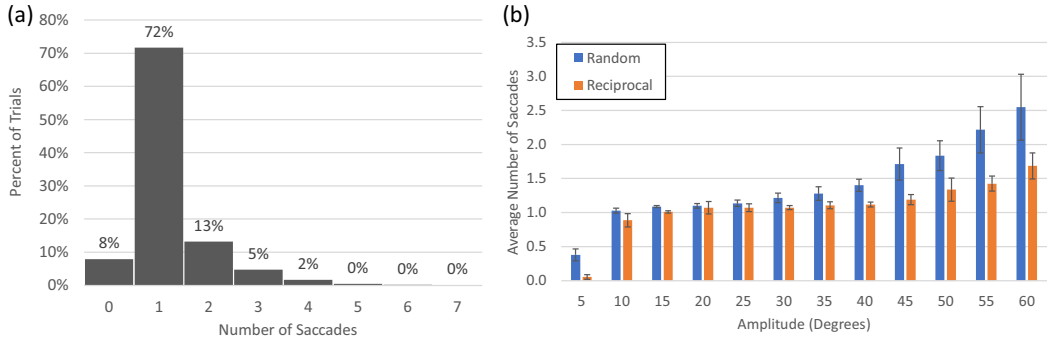


Fig. 18. (a) Histogram of the number of saccades per trial. (b) The average number of saccades by *Amplitude* and *Layout*.

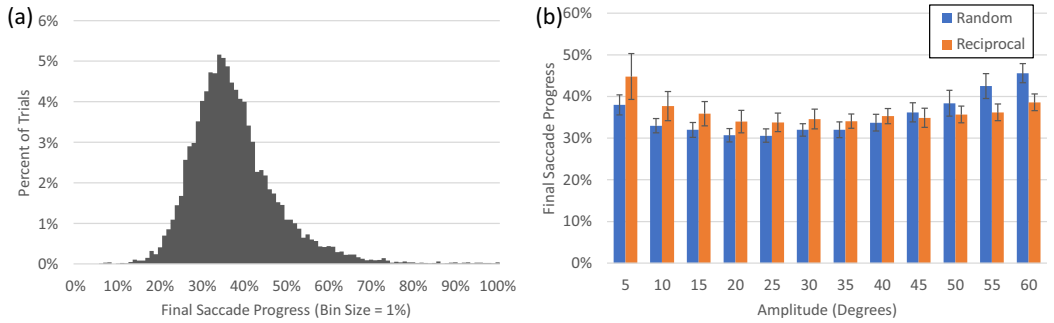
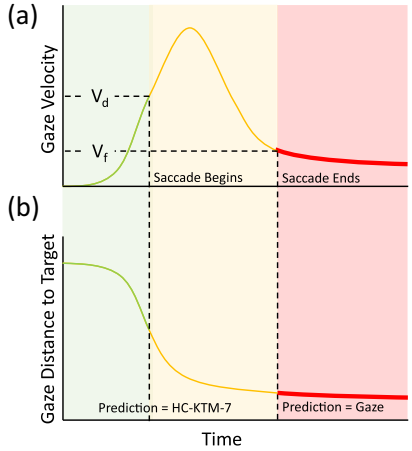


Fig. 19. (a) A histogram of trial progress when final saccade was completed. (b) The final saccade progress by *Layout* and *Amplitude*.

final saccade tended to end later for small and large *Amplitude* values, while for the random *Layout*, the final saccade tended to end later for just small *Amplitude* values.

6.9.7 Summary of Analysis. Overall, the analysis showed promise for using gaze as an additional input channel to predict a ray pointer's landing position, supplementing the controller and head movements. The results are largely consistent with prior work, which has shown that, in general, gaze precedes hand motions during acquisition (e.g., [50, 65, 101]), but this study provides validation for this finding, specifically, for 3D ray pointing selection in VR. The controller still had the most accurate landing position, but the final landing position of gaze more accurately reflected the position of the target (avg = 1.4°) compared to the HP (avg = 15.16°) (Section 6.9.2). Furthermore, gaze reached its peak velocity earlier than both the head and the controller (Section 6.9.3). This may indicate that gaze offers an earlier signal of intended landing position, which could potentially improve a template-based model. On average, the final saccade was completed at 37.7% of the total trial time (Section 6.9.6), at which point it should provide a good indication of the final cursor landing position. These results were generally consistent across both reciprocal and random target layouts; however, with random layouts, more saccades tended to occur (Section 6.9.5). This suggests that the accuracy of gaze-based predictive models may be reduced when the user cannot anticipate the target location.



Algorithm 1: HybridKTM$_7$ Model

```
gaze_armed = false                    (ignore the first fixation)
procedure SetPredictionRay(ctlr_ray, head_ray, gaze_ray)
    if not gaze_armed then
        predicted_ray = HC_KTM_7(ctlr_ray, head_ray)
    else if InSaccade(gaze_ray) then
        predicted_ray = HC_KTM_7(ctlr_ray, head_ray)
        gaze_armed = true
    else
        predicted_ray = gaze_ray
    end if
end procedure
```

Fig. 20. (a) A saccade began when the gaze velocity exceeded V_d and completed when the velocity dropped below V_f (Section 6.6). (b) The HybridKTM$_7$ model (Algorithm 1) transitions from using HeadKTM$_7$ prior to a saccade being completed to using the gaze ray after a saccade was completed (when the gaze should be closest to the intended target).

7 Gaze-Enhanced Models

Our data analysis from Experiment 3 indicates that integrating gaze data into the HeadKTM$_7$ model holds promise. We now describe our proposed gaze enhanced models and use then evaluate their performance, in comparison to HeadKTM$_7$.

7.1 GazeKTM$_7$ Model

The first proposed model, GazeKTM$_7$, utilizes the **Gaze Angular (GA)** velocity as a fifth input channel for template matching. The GA is calculated by averaging the gaze vectors of the left and right eyes. The origin of the gaze is equivalent to the *HP*, and thus, is not added to the template. With this model, the cumulative scoring function (Equation (1)) is extended to include the gaze angle score (S_{GA}), using a fifth constant weight, e (Equation (3)). All other aspects of the model are equivalent to HeadKTM$_7$. The same interpolation-based tuning procedure for e is used, as described in Section 4.6:

$$S(T_i) = aS_{CP}(T_i) + bS_{CA}(T_i) + cS_{HP}(T_i) + dS_{HA}(Ti) + eS_{GA}(T_i). \quad (3)$$

7.2 HybridKTM$_7$ Model

A second proposed model, HybridKTM$_7$, predicts the cursor landing position based on the current location of one's gaze (Figure 20, Algorithm 1). The rationale behind this model was that, if one's gaze lands close to a target prior to the cursor [108], then its absolute position could dictate the region close to where the ray pointer would land. Gaze data could, thus, indicate the target location, but only after a saccade is completed. Prior to a saccade, the predicted ray is based on the HeadKTM$_7$ model, but after the saccade is completed, the predicted ray is based on the current gaze ray. Note that in some cases, a second saccade might be needed to visually acquire a target. If HybridKTM$_7$ detects a second saccade beginning, it reverts to the HeadKTM$_7$ model and waits for the second saccade to complete before using the gaze ray again.

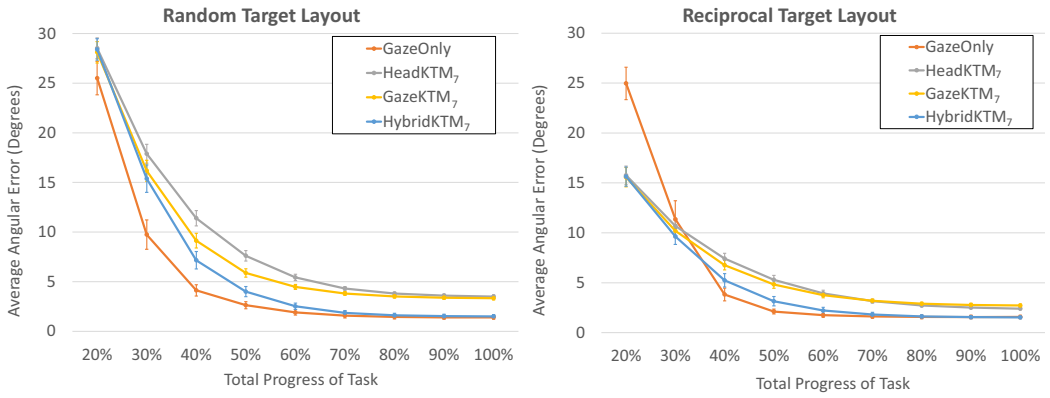


Fig. 21. A comparison of the four models, for (left) random target layouts and (right) reciprocal target layouts.

7.3 Modeling Results

Using the data collected from Experiment 3, we now validate these proposed gaze-enhanced models. We compare the following four models:

(1) *GazeKTM*$_7$: An extension of the HeadKTM$_7$ model, which included the velocity of the gaze forward vector as a fifth input channel (Section 7.1).
(2) *HybridKTM*$_7$: A version of HeadKTM$_7$ that transitioned to a prediction of the cursor landing position based on the current location of the gaze after a saccade has occurred (Section 7.2).
(3) *GazeOnly*: A model that uses gaze only, based entirely on the current position of the gaze vector, without any additional prediction applied.
(4) *HeadKTM*$_7$: A baseline model that did not utilize any gaze information (Section 3). This was found to be the best performing head-coupled model in Experiment 2.

All models were tested offline, using the data recorded during Experiment 3. For the evaluation of the template-based models, we evaluated each participant's recorded data against their own templates for the random and reciprocal target layouts separately. The same interpolation procedure from Section 4.6 was used to determine weighting values for the scoring functions in the template-based models (*a*, *b*, *c* and *d*, and *e* in Equation (3)). An RM-ANOVA was performed for random and reciprocal trials separately to compare the angular error of the four models at 10% intervals of task progress points. A Bonferroni correction was applied to all *post-hoc* pairwise comparisons, by multiplying the (uncorrected) p-values by the number of comparisons made [2]. Across all trials, the average movement time was 889 ms, so a prediction at 40% progress would be at the 356 ms mark, on average.

The results are illustrated in Figure 21. For both random and reciprocal trials, there were overall significant effects of model (random: $F_{3,\ 60} = 412.1$, $p < .001$; reciprocal: $F_{3,\ 60} = 33.5$, $p < .001$) and progress point (random: $F_{8,\ 160} = 1374.6$, $p < .001$; reciprocal: $F_{8,\ 160} = 752.3$, $p < .001$) on the angular error. There was also a significant interaction effect between model and task progress for both random ($F_{24,\ 480} = 58.8$, $p < .001$) and reciprocal ($F_{24,\ 480} = 71.6$, $p < .001$) trials.

When comparing just the template-based models (i.e., HeadKTM$_7$, GazeKTM$_7$, and HybridKTM$_7$), the results indicated that HybridKTM$_7$ outperformed the baseline HeadKTM$_7$ model and GazeKTM$_7$ for both random and reciprocal trials (all $p < .001$). This indicates that HybridKTM$_7$ would be the best template-based predictive model. For random layouts, at 40% progress, HybridKTM$_7$ had an angular error of 7.2°, outperforming the baseline HeadKTM$_7$ model (11.4°) by 37.2%. For

reciprocal layouts, the improvement was 29.3% (HybridKTM$_7$: 5.2°, HeadKTM$_7$: 7.4°). We also saw that all models performed better when reciprocal target layouts were used rather than random target layouts, especially earlier in a trial. This is likely because the target locations were more predictable, and, thus, the participants' movements were more uniform, which would improve the template-based model accuracies.

We next turn our attention to the GazeOnly model. For random target layouts, GazeOnly had the lowest overall error ($p < .001$) and outperformed HybridKTM$_7$ across the entire progress of the trials. At 40% progress, GazeOnly had an angular error of 4.1°, while HybridKTM$_7$ had an angular error of 7.2° (42.4% improvement). HybridKTM$_7$ had the lowest overall error for reciprocal trials ($p < .001$), but this is due to poor results for GazeOnly earlier in the trial. HybridKTM$_7$ was more accurate earlier in the movement (at 20% and 30% progress). At 20% progress, the errors were 15.6° for HybridKTM$_7$ and 25.0° for GazeOnly, a 37.4% improvement. A crossover occurred between the 30% and 40% progress mark, when GazeOnly had lower error from 40% to 60% progress. After 70% progress, the two models performed comparably well, with angular errors less than 2° for random and reciprocal layouts.

Overall, our modeling showed that gaze-enhanced models had observable improvements over the baseline model. The GazeOnly model may be the best option for landing position prediction, especially in interfaces that a user is less familiar with, which may induce behaviors similar to those in the random layout. For interfaces that users are more familiar with, the behaviors may be closer to those from the reciprocal layout, in which case HybridKTM$_7$ may provide better results, especially if predictions earlier in the motion are desirable. However, future work is needed to explore this further, as the reciprocal task also involves preplanned back and forth muscle movements which may not be a true representative of real-world interface usage behaviors.

8 Implications to Design and Practical Applications

In this section, we outline important considerations for real-world implementations, and then discuss two potential use cases of endpoint prediction: selection facilitation and activation latency reduction.

8.1 Beyond Abstract Targeting Environments

Our experiments were conducted in controlled and abstract targeting environments with a known start time for each targeting movement (e.g., when each trial began). In a real-life application, the start of a targeting movement would not be explicitly known. As such, it is important to discuss techniques to determine when to begin prediction, or when a targeting motion has reached a progress threshold, such as the 40% progress point.

When Does a New Movement Begin? Interpreting when a targeting movement begins for mid-air input may be challenging, because sensor input is constantly updating (unlike 2D mouse input, where there is a resting state). One approach could be to classify individual samples as being part of a targeting state or nontargeting state, based on the gaze, head, and controller trajectories. Similar approaches have proved effective to distinguish stroke vs. hover movements in mid-air drawing applications [25]. This would allow our endpoint prediction models to define the beginning of a new targeting trajectory (e.g., 0% progress) and filter out movements unrelated to targeting. The first step toward such a goal would be conducting a data collection study for a task that combines targeting with other nontargeting related actions. This training data could then be used to develop a binary classifier.

When to Start Predicting? Even if the beginning of a targeting movement could be identified (as per above), the percent progress toward the endpoint would still be unknown. In Experiment 3, mean targeting times ranged from 570 ms ($Theta = 5°$) to 1,155 ms ($Theta = 60°$), meaning the 40%

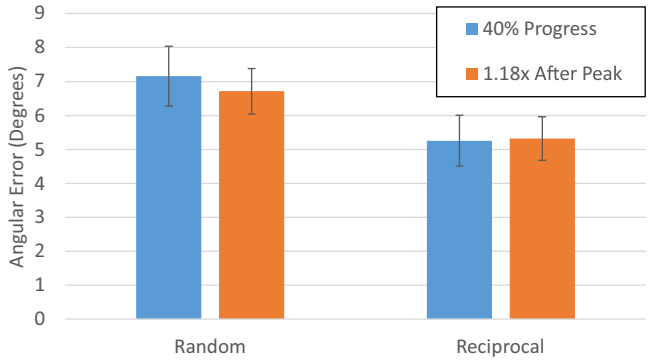


Fig. 22. Comparing the angular error of HybridKTM$_7$ at 40% progress (known only after the trial completes) and at the progress point calculated by multiplying the time at which peak velocity occurs (which can be detected during the movement) by 1.18.

progress mark could range from 228 ms and 462 ms. If predictions were applied too early, then they may not meet the desired accuracy level, whereas if predictions were applied too late, the benefit of endpoint prediction may be lost. One approach to address this would be to estimate pointing progress based on the kinematics of the velocity profiles [77]. In particular, our analysis of peak velocities (Figure 17) revealed that both the head and controller velocities tended to peak at around the 35% progress mark.

The validate this concept, we conducted a follow-up analysis based on the controller peak velocity. We used the controller velocity, as it had a smaller level of variation in its mean progress point (33.9%). Once the controller peak velocity occurs, the 40% progress mark can be estimated by multiplying the time at which it occurs by 1.18 (40.0/33.9). For example, if the controller velocity peaks at 300 ms, we assume 40% will be reached at 354 ms. We used this method to calculate the angular error of our top performing template-based model (HybridKTM$_7$) (Figure 22). For both random and reciprocal trials, the angular errors are comparable to those produced by taking angular errors at the 40% progress point, and the RM-ANOVA revealed no statistical difference ($p = .51$). As such, this may be a feasible method to determine when to start predicting in real-world scenarios, when the full trial time is not already known. However, it is important to note that the progress point at which peak velocity occurs (e.g., the 1.18 multiplier we used) may be dependent on the device, task, user, or application. As such, initial data gathering may be necessary before such real-time predictions can be made.

8.2 Use Case: Selection Facilitation Techniques

An exciting line of future research is to combine selection facilitation techniques [44, 63] with our endpoint prediction models. This could potentially improve selection of small and distant targets, which is an inherent challenge of ray pointing [69]. Indeed, one of the core promises of endpoint prediction more broadly, is to facilitate selection:

> "If we could create a means of knowing an endpoint in advance of its delivery by a mouse-click, we could increase the efficiency of mouse pointing, perhaps considerably, with techniques such as target expansion [73, 95] or gravity wells [47]. Such is the goal of *endpoint prediction*, an attempt to predict the future when pointing." [87, p. 743]

For example, techniques that dynamically adapt the CD ratio [23] could benefit from early prediction. As the user initially moves the cursor, such predictions could enable the cursor to

accelerate toward the predicted region and decelerate when it arrives. This type of facilitation may be particularly suitable for HybridKTM$_7$, as it performs well even at earlier stages of movement (e.g., 40%), In this case, the predicted landing position would not need to be precisely located at the intended target, as there would be benefit from the cursor accelerating toward the general target region. The user could refine the position of the cursor toward the goal region as the motion continues during closed-loop corrective movements [75]. One important consideration for such a technique is how the dynamic CD gain would impact the user's movements. Dynamically adjusting the CD gain in unpredictable ways could hinder, rather than assist, overall performance. We would argue that cursor acceleration, a default in most mouse-based systems today [29], has shown that dynamically changing CD gains can be beneficial, even if the actual mapping is not well understood by the end-users. This would be an interesting topic for future studies.

Alternatively, target snapping or cursor bending (e.g., [44, 86, 102, 106]) could benefit from early predictions—instead of just snapping to the closest target to the cursor, the technique could snap to the closest target in the predicted region. This could support faster access to targets when predictions are made during the initial stages of the movement. Another technique which could leverage endpoint prediction is target expansion, where knowing which target to expand is one of the main challenges [74]. Prior work has shown that even if expansion occurs at 90% progress, target expansion can improve selection [73]. As such, it is reasonable to believe a prediction at 40% progress could certainly improve the selection process.

If the target layout is known, probability distributions could be generated across the targets [114], and any facilitation technique could be disabled if the probability does not reach a threshold value. Future studies should be conducted to understand the performance of such techniques as a function of the target layout. It is also vital to understand how each of these techniques would behave in instances when the prediction is wrong. Care would need to be taken in the design of such techniques to ensure users could easily ignore or override incorrect predictions.

It is also important to note that any facilitation technique is likely to change the user's behaviors due to the target acquisition perception and action loop [75]. In particular, there is a risk that changing the mechanics of the cursor behavior in the midst of a targeting action could negatively influence a user's pointing action by breaking the continuous nature of cursor pointing. We would hypothesize the best performing facilitation techniques would be those that maintain continuous and predictable movements (e.g., bubble cursor [44, 71]; cursor acceleration [23]) rather than techniques like target jumping which could introduce unexpected or unpredictable movements.

8.3 Use Case: Reducing Activation Latency

Another application of endpoint prediction is to reduce perceived latency (e.g., activation delay) of interaction [110] or of computationally heavy operations. This has been a long sought after goal in operating systems to improve user experience (e.g., Microsoft's Superfetch [55]). This could be particularly beneficial in VR systems where executed commands could perform computationally heavy operations. If we take the average selection time from Experiment 3 (889 ms), beginning a predicted action at 40% progress would decrease latency by 533 ms ($.6 \times 889$ ms), which could substantially improve the perceived system responsiveness (prior work has shown latencies of even 100 ms are perceivable to the user and can be detrimental to user experience [7, 59, 83]).

9 Discussion and Future Work

We now summarize our main findings, list avenues for future research, and discuss our work's limitations.

9.1 Summary of Findings and Recommendations

The presented models show promise for VR ray pointer predictions due to the introduction of multimodal input channels in the KTM model. The HeadKTM$_7$ model, which incorporated HP and CP angles, demonstrated the initial validation of this concept, while the gaze-enabled models showed additional accuracy benefits. Below we highlight the critical findings across our three studies and provide recommendations.

In Experiment 1, we found that head movements, while less accurate than controller movements (Figure 6), may provide a better signal of the intended target in the early stages of movement. Indeed, modeling from Experiment 1 showed that relying solely on HA for endpoint prediction was better than relying solely on the CA (Figure 8). In Experiment 2, we validated that incorporating HA into the template matching model improved prediction accuracies, with the HeadKTM$_7$ model performing best overall. As such, our recommendation is to use HeadKTM$_7$ for target prediction when gaze tracking is not available.

In Experiment 3, we investigated the impact of incorporating gaze into the endpoint prediction model. We found that gaze reached its peak velocity earlier than both the head and the controller. The results of the modeling showed that both proposed gaze-enhanced models (GazeKTM$_7$ and HybridKTM$_7$) outperformed HeadKTM$_7$. As such, we recommend that if gaze tracking is available, then a gaze-enhanced model should be used for prediction. Furthermore, we found that a simple GazeOnly model may be the best option, although there were situations (e.g., early during reciprocal layouts) where the GazeOnly performed poorly, and HybridKTM$_7$ model performed best (Figure 21). Future work in real-world environments should be conducted to better understand the tradeoffs between these two techniques.

9.2 Generalizations to Other Platforms

Although developed and evaluated within a VR environment, the models should generalize to 2D platforms as well. Targets were shown in 3D space, but the task could be decomposed into the 2D angular movements of the ray pointer. It would be interesting to use the model for distant pointing on large, high-resolution displays, where 2D angular ray pointing is also used. Within this context, the present work can be seen as building on prior literature, which has coupled head and hand movements to divide large display pointing into coarse and precise modes [80]. Similarly, our models could be useful in traditional desktop environments [66, 87, 114], or mobile devices, where the onboard camera could be used for head and gaze tracking [67].

9.3 Personalization of Template Libraries

One characteristic of the models we developed is that they are personalized to individual users. An advantage of model personalization is that it can be tuned to each user; a drawback is that training data is needed. One solution is to start a new user with a generic template library, and slowly replace that library with the user's own data, as movements are collected. There may also be classes of users with similar behaviors, who could share predetermined template libraries. For example, users could be classified based on the extent with which they tend to move their head or on gaze behavior patterns [101]. Future research should be conducted to explore these topics.

9.4 Complexity

An important factor of our model is the performance it would achieve in a real-time environment. The real-time performance would be dependent on the number of templates that are in a user's template library. Experiment 2 used roughly 2,000 templates per participant, whereas fewer than 400 were used for Experiment 3. We have conducted an initial demonstration of the performance

of the HeadKTM$_7$ model in real time with a sample two-player VR shooting game [51]. The game included a "power-up" that enabled target snapping based on the model predictions. The game displayed both the actual ray cursor and a secondary ray cursor that snapped to the predicted targets. That way, if the prediction was incorrect, it could be ignored, and the user could still control their original ray cursor in a continuous manner. The implementation involved approximately 2,000 templates. The model was able to run in real-time, with minimal optimization, no perceivable impact on performance, and at an input rate of 90 Hz. With each incoming input event frame, a prediction occurred in just under 11 ms.

9.5 Model Considerations

Our work is one of the first efforts to adapt endpoint prediction techniques to VR ray pointing. We were motivated to leverage the KTM model for our work, given its promising results in the 2D desktop pointing literature [87]. However, as reviewed in Section 2.2, there are many alternative models that have been previously used for endpoint prediction. Below we contrast our algorithm with alternative endpoint prediction models, discuss how such previous models could be adapted to VR ray pointing, and contrast their relative strengths and applicable scenarios.

Extrapolation Using Motion Kinematics. Techniques that use basic linear regression (e.g., doubling distance of movement at peak velocity [85]) or extrapolation based on motion kinematics (e.g., minimum jerk law [66]) have the benefit of being simple to implement, and run efficiently, and could require less training data then the KTM approach which we used. There may be future opportunities to adapt such techniques to ray pointing using multimodal input channels. For example, independent extrapolation models could be developed to map cursor, head, and gaze velocities to target distance. A weighted average of these three distance calculations could then determine the final endpoint estimate. However, regression-based extrapolation models have been shown to be less accurate than KTM [87], so we would hypothesize that such models should only be considered if runtime efficiency or implementation cost was being prioritized.

OFC Models. Models based on OFC have received recent attention in the HCI literature, and have shown to be able to predict motion trajectories, given known endpoints/targets, for both 2D pointing [39] and 3D mid-air movements [12]. While these techniques show great promise, they are not designed to predict the user's goal state (e.g., the target location). Conversely, they are used to predict trajectories given an intended goal. As such, these models may be useful in obtaining a better understanding and explanation of observed behaviors, but they are not directly applicable for endpoint prediction.

Inverse Optimal Control Methods. In contrast, inverse optimal control models *can* be used to predict the user objective based on behavioral observations. More specifically, given a pointing trajectory, they could be used to infer an endpoint. Ziebart et al. [114] used this principle to predict the desired target of a partial pointing motion for desktop mouse pointing. Results indicated that when 40–60% of the pointing motion remained, the inverse optimal control method outperformed the extrapolation methods described above. However, the technique requires a higher level of complexity for implementation, especially if it were to be adapted for a 3D ray pointing task. Furthermore, the technique by Ziebart et al. relies on the preexisting knowledge of the potential target locations. As such, the technique may be suitable for ray pointer prediction when implementation cost is not a concern, and the potential target locations are known. Models based on inverse optimal control also have the advantage that they can predict entire motion trajectories, not just the final endpoint. This could be useful when full trajectory data may be needed, such as for gesture-based input, or predicting tunneling or steering movements [4]. Adapting such techniques to 3D ray pointing was beyond the scope of our own research. Future research should investigate the applicability of

inverse optimal control models for ray pointing prediction in VR, comparing its performance to the benchmark results which we have established.

9.6 Limitations

Our work tested models with data that were collected in controlled lab experiments, in abstract target environments. While this is standard practice for HCI research to improve internal validity, it also gives rise to important limitations related to external validity. For example, Study 3 revealed that changing the task (between reciprocal and random) could impact the performance of the models. Further work is needed to understand how the results would generalize to actual interface usage. Our hypothesis is that the random target layout should be a good approximation of real-world behaviors, but further validation is needed. Another aspect of VR pointing, that was not addressed, is that multiple targets can be located along the same projected path at varying depths, thereby requiring disambiguation. The proposed models only predict the location of the ray, not the depth of the target. Future work should explore additional input channels, such as gaze [108], to predict object depth and extend the model to truly 3D predictions. Selection refinement techniques [45, 63] could also be used when multiple targets fall along the ray. Another simplification was the lack of distractor targets. The visual presence of distractors could influence a user's behavior, which could in turn interfere with the model. Incorporating or filtering out such behaviors would be an interesting topic for future studies. Furthermore, our study removed visual search from the task requirement. Future work could attempt to infer when a user transitions from visual search to target acquisition, to define the beginning of a new candidate movement.

In addition, the controlled studies of our work assumed a continuous path to the goal target which may not always be present in a real-world application. If a user were to change their intended target mid-movement, the template-based models would likely break down. For example, if a user was performing a 3D sketch in VR, their intention regarding where to begin their next stroke may change at any time during a movement. As such, their movement would no longer be comparable to a library of templates which consist of single continuous movements. Our approach applies the predicted angular distance to the current angle of movement, defined by the angle between the current controller forward vector and its initial forward vector. An alternative approach would be to define the current angle of movement based on a window of recent samples; however, samples would need to be filtered to account for noise in the spatial data. Beyond this, a classifier may be needed to detect when such intent changes occur, at which point any attempts for prediction are abandoned. Alternatively, the template matching procedure could be reinitialized from the exact point at which the intent changes, if it could be identified (perhaps at an inflection point in the velocity profile).

In relation to the parameter values of our model (e.g., a–e in Equation (3)), we used a straightforward interpolation procedure to weigh the relative importance of each velocity profile. Future work could explore more advanced optimization techniques, which could validate the range of values we used and their relative ranking.

In Experiment 3, the gaze data had calibration errors and noise, resulting in some trials being removed from the analysis. As we expect future headsets to provide more reliable gaze data, we took a liberal approach to removing outliers, so the data were as clean as possible, and thus, best simulated the performance of the models when gaze tracking improves. Future work should consider adaptive techniques to transition between models based on the reliability of incoming data [99].

In all three studies, we used angular divergence from the goal target as the metric for accuracy. We chose this metric as it provides a single representative metric that is analogous to pixel distance used in 2D endpoint prediction studies. However, the model predicts both a Cartesian position of

the controller and its forward vector angle. By using a single metric, we do lose some information about the root cause of any imprecision. Even if the angle is perfectly calculated, our metric of angular error could be nonzero due to inaccuracy in the position prediction. More advanced metrics of error could be considered in the future, such as looking at each of the individual components of the predicted ray position and angle separately. Alternatively, the Cartesian distance between the predicted ray and the actual endpoint could be used; however, such a metric would be sensitive to the distance of the target from the user (the further the target is, the larger the distance would be).

10 Conclusion

This work demonstrated that multimodal data can be used to improve predictions of ray pointer landing positions. Specifically, we found that it is beneficial to integrate the movements of both the head and gaze patterns into the predictive models. We first showed that a model that utilized head movements (i.e., HeadKTM$_7$) provided an angular error of 10.0° at 40% of the way through a movement, representing an increased prediction accuracy by 44.1% compared to the existing 2D KTM model. We then showed that models that incorporate gaze information (i.e., GazeKTM$_7$, HybridKTM$_7$, and GazeOnly) can further enhance the predictions. In particular, the HybridKTM$_7$ model provided additional improvements with angular errors of 5.2° for reciprocal target layouts and 7.2° for random target layouts at 40% progress in comparison to the HeadKTM$_7$ model. Furthermore, we found that simply using gaze position can be less accurate early in a movement, but performs best overall at 40% progress, with average angular errors of 4.1° for random target layouts and 3.8° for reciprocal target layouts. Our hope is that our findings related to pointer prediction in VR will open the door for future enhancements to 3D user experiences.

References

[1] Eye Tracking on Meta Quest Pro. Retrieved November 21, 2023 from https://www.meta.com/help/quest/articles/getting-started/getting-started-with-quest-pro/eye-tracking/

[2] The Calculation of Bonferroni-Adjusted p-Values. Retrieved November 28, 2023 from https://www.ibm.com/support/pages/calculation-bonferroni-adjusted-p-values

[3] VIVE Pro Eye Features. Retrieved November 21, 2023 from https://www.vive.com/sea/product/vive-pro-eye/overview/

[4] Johnny Accot and Shumin Zhai. 2002. More than dotting the i's — Foundations for crossing-based interfaces. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '02)*. ACM, New York, NY, 73–80. DOI: https://doi.org/10.1145/503376.503390

[5] Bashar I. Ahmad, Patrick M. Langdon, Simon J. Godsill, Richard Donkor, Rebecca Wilde, and Lee Skrypchuk. 2016. You do not have to touch to select: A study on predictive in-car touchscreen with mid-air selection. In *Proceedings of the 8th International Conference on Automotive User Interfaces and Interactive Vehicular Applications (Automotive'UI 16)*. ACM, New York, NY, 113–120. DOI: https://doi.org/10.1145/3003715.3005461

[6] Richard Andersson, Linnea Larsson, Kenneth Holmqvist, Martin Stridh, and Marcus Nyström. 2017. One algorithm to rule them all? An evaluation and discussion of ten eye movement event-detection algorithms. *Behavior Research Methods*, 49 (2017), 616–637. DOI: https://doi.org/10.3758/s13428-016-0738-9

[7] Michelle Annett, Albert Ng, Paul Dietz, Walter F. Bischof, and Anoop Gupta. 2014. How low should we go? Understanding the perception of latency while inking. In *Proceedings of Graphics Interface 2014 (GI '14)*. Canadian Information Processing Society, CAN, 167–174. DOI: https://dl.acm.org/doi/10.5555/2619648.2619677

[8] Elena Arabadzhiyska, Okan Tarhan Tursun, Karol Myszkowski, Hans-Peter Seidel, and Piotr Didyk. 2017. Saccade landing position prediction for gaze-contingent rendering. *ACM Transactions on Graphics* 36, 4, Article 50 (July 2017), 12 pages. DOI: https://doi.org/10.1145/3072959.3073642

[9] Ferran Argelaguet and Carlos Andujar. 2013. A survey of 3D object selection techniques for virtual environments. *Computers & Graphics*, 37, 3 (2013), 121–136. DOI: https://doi.org/10.1016/j.cag.2012.12.003

[10] Takeshi Asano, Ehud Sharlin, Yoshifumi Kitamura, Kazuki Takashima, and Fumio Kishino. 2005. Predictive interaction using the Delphian desktop. In *Proceedings of the 18th Annual ACM Symposium on User Interface Software and Technology (UIST '05)*. ACM, New York, NY, 133–141. DOI: http://dx.doi.org/10.1145/1095034.1095058

[11] Gökçen Aslan Aydemir, Patrick M. Langdon, and Simon Godsil. 2013. User target intention recognition from cursor position using Kalman filter. In *Proceedings of the International Conference on Universal Access in Human-Computer Interaction*. Springer, Berlin, 419–426. DOI: https://10.1007/978-3-642-39188-0_45

[12] Myroslav Bachynskyi and Jörg Müller. 2020. Dynamics of aimed mid-air movements. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems (CHI '20)*. ACM, New York, NY, 1–12. DOI: https://doi.org/10.1145/3313831.3376194

[13] Thomas Bader, Matthias Vogelgesang, and Edmund Klaus. 2009. Multimodal integration of natural gaze behavior for intention recognition during object manipulation. In *Proceedings of the 2009 International Conference on Multimodal Interfaces (ICMI-MLMI '09)*. ACM, New York, NY, 199–206. DOI: https://doi.org/10.1145/1647314.1647350

[14] A. Terry Bahill, Michael R. Clark, and Lawrence Stark. 1975. The main sequence, a tool for studying human eye movements. *Mathematical Biosciences*, 24, 3–4 (1975), 191–204. DOI: https://doi.org/10.1016/0025-5564(75)90075-9

[15] Marc Baloup, Thomas Pietrzak, and Géry Casiez. 2019. RayCursor: A 3D pointing facilitation technique based on Raycasting. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems (CHI '19)*. ACM, New York, NY, 12 pages. DOI: https://doi.org/10.1145/3290605.3300331

[16] Patrick Baudisch and Ruth Rosenholtz. 2003. Halo: A technique for visualizing off-screen objects. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '03)*. ACM, New York, NY, 481–488. DOI: https://doi.org/10.1145/642611.642695

[17] Xiaojun Bi, Yang Li, and Shumin Zhai. 2013. FFitts law: Modeling finger touch with Fitts' law. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '13)*. ACM, New York, NY, 1363–1372. DOI: https://doi.org/10.1145/2470654.2466180

[18] Xiaojun Bi and Shumin Zhai. 2013. Bayesian touch: A statistical criterion of target selection with finger touch. In *Proceedings of the 26th Annual ACM Symposium on User Interface Software and Technology (UIST '13)*, 51–60. DOI: http://dx.doi.org/10.1145/2501988.2502058

[19] Xiaojun Bi and Shumin Zhai. 2016. Predicting finger-touch accuracy based on the dual Gaussian distribution model. In *Proceedings of the 29th Annual Symposium on User Interface Software and Technology (UIST '16)*. ACM, New York, NY, 13–319. DOI: https://doi.org/10.1145/2984511.2984546

[20] Hans-Joachim Bieg, Lewis L. Chuang, Roland W. Fleming, Harald Reiterer, and Heinrich H. Bülthoff. 2010. Eye and pointer coordination in search and selection tasks. In *Proceedings of the 2010 Symposium on Eye-Tracking Research & Applications (ETRA '10)*. ACM, New York, NY, 89–92. DOI: http://dx.doi.org/10.1145/1743666.1743688

[21] Pradipta Biswas, Gokcen Aslan Aydemir, Pat Langdon, and Simon Godsill. 2013. Intent recognition using neural networks and Kalman filters. In *Proceedings of the International Workshop on Human-Computer Interaction and Knowledge Discovery in Complex, Unstructured, Big Data*. Springer, Berlin, 112–123. DOI: https://10.1007/978-3-642-39146-0_11

[22] Pradipta Biswas and Patrick M. Langdon. 2014. Multi-modal target prediction. In *Proceedings of the International Conference on Universal Access in Human-Computer Interaction*. Springer, Cham, 313–324. DOI: https://doi.org/10.1007/978-3-319-07437-5_30

[23] Renaud Blanch, Yves Guiard, and Michel Beaudouin-Lafon. 2004. Semantic pointing: Improving target acquisition with control-display ratio adaptation. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '04)*. ACM, New York, NY, 519–526. DOI: https://doi.org/10.1145/985692.985758

[24] Renaud Blanch and Michaël Ortega. 2009. Rake cursor: Improving pointing performance with concurrent input channels. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '09)*. ACM, New York, NY, 1415-1418. DOI: https://doi.org/10.1145/1518701.1518914

[25] Umema Bohari, Ting-Ju Chen, and Vinayak. 2018. To draw or not to draw: Recognizing stroke-hover intent in non-instrumented gesture-free mid-air sketching. In *Proceedings of the 23rd International Conference on Intelligent User Interfaces (IUI '18)*. ACM, New York, NY, 177–188. DOI: https://doi.org/10.1145/3172944.3172985

[26] Doug A. Bowman and Larry F. Hodges. 1997. An evaluation of techniques for grabbing and manipulating remote objects in immersive virtual environments. In *Proceedings of the 1997 Symposium on Interactive 3D Graphics (I3D '97)*. ACM, New York, NY, 35–ff. DOI: http://dx.doi.org/10.1145/253284.253301

[27] Juan Sebastian Casallas, James H. Oliver, Jonathan W. Kelly, Frederic Merienne, and Samir Garbaya. 2014. Using relative head and hand-target features to predict intention in 3D moving-target selection. In *Proceedings of the 2014 IEEE Virtual Reality (VR)*. IEEE, 51–56. DOI: https://10.1109/VR.2014.6802050

[28] Elie Cattan, Amélie Rochet-Capellan, Pascal Perrier, and François Bérard. 2015. Reducing latency with a continuous prediction: Effects on users' performance in direct-touch target acquisitions. In *Proceedings of the 2015 International Conference on Interactive Tabletops & Surfaces (ITS '15)*. ACM, New York, NY, 205–214. DOI: https://doi.org/10.1145/2817721.2817736

[29] Géry Casiez and Nicolas Roussel. 2011. No more bricolage! Methods and tools to characterize, replicate and compare pointing transfer functions. In *Proceedings of the 24th Annual ACM Symposium on User Interface Software and Technology (UIST '11)*. ACM, New York, NY, 603–614. DOI: https://doi.org/10.1145/2047196.2047276

[30] Di Laura Chen, Marcello Giordano, Hrvoje Benko, Tovi Grossman, and Stephanie Santosa. 2023. GazeRayCursor: Facilitating virtual reality target selection by blending gaze and controller raycasting. In *Proceedings of the 29th ACM Symposium on Virtual Reality Software and Technology (VRST '23)*. ACM, New York, NY, Article 19, 1–11. DOI: https://doi.org/10.1145/3611659.3615693

[31] Mon Chu Chen, John R. Anderson, and Myeong Ho Sohn. 2001. What can a mouse cursor tell us more? Correlation of eye/mouse movements on web browsing. In *Proceedings of the CHI '01 Extended Abstracts on Human Factors in Computing Systems (CHI EA '01)*. ACM, New York, NY, 281–282. DOI: https://doi.org/10.1145/634067.634234

[32] Lung-Pan Cheng, Eyal Ofek, Christian Holz, Hrvoje Benko, and Andrew D. Wilson. 2017. Sparse haptic proxy: Touch feedback in virtual environments using a general passive prop. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems (CHI '17)*. ACM, New York, NY, 3718–3728. DOI: https://doi.org/10.1145/3025453.3025753

[33] Nathan Cournia, John D. Smith, and Andrew T. Duchowski. 2003. Gaze- vs. hand-based pointing in virtual environments. In *Proceedings of the CHI '03 Extended Abstracts on Human Factors in Computing Systems (CHI EA '03)*. ACM, New York, NY, 772–773. DOI: http://dx.doi.org/10.1145/765891.765982

[34] Brendan David-John, Candace Peacock, Ting Zhang, T. Scott Murdison, Hrvoje Benko, and Tanya R. Jonker. 2021. Towards gaze-based prediction of the intent to interact in virtual reality. In *Proceedings of the ACM Symposium on Eye Tracking Research and Applications (ETRA '21 Short Papers)*. ACM, New York, NY, Article 2, 1–7. DOI: https://doi.org/10.1145/3448018.3458008

[35] Shujie Deng, Jian Chang, Julie A. Kirkby, and Jian J. Zhang. 2016. Gaze–mouse coordinated movements and dependency with coordination demands in tracing. *Behaviour & Information Technology* 35, 8 (Aug. 2016), 665–679. DOI: http://dx.doi.org/10.1080/0144929X.2016.1181209

[36] Seungwon Do, Minsuk Chang, and Byungjoo Lee. 2021. A simulation model of intermittently controlled point-and-click behaviour. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems (CHI '21)*. ACM, New York, NY, Article 286, 1–17. DOI: https://doi.org/10.1145/3411764.3445514

[37] Ribel Fares, Dustin Downing, and Oleg Komogortsev. 2012. Magic-sense: Dynamic cursor sensitivity-based magic pointing. In *Proceedings of the CHI '12 Extended Abstracts on Human Factors in Computing Systems (CHI EA '12)*. ACM, New York, NY, 2489–2494. DOI: https://doi.org/10.1145/2212776.2223824

[38] Ribel Fares, Shaomin Fang, and Oleg Komogortsev. 2013. Can we beat the mouse with MAGIC? In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '13)*. ACM, New York, NY, 1387–1390. DOI: https://doi.org/10.1145/2470654.2466183

[39] Florian Fischer, Arthur Fleig, Markus Klar, and Jörg Müller. 2022. Optimal feedback control for modeling human–computer interaction. *ACM Transactions on Computer-Human Interaction* 29, 6, Article 51 (Dec. 2022), 70 pages. DOI: https://doi.org/10.1145/3524122

[40] Andrew Forsberg, Kenneth Herndon, and Robert Zeleznik. 1996. Aperture based selection for immersive virtual environments. In *Proceedings of the 9th Annual ACM Symposium on User Interface Software and Technology (UIST '96)*. ACM, New York, NY, 95–96. DOI: http://dx.doi.org/10.1145/237091.237105

[41] C. C. A. M. Gielen, K. van den Oosten, and F. Pull ter Gunne. 1985. Relation between EMG activation patterns and kinematic properties of aimed arm movements. *Journal of Motor Behavior* 17, 4 (1985), 421–442. DOI: https://doi.org/10.1080/00222895.1985.10735359

[42] Eric J. Gonzalez and Sean Follmer. 2023. Sensorimotor simulation of redirected reaching using stochastic optimal feedback control. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems (CHI '23)*. ACM, New York, NY, Article 776, 1–17. DOI: https://doi.org/10.1145/3544548.3580767

[43] Tovi Grossman and Ravin Balakrishnan. 2004. Pointing at trivariate targets in 3D environments. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '04)*. ACM, New York, NY, 447–454. DOI: https://doi.org/10.1145/985692.985749

[44] Tovi Grossman and Ravin Balakrishnan. 2005. The bubble cursor: Enhancing target acquisition by dynamic resizing of the cursor's activation area. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '05)*. ACM, New York, NY, 281–290. DOI: https://doi.org/10.1145/1054972.1055012

[45] Tovi Grossman and Ravin Balakrishnan. 2006. The design and evaluation of selection techniques for 3D volumetric displays. In *Proceedings of the 19th Annual ACM Symposium on User Interface Software and Technology (UIST '06)*. ACM, New York, NY, 3–12. DOI: https://doi.org/10.1145/1166253.1166257

[46] Jin Huang, Feng Tian, Nianlong Li, and Xiangmin Fan. 2019. Modeling the uncertainty in 2D moving target selection. In *Proceedings of the 32nd Annual ACM Symposium on User Interface Software and Technology (UIST '19)*. ACM, New York, NY, 1031–1043. DOI: https://doi.org/10.1145/3332165.3347880

[47] Faustina Hwang, Simeon Keates, Patrick Langdon, and P. John Clarkson. 2003. Multiple haptic targets for motion-impaired computer users. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '03)*. ACM, New York, NY, 41–48. DOI: https://doi.org/10.1145/642611.642620

[48] Werner F. Helsen, Janet L. Starkes, and Martinus J. Buekers. 1997. Effects of target eccentricity on temporal costs of point of gaze and the hand in aiming. *Motor Control* 1, 2 (1997), 161–177. DOI: https://doi.org/10.1123/mcj.1.2.161

[49] Werner F. Helsen, Digby Elliott, Janet L. Starkes, and Kathryn L. Ricker. 1998. Temporal and spatial coupling of point of gaze and hand movement in aiming. *Journal of Motor Behavior* 30, 3 (1998), 249–259. DOI: https://doi.org/10.1080/00222899809601340

[50] Werner F. Helsen, Janet L. Starkes, Digby Elliott, and Kathryn L. Ricker. 1998. Sampling frequency and the study of eye-hand coordination in aiming. *Behavior Research Methods, Instruments, & Computers* 30, 4 (1998), 617–623. DOI: https://doi.org/10.3758/BF03209479

[51] Rorik Henrikson, Daniel Clarke, Thomas White, Frances Lai, Michael Glueck, Stephanie Santosa, Daniel Wigdor, Tovi Grossman, Sean Trowbridge, and Hrvoje Benko. 2020. Head-coupled kinematic template matching for target selection in Hangry Piggos. In *Proceedings of the Extended Abstracts of the 2020 CHI Conference on Human Factors in Computing Systems (CHI EA '20)*. ACM, New York, NY, 1–4. DOI: https://doi.org/10.1145/3334480.3383176

[52] Rorik Henrikson, Tovi Grossman, Sean Trowbridge, Daniel Wigdor, and Hrvoje Benko. 2020. Head-coupled kinematic template matching: A prediction model for ray pointing in VR. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems (CHI '20)*. ACM, New York, NY, 1–14. DOI: https://doi.org/10.1145/3313831.3376489

[53] Ken Hinckley, Randy Pausch, John C. Goble, and Neal F. Kassell. 1994. A survey of design issues in spatial input. In *Proceedings of the 7th Annual ACM Symposium on User Interface Software and Technology (UIST '94)*. ACM, New York, NY, 213–222. DOI: http://dx.doi.org/10.1145/192426.192501

[54] Teresa Hirzle, Jan Gugenheimer, Florian Geiselhart, Andreas Bulling, and Enrico Rukzio. 2019. A design space for gaze interaction on head-mounted displays. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems (CHI '19)*. ACM, New York, NY, Paper 625, 12 pages. DOI: https://doi.org/10.1145/3290605.3300855

[55] Eric Horvtiz. 2006. Machine learning, reasoning, and intelligence in daily life: Directions and challenges. Microsoft Technical Report. MSR-TR-2006-185. Retrieved from https://www.microsoft.com/en-us/research/publication/machine-learning-reasoning-and-intelligence-in-daily-life-directions-and-challenges/

[56] Jeff Huang, Ryen White, and Georg Buscher. 2012. User see, user point: Gaze and cursor alignment in web search. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '12)*. ACM, New York, NY, 1341–1350. DOI: https://doi.org/10.1145/2207676.2208591

[57] Shahram Jalaliniya, Diako Mardanbegi, and Thomas Pederson. 2015. MAGIC pointing for eyewear computers. In *Proceedings of the 2015 ACM International Symposium on Wearable Computers (ISWC '15)*. ACM, New York, NY, 155–158. DOI: https://doi.org/10.1145/2802083.2802094

[58] Ricardo Jota, Miguel A. Nacenta, Joaquim A. Jorge, Sheelagh Carpendale, and Saul Greenberg. 2010. A comparison of ray pointing techniques for very large displays. In *Proceedings of Graphics Interface 2010 (GI '10)*. Canadian Information Processing Society, Toronto, ON, 269–276. Retrieved from https://dl.acm.org/doi/abs/10.5555/1839214.1839261

[59] Ricardo Jota, Albert Ng, Paul Dietz, and Daniel Wigdor. 2013. How fast is fast enough? A study of the effects of latency in direct-touch pointing tasks. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '13)*. ACM, New York, NY, 2291–2300. DOI: https://doi.org/10.1145/2470654.2481317

[60] Hilde Keuning-van Oirschot and Adrian J. M. Houtsma. 2001. Cursor displacement and velocity profiles for targets in various locations. In *Proceedings of Eurohaptics*, 108–112. Retrieved from https://portal.issn.org/resource/ISSN/1463-9394

[61] Yu-Jung Ko, Hang Zhao, Yoonsang Kim, I. V. Ramakrishnan, Shumin Zhai, and Xiaojun Bi. 2020. Modeling two dimensional touch pointing. In *Proceedings of the 33rd Annual ACM Symposium on User Interface Software and Technology (UIST '20)*. ACM, New York, NY, 858–868. DOI: https://doi.org/10.1145/3379337.3415871

[62] Regis Kopper, Doug A. Bowman, Mara G. Silva, and Ryan P. McMahan. 2010. A human motor behavior model for distal pointing tasks. *International Journal of Human-Computer Studies* 68, 10 (2010), 603–615. DOI: https://doi.org/10.1016/j.ijhcs.2010.05.001

[63] Regis Kopper, Felipe Bacim, and Doug A. Bowman. 2011. Rapid and accurate 3D selection by progressive refinement. In *Proceedings of the 2011 IEEE Symposium on 3D User Interfaces (3DUI)*. IEEE, 67–74. DOI: https://doi.org/10.1109/3DUI.2011.5759219

[64] Mikko Kytö, Barrett Ens, Thammathip Piumsomboon, Gun A. Lee, and Mark Billinghurst. 2018. Pinpointing: Precise head- and eye-based target selection for augmented reality. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems (CHI '18)*. ACM, New York, NY, Paper 81, 14 pages. DOI: https://doi.org/10.1145/3173574.3173655

[65] Michael Land, Neil Mennie, and Jennifer Rusted. 1999. The roles of vision and eye movements in the control of activities of daily living. *Perception* 28, 11 (1999), 1311–1328. DOI: https://doi.org/10.1068/p2935

[66] Edward Lank, Yi-Chun Nikko Cheng, and Jaime Ruiz. 2007. Endpoint prediction using motion kinematics. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '07)*. ACM, New York, NY, 637–646. DOI: https://doi.org/10.1145/1240624.1240724

[67] Yaxiong Lei, Shijing He, Mohamed Khamis, and Juan Ye. 2023. An end-to-end review of gaze estimation and its interactive applications on handheld mobile devices. *ACM Computing Surveys* 56, 2, Article 34 (Feb. 2024), 38 pages. DOI: https://doi.org/10.1145/3606947

[68] Zhe Li, Pietro Mazzoni, Sen Song, and Ning Qian. 2018. A single, continuously applied control policy for modeling reaching movements with and without perturbation. *Neural Computation* 30, 2 (Feb. 2018), 397–427. DOI: https://doi.org/10.1162/neco_a_01040

[69] Jiandong Liang and Mark Green. 1994. JDCAD: A highly interactive 3D modeling system. *Computers and Graphics*. 18, 4 (1994), 499–506. DOI: https://doi.org/10.1016/0097-8493(94)90062-0

[70] Daniel J. Liebling and Susan T. Dumais. 2014. Gaze and mouse coordination in everyday work. In *Proceedings of the 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing: Adjunct Publication (UbiComp '14 Adjunct)*. ACM, New York, NY, 1141–1150. DOI: https://doi.org/10.1145/2638728.2641692

[71] Y. Lu, C. Yu, and Y. Shi. 2020. Investigating bubble mechanism for ray-casting to improve 3D target acquisition in virtual reality. In *Proceedings of the 2020 IEEE Conference on Virtual Reality and 3D User Interfaces (VR)*, 35–43. DOI: 10.1109/VR46266.2020.00021

[72] Diako Mardanbegi, Tobias Langlotz, and Hans Gellersen. 2019. Resolving target ambiguity in 3D gaze interaction through VOR depth estimation. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems (CHI '19)*. ACM, New York, NY, Paper 612, 12 pages. DOI: https://doi.org/10.1145/3290605.3300842

[73] Michael McGuffin and Ravin Balakrishnan. 2002. Acquisition of expanding targets. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '02)*. ACM, New York, NY, 57–64. DOI: https://doi.org/10.1145/503376.503388

[74] Michael J. McGuffin and Ravin Balakrishnan. 2005. Fitts' law and expanding targets: Experimental studies and designs for user interfaces. *ACM Transactions on Computer-Human Interaction* 12, 4 (Dec. 2005), 388–422. DOI: https://doi.org/10.1145/1121112.1121115

[75] David E. Meyer, Richard A. Abrams, Sylvan Kornblum, Charles E. Wright, and J. E. Keith Smith. 1988. Optimality in human motor performance: Ideal control of rapid aimed movements. *Psychological Review* 95, 3 (1988), 340. DOI: https://doi.org/10.1037/0033-295X.95.3.340

[76] Mark R. Mine. 1995. Virtual environment interaction techniques. UNC Technical Report. TR95-018. Retrieved from https://techreports.cs.unc.edu/papers/95-018.pdf

[77] Martez E. Mott and Jacob O. Wobbrock. 2014. Beating the bubble: Using kinematic triggering in the bubble lens for acquiring small, dense targets. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '14)*. ACM, New York, NY, 733–742. DOI: https://doi.org/10.1145/2556288.2557410

[78] Atsuo Murata. 1998. Improvement of pointing time by predicting targets in pointing with a PC mouse. *International Journal of Human-Computer Interaction* 10, 1 (1998), 23–32. DOI: https://doi.org/10.1207/s15327590ijhc1001_2

[79] Brad A. Myers, Rishi Bhatnagar, Jeffrey Nichols, Choon Hong Peck, Dave Kong, Robert Miller, and A. Chris Long. 2002. Interacting at a distance: Measuring the performance of laser pointers and other devices. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '02)*. ACM, New York, NY, 33–40. DOI: https://doi.org/10.1145/503376.503383

[80] Mathieu Nancel, Emmanuel Pietriga, Olivier Chapuis, and Michel Beaudouin-Lafon. 2015. Mid-air pointing on ultra-walls. *ACM Transactions on Computer-Human Interaction* 22, 5, Article 21 (Aug. 2015), 62 pages. DOI: https://doi.org/10.1145/2766448

[81] Mathieu Nancel, Daniel Vogel, Bruno De Araujo, Ricardo Jota, and Géry Casiez. 2016. Next-point prediction metrics for perceived spatial errors. In *Proceedings of the 29th Annual Symposium on User Interface Software and Technology (UIST '16)*. ACM, New York, NY, 271–285. DOI: https://doi.org/10.1145/2984511.2984590

[82] Mathieu Nancel, Stanislav Aranovskiy, Rosane Ushirobira, Denis Efimov, Sebastien Poulmane, Nicolas Roussel, and Géry Casiez. 2018. Next-point prediction for direct touch using finite-time derivative estimation. In *Proceedings of the 31st Annual ACM Symposium on User Interface Software and Technology (UIST '18)*. ACM, New York, NY, 793–807. DOI: https://doi.org/10.1145/3242587.3242646

[83] Albert Ng, Julian Lepinski, Daniel Wigdor, Steven Sanders, and Paul Dietz. 2012. Designing for low-latency direct-touch input. In *Proceedings of the 25th Annual ACM Symposium on User Interface Software and Technology (UIST '12)*. ACM, New York, NY, 453–464. DOI: https://doi.org/10.1145/2380116.2380174

[84] Tomi Nukarinen, Jari Kangas, Jussi Rantala, Olli Koskinen, and Roope Raisamo. 2018. Evaluating ray casting and two gaze-based pointing techniques for object selection in virtual reality. In *Proceedings of the 24th ACM Symposium on Virtual Reality Software and Technology (VRST '18)*. Stephen N. Spencer (Ed.), ACM, New York, NY, Article 86, 2 pages. DOI: https://doi.org/10.1145/3281505.3283382

[85] Keuning-Van Oirschot H. and Houtsma A. J. M. 2001. Cursor displacement and velocity profiles for targets in various locations. In *Proceedings of EuroHaptics 2001*. EuroHaptics Society, 108–112.

[86] Alex Olwal and Steven Feiner. 2003. The flexible pointer: An interaction technique for selection in augmented and virtual reality. In *Proceedings of the ACM Symposium on User Interface Software and Technology (UIST '03)*, Vol. 3, 81–82. Retrieved from https://uist.acm.org/archive/adjunct/2003/pdf/posters/p17-olwal.pdf

[87] Phillip T. Pasqual and Jacob O. Wobbrock. 2014. Mouse pointing endpoint prediction using kinematic template matching. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '14)*. ACM, New York, NY, 743–752. DOI: https://doi.org/10.1145/2556288.2557406

[88] Ken Pfeuffer, Jason Alexander, Ming Ki Chong, and Hans Gellersen. 2014. Gaze-touch: Combining gaze with multi-touch for interaction on the same surface. In *Proceedings of the 27th Annual ACM Symposium on User Interface Software and Technology (UIST '14)*. ACM, New York, NY, 509–518. DOI: https://doi.org/10.1145/2642918.2647397

[89] Ken Pfeuffer and Hans Gellersen. 2016. Gaze and touch interaction on tablets. In *Proceedings of the 29th Annual Symposium on User Interface Software and Technology (UIST '16)*. ACM, New York, NY, 301–311. DOI: https://doi.org/10.1145/2984511.2984514

[90] Ken Pfeuffer, Benedikt Mayer, Diako Mardanbegi, and Hans Gellersen. 2017. Gaze + pinch interaction in virtual reality. In *Proceedings of the 5th Symposium on Spatial User Interaction (SUI '17)*. ACM, New York, NY, 99–108. DOI: https://doi.org/10.1145/3131277.3132180

[91] Jeffrey S. Pierce, Andrew S. Forsberg, Matthew J. Conway, Seung Hong, Robert C. Zeleznik, and Mark R. Mine. 1997. Image plane interaction techniques in 3D immersive environments. In *Proceedings of the 1997 Symposium on Interactive 3D Graphics (I3D '97)*. ACM, New York, NY, 39–43. DOI: http://dx.doi.org/10.1145/253284.253303

[92] Ivan Poupyrev, Mark Billinghurst, Suzanne Weghorst, and Tadao Ichikawa. 1996. The go-go interaction technique: Non-linear mapping for direct manipulation in VR. In *Proceedings of the 9th Annual ACM Symposium on User Interface Software and Technology (UIST '96)*. ACM, New York, NY, 79–80. DOI: http://dx.doi.org/10.1145/237091.237102

[93] Yuan Yuan Qian and Robert J. Teather. 2017. The eyes don't have it: An empirical comparison of head-based and eye-based selection in virtual reality. In *Proceedings of the 5th Symposium on Spatial User Interaction (SUI '17)*. ACM, New York, NY, 91–98. DOI: https://doi.org/10.1145/3131277.3132182

[94] Jaime Ruiz and Edward Lank. 2009. Effects of target size and distance on kinematic endpoint prediction. Technical Report CS-2009-25. University of Waterloo. Retrieved from https://cs.uwaterloo.ca/research/tr/2009/CS-2009-25.pdf

[95] Jaime Ruiz and Edward Lank. 2010. Speeding pointing in tiled widgets: Understanding the effects of target expansion and misprediction. In *Proceedings of the 15th International Conference on Intelligent User Interfaces (IUI '10)*. ACM, New York, NY, 229–238. DOI: https://doi.org/10.1145/1719970.1720002

[96] Ludwig Sidenmark and Anders Lundström. 2019. Gaze behaviour on interacted objects during hand interaction in virtual reality for eye tracking calibration. In *Proceedings of the 11th ACM Symposium on Eye Tracking Research & Applications (ETRA '19)*. ACM, New York, NY, Article 6, 9 pages. DOI: https://doi.org/10.1145/3314111.3319815

[97] Ludwig Sidenmark and Hans Gellersen. 2019. Eye & head: Synergetic eye and head movement for gaze pointing and selection. In *Proceedings of the 32nd Annual ACM Symposium on User Interface Software and Technology (UIST '19)*. ACM, New York, NY, 1161–1174. DOI: https://doi.org/10.1145/3332165.3347921

[98] Ludwig Sidenmark and Hans Gellersen. 2019. Eye, head and torso coordination during gaze shifts in virtual reality. *ACM Transactions on Computer-Human Interaction* 27, 1, Article 4 (Feb. 2020), 40 pages. DOI: https://doi.org/10.1145/3361218

[99] Ludwig Sidenmark, Mark Parent, Chi-Hao Wu, Joannes Chan, Michael Glueck, Daniel Wigdor, Tovi Grossman, and Marcello Giordano. 2022. Weighted pointer: Error-aware gaze-based interaction through fallback modalities. *IEEE Transactions on Visualization and Computer Graphics* 28, 11 (Nov. 2022), 3585–3595. DOI: https://doi.org/10.1109/tvcg.2022.3203096

[100] Vincent Sitzmann, Ana Serrano, Amy Pavel, Maneesh Agrawala, Diego Gutierrez, Belen Masia, and Gordon Wetzstein. 2018. Saliency in VR: How do people explore virtual environments? *IEEE Transactions on Visualization and Computer Graphics* 24, 4 (2018), 1633–1642. DOI: https://doi.org/10.1109/TVCG.2018.2793599

[101] Barton A. Smith, Janet Ho, Wendy Ark, and Shumin Zhai. 2000. Hand eye coordination patterns in target selection. In *Proceedings of the 2000 Symposium on Eye Tracking Research & Applications (ETRA '00)*. ACM, New York, NY, 117–122. DOI: https://doi.org/10.1145/355017.355041

[102] Frank Steinicke, Timo Ropinski, and Klaus Hinrichs. 2006. Object selection in virtual environments using an improved virtual pointer metaphor. In *Computer Vision and Graphics. Computational Imaging and Vision*, Vol. 32. Wojciechowski K., Smolka B., Palus H., Kozera R., Skarbek W., and Noakes L. (Eds.), Springer, Dordrecht, 320–326. DOI: https://doi.org/10.1007/1-4020-4179-9_46

[103] Sophie Stellmach and Raimund Dachselt. 2012. Look & touch: Gaze-supported target acquisition. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '12)*. ACM, New York, NY, 2981–2990. DOI: https://doi.org/10.1145/2207676.2208709

[104] Vildan Tanriverdi and Robert J. K. Jacob. 2000. Interacting with eye movements in virtual environments. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '00)*. ACM, New York, NY, 265–272. DOI: http://dx.doi.org/10.1145/332040.332443

[105] Emanuel Todorov and Michael I. Jordan. 2002. Optimal feedback control as a theory of motor coordination. *Nature Neuroscience* 5, 11 (Nov. 2002), 1226-1235. DOI: http://dx.doi.org/10.1038/nn963

[106] Lode Vanacken, Tovi Grossman, and Karin Coninx. 2007. Exploring the effects of environment density and target visibility on object selection in 3D virtual environments. In *Proceedings of the IEEE 3D User Interfaces*. IEEE, 117–124. DOI: https://doi.org/10.1109/3DUI.2007.340783

[107] Eduardo Velloso, Jayson Turner, Jason Alexander, Andreas Bulling, and Hans Gellersen. 2015. An empirical investigation of gaze selection in mid-air gestural 3D manipulation. In *Human-Computer Interaction*. Julio Abascal, Simone Barbarosa, Mirko Fetter, Tom Gross, Philippe Palanque, and Marco Winckler (Eds.), Springer, 315–330. DOI: https://doi.org/10.1007/978-3-319-22668-2_25

[108] Yushi Wei, Rongkai Shi, Difeng Yu, Yihong Wang, Yue Li, Lingyun Yu, and Hai-Ning Liang. 2023. Predicting gaze-based target selection in augmented reality headsets based on eye and head endpoint distributions. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems (CHI '23)*. ACM, New York, NY, Article 283, 1–14. DOI: https://doi.org/10.1145/3544548.3581042

[109] Cahdwick A. Wingrave, Doug A. Bowman, and Naren Ramakrishnan. 2002. Towards preferences in virtual environment interfaces. In *Proceedings of the Workshop on Virtual Environments (EGVE '02)*, Vol. 2, 63–72. DOI: http://dx.doi.org/10.2312/EGVE/EGVE02/063-072

[110] Haijun Xia, Ricardo Jota, Benjamin McCanny, Zhe Yu, Clifton Forlines, Karan Singh, and Daniel Wigdor. 2014. Zero-latency tapping: Using hover information to predict touch locations and eliminate touchdown latency. In *Proceedings of the 27th Annual ACM Symposium on User Interface Software and Technology (UIST '14)*. ACM, New York, NY, 205–214. DOI: https://doi.org/10.1145/2642918.2647348

[111] Difeng Yu, Hai-Ning Liang, Xueshi Lu, Kaixuan Fan, and Barrett Ens. 2019. Modeling endpoint distribution of pointing selection tasks in virtual reality environments. *ACM Transactions on Graphics* 38, 6, Article 218 (Dec. 2019), 13 pages. DOI: https://doi.org/10.1145/3355089.3356544

[112] Shumin Zhai, Carlos Morimoto, and Steven Ihde. 1999. Manual and gaze input cascaded (MAGIC) pointing. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '99)*. ACM, New York, NY, 246–253. DOI: http://dx.doi.org/10.1145/302979.303053

[113] Fengyuan Zhu, Ludwig Sidenmark, Mauricio Sousa, and Tovi Grossman. 2023. PinchLens: Applying spatial magnification and adaptive control display gain for precise selection in virtual reality. In *Proceedings of the IEEE ISMAR International Symposium on Mixed and Augmented Reality (ISMAR)*, 1221–1230. Retrieved from https://doi.org/10.1109/ISMAR59233.2023.00139

[114] Brian Ziebart, Anind Dey, and J. Andrew Bagnell. 2012. Probabilistic pointing target prediction via inverse optimal control. In *Proceedings of the 2012 ACM International Conference on Intelligent User Interfaces (IUI '12)*. ACM, New York, NY, 1–10. DOI: https://doi.org/10.1145/2166966.2166968

Received 7 December 2023; revised 19 July 2024; accepted 22 July 2024