IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 15 November 2023, accepted 28 December 2023, date of publication 8 January 2024,
date of current version 16 January 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3350788

RESEARCH ARTICLE

Continuous Prediction of Pointing Targets With Motion and Eye-Tracking in Virtual Reality

CHOONGHO CHUNG, (Student Member, IEEE), AND SUNG-HEE LEE, (Member, IEEE)
Graduate School of Culture Technology, KAIST, Daejeon 34141, South Korea

Corresponding author: Sung-Hee Lee (sunghee.lee@kaist.ac.kr)

This work was supported by NRF, South Korea, under Grant 2022R1A4A5033689.

This work involved human subjects or animals in its research. The authors confirm that all human/animal subject research procedures and protocols are exempt from review board approval.

ABSTRACT We present a study on continuously predicting the direction to a pointing target in virtual environments using motion and eye-tracker data throughout the pointing process. We first collect time series data for user motion and eye-tracker in a cursorless, single-target pointing task. Results from analyzing fixation points from different sensors and observing velocity profiles over the course of pointing provide insights into optimally configuring features for predicting the target angles. Following this analysis, we train a recurrent neural network that feeds on sliding window inputs for continuously operating target direction prediction from start to finish. The input window contains historical data from past to current frames, capturing temporal changes in the feature data. By feeding on this input, our model can predict the direction of the target at any given time during pointing. Our findings demonstrate that incorporating eye-tracker data into the prediction model boosts the maximum achievable accuracy by 2.5 times when compared to baselines without eye-tracker data inputs. The results suggest that using features from both the eye-tracker and joint motion contributes to higher prediction performance, as well as faster stabilization of output values at the starting phase of pointing.

INDEX TERMS Gaze, pointing, prediction, eye-tracker, virtual reality.

I. INTRODUCTION

Pointing gesture is an intuitive, non-verbal mode of communication that selects a position or a virtual object in Collaborative Virtual Environments (CVE). The current Virtual Reality (VR) devices provide various off-the-shelf pointing interfaces, mostly using controller-based ray-casting methods.

Head-Mounted Displays (HMDs), such as the Vive Pro Eye and Hololens 2, are equipped with pose tracking capabilities for heads, hands, and even pupil movements, allowing tracking of the user's gaze. The integration of both head, hand and eye-tracking features in commercial HMDs offers an affordable option for leveraging natural user motion in real-time pointing activities, which is less artificial and intrusive than ray-casting methods.

Recent studies to infer pointing targets from user's natural pointing gesture mostly assume the state of fixation, in which the user is stably pointing at a target [1]. While inferring the target at the fixated state has many applications, our work tackles a higher challenge of predicting a target even at earlier phases of pointing. Previous studies addressed continuous prediction of user intention and behavior by observing user's motion to predict collisions with virtual objects [2], [3] or pointing targets [4]. Developing more accurate and earlier predictions for pointing targets can improve the responsiveness and reduce time delay in interactive applications, such as teleconference, game, and design.

User behavior before pointing fixation can be affected by individual differences in motion and body dimensions [5]. This can pose challenges in formulating an adaptive structure for a pointing prediction model. We address this problem by leveraging a deep neural network structure to continuously predict a pointing target throughout the course of the pointing

The associate editor coordinating the review of this manuscript and approving it for publication was Lei Wei.

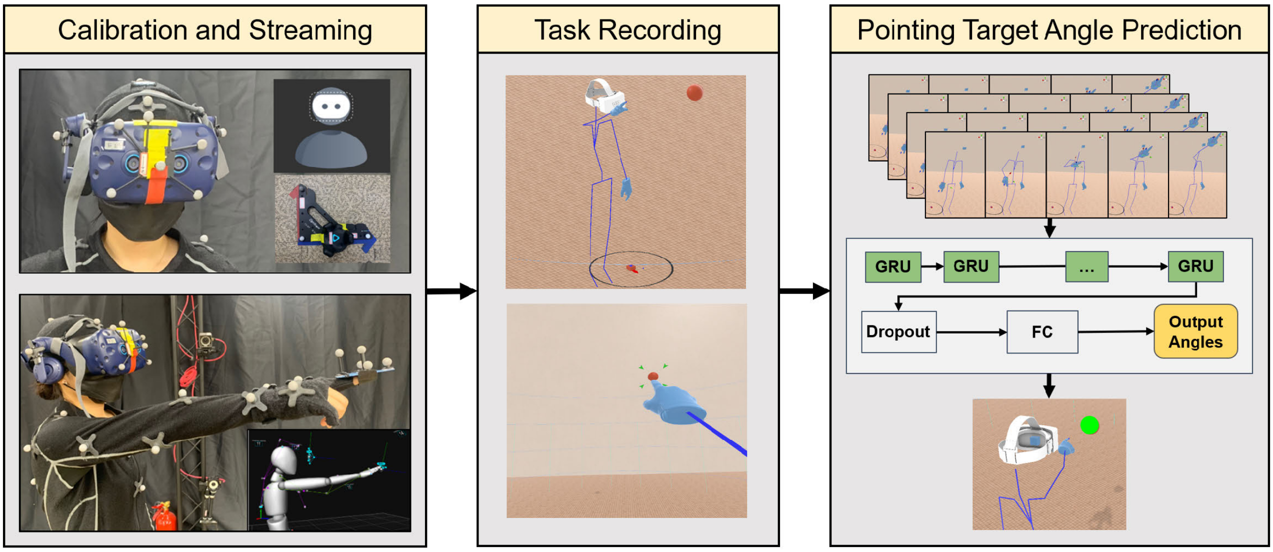


FIGURE 1. We collect eye-tracker and body motion data of a user in a mocap suit and a VR headset inside a virtual scene. The user conducts multiple single-target pointing tasks in the virtual environment, and the streamed sensor data is recorded. The recorded data is processed and configured into a sliding window format to be fed into a prediction model. The model outputs predicted target azimuth and elevation angles with respect to the user HMD's orientation throughout the duration of recorded pointing session.

action. To develop a framework that can provide accurate predictions for various pointing states and individuals, we first focus on identifying useful input features for pointing target inference.

Previous studies on inferring human pointing targets [6], [7], [8] use tracked position and orientation of body parts, such as the head, torso, and hands. Among the most intuitive of such features are those rooted in the principles of ray-casting, where a virtual ray is defined by an object orientation or between two reference points, originating from arms, hands, or eyes [9], [10], [11], [12]. Ray-casting vector, while not as accurate as eye-trackers, provide useful information for modeling human pointing at fixation pose, and for classification and inference of pointing targets [7], [8], [13]. However, its use is not widely explored for early-prediction of object interactions, as is the case for kinematic joint information.

Kinematic motion data, ray-casting vectors, eye-tracker measurements all qualify as great candidate features for inferring or predicting pointing targets. To gain insights into how a continuously operating model should function for this application and to come up with appropriate input features for our prediction model, we analyze the characteristics of these features not only at times of fixation but also during the course of pointing.

After analyzing the input features, we train a deep neural recurrent network structure for receiving input values of both motion and eye-tracker based features during progression of pointing. For the continuous operation of our model, we feed a sliding window consisting of the history of past and current feature data, allowing our model to predict target angular directions (Fig. 1). We conduct analysis to decide on best input feature combination, network type, and window length for achieving optimal performance. Our final recurrent neural network model is configured to provide the maximum accuracy and early prediction capabilities for pointing target directions.

The contributions of our work can be summarized as follows.

- We introduce a recurrent pointing angle prediction model that utilizes features from data of the head, eyes, dominant hand fingertip, and ray-casting vectors. An ablation study on input features and sliding window length is conducted to identify the optimal settings for our model. Through our approach, we achieve high accuracy in early prediction of pointing direction.
- In addition to our prediction framework, we present a multimodal dataset consisting of eye-tracker and full-motion capture data. We explain the appropriate steps for merging and processing data from two different tracking systems, resulting in thousands of high quality data sequences for future use in pointing prediction research.
- Our collected data is analyzed through fixation point and velocity profile analysis. The fixation point analysis involves examining the user's final fixation state to determine the accuracy of crucial raw data types. In the profile analysis, the progression velocity of sensor data is observed for the entire sequence to evaluate how each data type can differentiate between various target positions.

In the following background section, we discuss contributions of contemporary works on pointing behaviors. This is followed by explanations on user data collection procedures in section III. In section IV, we explain how the collected data is analyzed to identify defining features for each types of data.

We then proceed to designing input features and outputs for a pointing angle prediction model, based on the preliminary data analysis. Evaluation of our model and its implications are discussed in section VI and VII.

II. BACKGROUND

A. MODELING HUMAN POINTING AT TARGET FIXATION

A person's final pointing pose can show variability among individuals [14] with tendencies to over-reach or under-reach for targets. When the human pointer fixates at a target, another observer, looking at the final pose, can make systematic errors in interpreting the intended target direction by the pointer [15], [16]. Sousa et al. [6] conducted a 1D pointing analysis on correcting such errors, while Mayer et al. [7] used data from final pointing poses and paired observations of such poses to model and correct systematic human interpretation errors by observers in a cylindrical target setup. In their work, ray-casting vector like Eye-Finger-Ray-Casting (EFRC) is used as an effective tool for highlighting non-linear behaviors of human pointing interactions.

A different approach aims to train models that infers pointing directions by feeding in input features from the human pointer. Information on handedness and ocular dominance was found to influence performance of pointing behavior [14], and incorporating these elements in training a prediction model leads to a higher accuracy in predicting target position [13]. The modeling approach for human pointing establishes the use of various motion and encoded features as effective inputs for inferring target directions [8].

B. TARGET INFERENCE WITH EYE GAZE

Eye gaze serves as a fast and accurate modality for visual attention [17], [18], [19]. Saccade, a quick eye movement towards a target, can reveal crucial information on timing of perceptual decision making, by studying its velocity and a following gaze fixation to a visual target [20].

When eyes are used in selection tasks, eye-gaze aligns with Fitts' Law in task performance analysis [21]. In many cases, eye movements are linked with head movements, and their coordination is a frequently explored behavior for accurately detecting 2D gaze [22], [23], [24], [25].

Recent study on eye and head gaze models distribution of fixation points in augmented reality (AR) environments to demonstrate high accuracy in target selection tasks [1]. The same study also finds the use of non-intrusive confirmation methods for eye movements, such as air-tapping, to be beneficial in enhancing selection accuracy.

C. PREDICTION OF TARGETS DURING POINTING

Pointing motion involves synchronized movements of multiple joints, including the torso, arms, and head. The kinematic relationships among these joints at pointing fixation or during pointing offer valuable information for predicting the endpoint trajectories of the head and hands [2], [24].

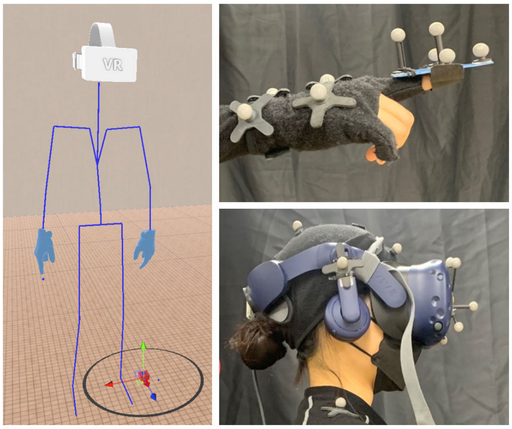


FIGURE 2. Motion capture system and eye-tracker are calibrated to share a common origin and visualized in data collection application as a user skeleton wearing an HMD (left). Our data includes positional tracking of dominant index fingertip (upper right), as well as eye-tracker integrated VR headset (lower right).

Henrikson et al. [4] use a Kinematic Template Matching (KTM) technique for continuous target prediction by using data from head and controller movements. This technique involves matching the end-effector trajectory with templates stored in libraries of previously collected data to predict the cursor point. Additionally, KTM can be applied during pointing progression to monitor prediction errors in the early stages of pointing, demonstrating how time-series data can be used in continuous prediction of pointing target.

In summary, most recent works on pointing focus on natural arm-extending motion, minimizing the use of UI cursor elements for selecting distant objects. Error modeling and correction approaches are successful at discovering universal patterns in human pointing and input features for target prediction. Works on dynamic joint movements demonstrate relevant motion characteristics that show correlation with pointing targets. Applications with eye-tracking offer the most accurate target prediction potentials but do not consider the kinematics of extended arm movements or upper bodies and focus on gaze at the time of target fixations.

To enhance target inference performance both upon pointing completion and during a pointing task, it is advantageous to utilize multiple forms of sensor data such as gaze, motion, and ray-casting features throughout the entire pointing progression. Integrating this data into the input features of a model structure may allow for the continuous prediction of target directions, particularly for early prediction of pointing targets. In this study, we collect kinematic and eye-tracking data, processing them as input features for a target prediction model. This approach enables us to examine the temporal characteristics of various sensor data and input features across the entire duration of the pointing sequence.

III. POINTING BEHAVIOR OF USERS IN VR

We gather time-series data from human subjects performing single-target pointing tasks. The collected data will enable us

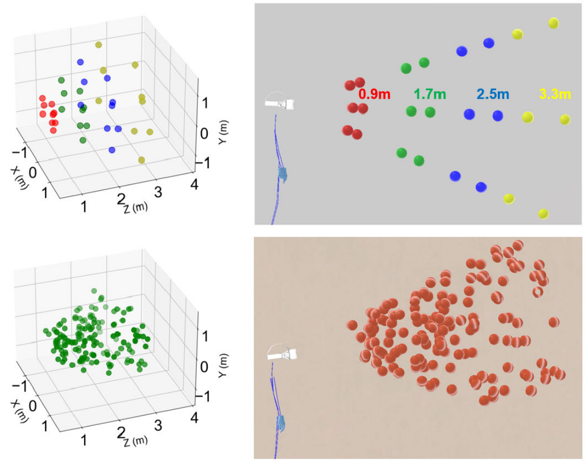


FIGURE 3. In a spherical setup, targets are placed for discrete values of angles and distances. For a random setup, target placements are determined by a uniform probability density function on angle and distance values, which is reset for different subjects.

to explore the accuracy of previously employed data features throughout the duration of a pointing task, as well as the relative importance among possible combinations of input features for our model.

A. PARTICIPANTS AND ENVIRONMENT SET-UP

We recruit participants (gender: male=13, female=11) from a local university (age: mean=26.4, standard deviation=2.5) for data collection. All participants had normal or corrected eyesight, and did not report motion sickness or eyesight impairment during the recording sessions. Twelve individuals reported previous experience in using VR devices. All subjects were identified as right-handed.

The participants were divided into two groups to record data in two distinct target placement scenarios: 1) a spherical grid target setup, and 2) a randomized target placement setup. The first group (age mean=27.0, standard deviation=2.1) comprised 16 subjects (gender: male=11, female=5), while the second group (age mean=25.9, standard deviation=2.9) consisted of 8 subjects (gender: male=2, female=6).

We set up a VR environment in Unity3D Game Engine for data collection (Fig. 1). We used 23 OptiTrack motion capture cameras (Prime13 and Prime13W) to collect motion data and the HTC Vive Pro Eye for gathering eye-tracking data. Two independent threads, each connected to a motion capture software (Motive 2.3.0) and an eye-tracking software (SRanipal SDK v1.3.1.0), streamed and recorded data at 120 Frames-Per-Second (FPS).

Our motion capture system tracks a full body pose and two separately defined rigidbody objects: one for tracking the dominant hand index fingertip and the other for tracking the VR headset (Fig. 2).

Our motion capture system and eye-tracker have different world origins, and we conducted a calibration for them to share a common point of reference. We then replaced the native Vive headset tracking system with the tracking data of the HMD rigidbody for higher accuracy and more consistent frametime.

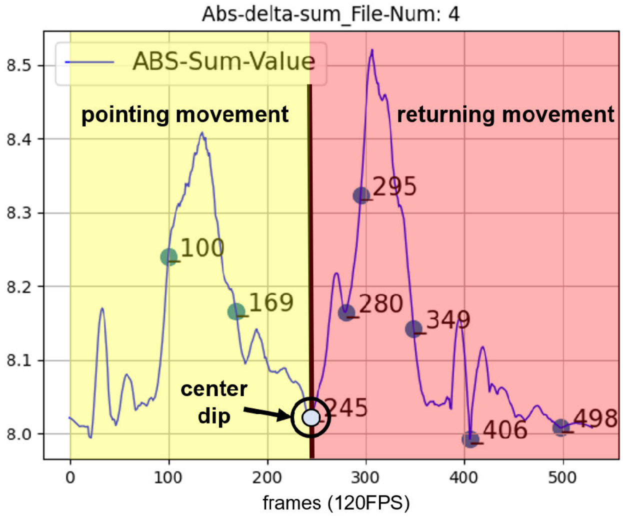


FIGURE 4. We plot absolute summation values of selected delta features over a pointing task sequence to find local minima that corresponds to pointing fixation. A minima closest to a center dip between the pointing movement and the returning movement is chosen as time of pointing fixation.

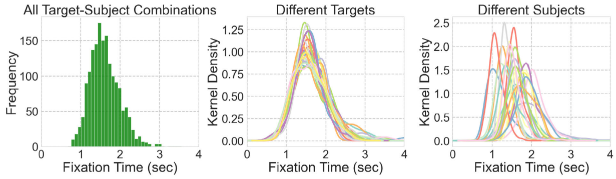


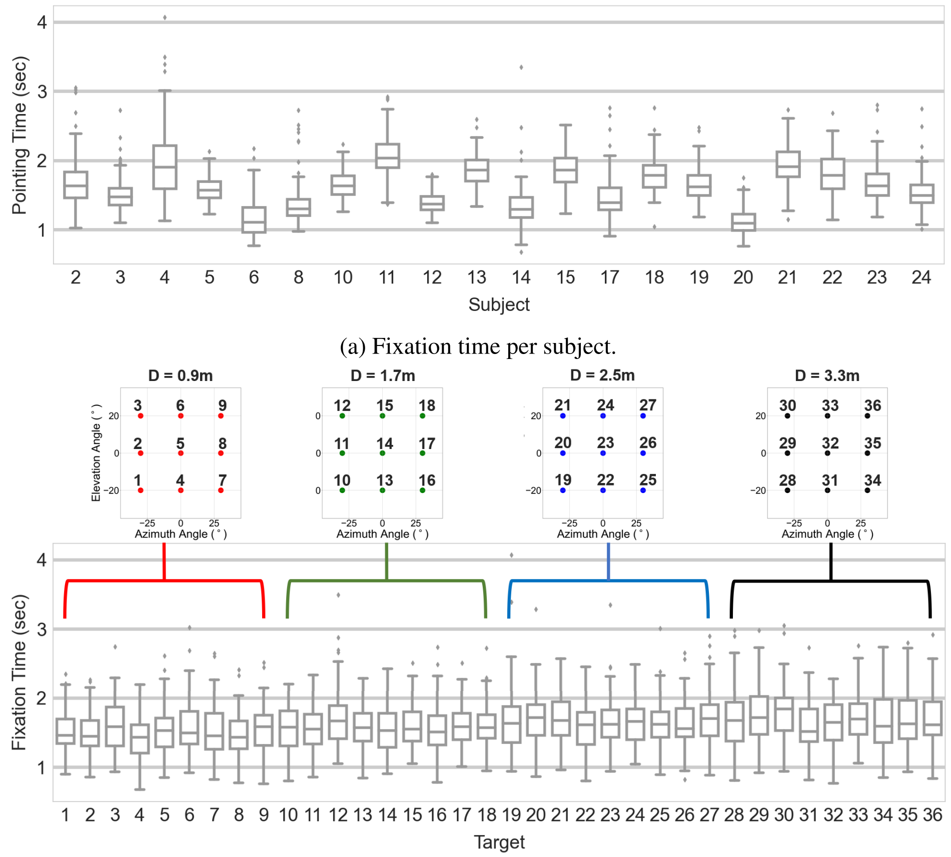
FIGURE 5. Visualization for pointing fixation time consists of (a) histogram for all subject-target combinations (left), distributions per target (center), and per subject (right).

B. TARGET PLACEMENT AND SIZE

In our work, we use a spherical target setup and random spatial placement for pointing tasks (Fig. 3). In the spherical setup, a target is positioned on a spherical grid, with its candidate position spaced uniformly across multiple levels of azimuth, elevation, and distance. In the random spatial placement, target locations follow a randomly distributed uniform pattern within a specified volume, constrained by the distance and angle ranges set in the first setup. Our aim of employing the second setup is to avoid overfitting our model to specific target coordinates during training.

For the spherical grid setup, we selected target depths of 0.9m, 1.7m, 2.5m, and 3.3m from the user's HMD, allowing for extended-arm pointing with an extended index finger for various distances. Apart from placing targets at the center, we adjusted additional azimuth and elevation parameters to ±30° and ±20°, respectively. These parameters were set based on a study regarding the realistic error distribution and Field Of View (FOV) of the Vive Pro Eye [26].

Regarding the target size, we referred to guidelines from previous works [27], [28] and set our target as a sphere with a fixed diameter of 15.4cm. The sphere's radius is set to be



(a) Fixation time per subject.

(b) Fixation time per target, with respective positions.

FIGURE 6. Bar plot visualizes pointing fixation time (a) per-subject in both spherical and randomized target setups, and (b) per-target in a spherical setup.

20% larger than the claimed maximum error of the Vive Pro Eye HMD at a distance of 3.3m. Throughout the recording phase, the target was consistently colored red.

C. DATA COLLECTION PROCEDURES

Upon their arrival, each participant received a briefing regarding the scope and intended use of the collected data, submitted written consent to proceed, provided basic demographic information, and changed into a motion capture suit.

We manually placed optical markers on the suit for each subject, set the inter-pupillary distance for the HMD, and calibrated the eye-tracker. Next, we aligned the origin for both the Vive and motion capture systems and projected the user's skeleton and HMD object into the virtual environment. Additionally, we attached a virtual 3D hand model to the 3D skeleton, adjusting its scale and index finger length to fit the user. An inverse kinematic algorithm was then applied to the index finger to enable natural finger movement.

Each participant conducted 144 pointing sessions. In the spherical grid target setup, target placement covered all 36 combinations of angle and distance values. Pointing action was performed four times for each target position. In the random target setup, targets were placed randomly within a designated target volume. Order of target positions were randomized for all setups.

Before each pointing task, participants were shown an approximate location of the target to ensure they had a clear mental picture of their future pointing position. They were instructed to maintain a forward-facing prone ready-position in a standing pose, ensuring that each sequence starts from the same ready pose. An audio beep triggered participants to start pointing.

Upon initiation of the start audio cue, data recording began. Simultaneously, a single target was positioned relative to the user's HMD pose. Participants pointed at the center of the target and then freely returned to the original prone pose. The target is then removed, and the step is repeated for the remaining target positions.

Previous works on pointing pose instructed participants to mark point completion time either by manual controller inputs or encoded user motion [1], [7], [8]. This instruction does not fit our application of collecting natural pointing movements from start to finish. Instructing participants to manually label their pose fixation time can introduce

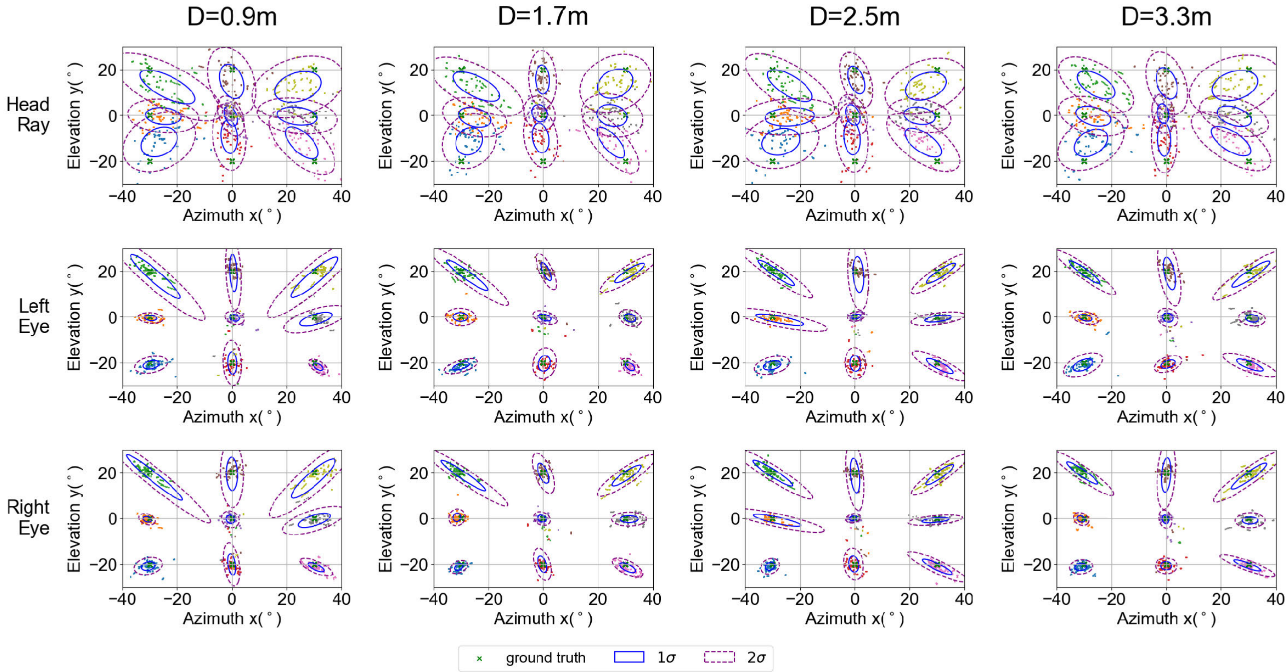


FIGURE 7. Ray projection points and their 95 percentile and 99 percentile confidence boundaries on spherical surfaces at different distances. Data is taken ±5ms around the time of least-movement for pointing fixation. Each row applies for raw data of the head ray, left eye gaze, and right eye gaze. Each column corresponds to different target distances from initial HMD positions.

distractions in progressing participant movements towards pointing fixation. Therefore, instead of instructing participants to label fixation time themselves, we manually labeled them in the post-processing step.

Regarding the pointing motion, participants were advised to point at the target's center as they would point at an object for a nearby friend. We verbally instructed the participants that they were free to point as they would do in real life. No further advice was given to ensure that the recorded data solely represented the participants' normal pointing behaviors.

Upon completion of all pointing tasks, participants took a brief interview regarding their opinions on the experiment setup and procedures. The whole procedure took around 45 minutes, including the briefing and the calibration process.

D. POST PROCESSING

In the post-processing step, we synchronized the eye-tracker data with the motion capture timestamps. The final synchronized data measured at 120 FPS.

In eye-tracker data, we observed blinking events, causing the loss of eye-tracking frames around $150-250$ miliseconds (ms). It is reported that eye blinking typically lasts around 250ms to 450ms and may induce horizontal and vertical eye movements averaging at 0.6° and 2.1°, respectively [29]. The data show that participants rarely blink before completing a pointing task, introducing minimal effect to overall eye-tracking values. We replaced the lost eye root positions and gaze vectors with interpolation from neighboring values.

For encoding rate of changes in values, we also calculated delta values between subsequent frames for the motion and eye-tracker measurements. Noise and jitters resulted in occasional spikes in calculated delta values, introducing significant deviations from the main profile. To suppress outliers, we identified outlying values beyond 1.5 interquartile ranges, and replaced them with the closest neighboring value.

We manually labeled the pointing completion time, heuristically assuming that the user shows the least movement of key features at time of pointing fixation. These features originate from the HMD, dominant hand, and its index fingertip, and vector product between various ray casting features using the head, indextip, and eyes.

The HMD transform is chosen for its stated importance in user visual attention. The transforms for the dominant hand and the index fingertip enable us to narrow down time interval that contains pointing fixation by showing rapid accelerations at the pointing phase, and the following returning motion. Finally, we consider synchronized movements of various body parts by calculating linear Vector product between Eye-Finger-Ray-Cast (EFRC) vectors, and Head-Finger-Ray-Cast (HFRC) vector. These vectors originates from each eyes or head root to intersect with the position of the fingertip. Linear vector product between these vectors can represent synchronization of alignment for pointing behaviors involving the index finger.

The magnitudes of delta elements for these features were summed up and visualized for each pointing file for each pointing sequence file (Fig. 4. This yields a double-peaked profile graph, where the each of the peaks represents rapid movement at the initial target pointing and the returning phase, respectively. We identify all local minima in the graph by using linear penalized summation method. Of all the candidate minima frames, the one closest to the center dip is chosen as the timestamp of least movement, based on reasoning that the dip represents temporary moment of stabilization, just before the user returns back to the prone pose. Out of the 3450 files annotated from 24 participants in the spherical setup, we discarded 108 files (2.9%) showing unstable tracking values.

Before analyzing the data, we drew trajectories of projection rays for different subjects, to detect for any anomalies in the data. Among the 24 participants, three subjects consistently stood out by exhibiting the following conditions: a) overly restricting head rotations, and b) displaying hesitations in deciding final pointing poses. Restrictive head movements introduces bias into our data, which does not represent natural pointing motions. Indecision in final pointing results in more than 2-3 random jerks and oscillations in projection point trajectories for more than one third of all pointing tasks for a subject.

We observed that restrictive head movements usually result from over-complying to instructions on maintaining prone pose at the start, which influenced the affected participants to fix upper body and head unnaturally. This was visualized in projection plot where the head moved less than half when compared to the other participants' projections consistently for multiple files. On the other hand, causes for sudden multiple trajectory changes during pointing was difficult to speculate.

Apart from being unnatural, restricted head can also increase eye-tracker errors due to higher measurement errors of eye trackers at higher eye rotation angles. Unpredictable trajectories can increase spread of model prediction errors. Consequently, we excluded additional 452 sequences from the affected subjects no.1, no.7, and no.16.

Finally, subject no.9 (141 sequences) was also excluded due to measurement errors in root origin calibration process. The excluded four subjects account for a removal of an additional 16.3% of the collected data.

IV. CHARACTERISTICS OF POINTING DATA

We analyze the ray projections and temporal profiles of the collected raw data to identify their key characteristics. We mainly look at: a) task completion time, b) data at the fixation time of pointing, and c) the angular velocity profiles of key sensor data for different target positions. The results obtained from this analysis are used to select effective input feature candidates for our pointing angle prediction model. The majority of the analysis is conducted using data from the spherical target setup.

A. TASK COMPLETION TIME

We examined the completion time of pointing for different target positions and various subjects (Fig. 5, Fig. 6). Averaging all four repetition sessions into one (N = 20 subjects × 36 target positions), we generated a histogram showing a positively skewed distribution (skew = 0.80). Next, we created a Kernel Density Estimation (KDE) plot displaying completion times for different targets in the spherical setup and repeated this for plotting KDE for different subjects.

TABLE 1. Fixation projection RMS errors for different distances. Highest values for each error category are highlighted in bold text.

Datatypes	Errors for different Target Distances							
	d=0.9m		d=1.7m		d=2.5m		d=3.3m	
	m (°)	std (°)	m (°)	std (°)	m (°)	std (°)	m (°)	std (°)
HMD	**6.9**	2.8	6.5	2.6	6.7	2.8	6.6	2.7
left eye	**1.7**	0.9	1.4	0.7	1.5	0.9	1.3	0.5
right eye	**1.4**	0.8	1.1	0.4	1.1	0.4	1.0	0.5
	Errors for different Vertical Angles							
	y=−20°		y=0°		y=20°			
	m (°)	std (°)	m (°)	std (°)	m (°)	std (°)		
HMD	**9.0**	1.2	3.8	2.2	7.1	1.4		
left eye	1.1	0.2	1.2	0.7	**2.2**	0.7		
right eye	0.8	0.1	1.0	0.4	**1.6**	0.6		
	Errors for different Horizontal Angles							
	x=−30°		x=0°		x=30°			
	m (°)	std (°)	m (°)	std (°)	m (°)	std (°)		
HMD	6.9	1.7	5.6	3.9	**7.5**	1.4		
left eye	**1.7**	1.1	1.3	0.6	1.5	0.5		
right eye	**1.2**	0.7	1.0	0.5	1.2	0.6		

The histogram, exhibiting a positive skew, underwent a log transformation to check for normality. The transformed data from the distributions passed the Shapiro-Wilk normality test. The per-target analysis (N = 12 subjects × 36 targets) passed Levene's Test for equal variances. In contrast, distributions from the per-subject plots (N = 12 subjects × 36 targets) were distinctly different due to variations in completion times, and did not have the same variance to pass Levene's Test.

From the KDE distributions for subjects, it is evident that the positive skew of the histogram could be attributed to differences both between and within subjects. A one-way ANOVA for the target placement distributions demonstrated a significant difference (F (35, 396) = 3.09, $p < 0.001$) in pointing times across target positions.

B. FIXATION-POINT ANALYSIS

We selected measurement values within ±5ms around the point of least movement to select data at pointing fixation. From this data, we plotted 2D projections of the HMD and eye forward vectors onto spherical surfaces with different distance radii (Fig. 7).

The RMS angular deviations for the left eye (M=1.49°, SD=0.78°) and the right eye (M=1.13°, SD=0.59°) showed lower angular errors compared to the HMD ray projection (M=7.48°, SD=3.01°), by at least 4.5 times. For all target placements, we consistently observed higher accuracy of the right eye over the left eye, influenced by higher samples of right-eye dominant subjects. When mean projection error of both eyes were calculated for each of the subjects, 17 of the

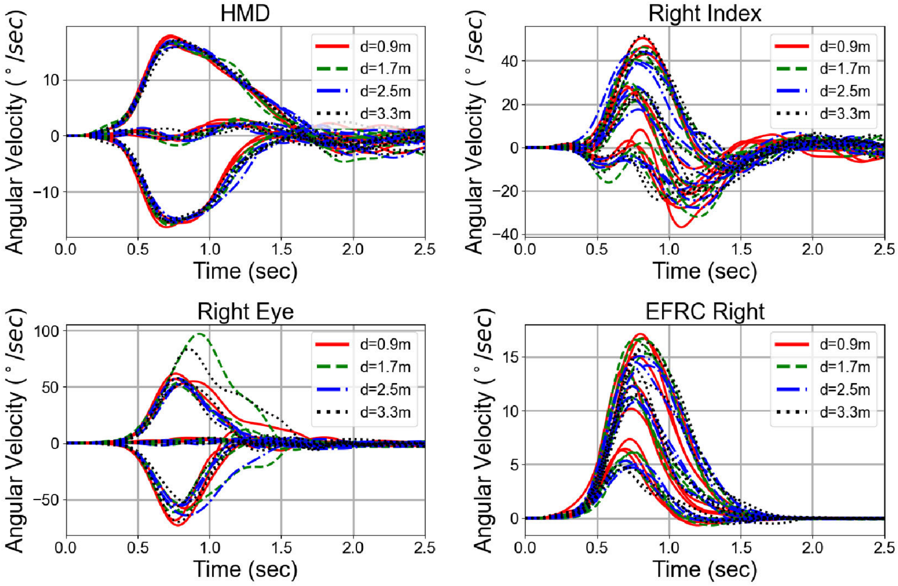


FIGURE 8. Angular velocity profile (w.r.t world origin) for targets is plotted for: (a) head, (b) right index fingertip, (c) right eye, and (d) right EFRC vector.

24 subjects showed lower mean angular errors for the right eye, indicating more right-ocular dominant participants in the study.

Projection errors did not exhibit significant variations by changes in distance (Table. 1). However, it was notable that the maximum error was consistently observed for the shortest distance for both the HMD and eyes. It is possible that relatively small distance of 0.9m affected subjects to express more diverse pointing behaviors of the body and eye fixations, when compared to larger target distances that necessitate users to use more uniform pointing poses.

For vertical target placements, unstable measurements at upper targets were reported in previous work for Vive Pro Eye testing [26]. We also report similar trend for a single target pointing task, where up to twice the error is reported for upper row target positions over the lower row for both eyes.

In the case of horizontal target placements, center placement yielded slightly higher projection accuracy with small asymmetry for left and right target placements for head and eyes.

A Kolmogorov-Smirnov test for projection angular errors for HMD and eyes revealed that they did not stem from the same distribution. HMD errors showed a mildly positive skew (skew=0.78), but distributions for both eyes were extremely skewed (left eye: skew=1.38, right eye: skew=1.35).

C. VELOCITY PROFILE ANALYSIS

In the profile analysis, we observed the world angular velocities of the HMD, dominant hand, and eyes, and the EFRC vector, a vector ray originating from eye root, passing through the fingertip position (Fig. 8). Angular velocity is recognized for its effectiveness in distinguishing targets at varying azimuth and elevation angles [4].

Initially, data for all pointing sequences across the 36 positions were averaged to generate angular velocity profiles. To reduce high-frequency noise in the raw eye data, a low-pass filter was applied to all eye data sequences and averaged. For gaze and EFRC, we calculated rotation from vectors from two consequent frames. Due to the dominance of right-eyed participants, we use right eye vector and right EFRC for profile analysis.

To calculate angular velocity, we calculated the rotation difference matrix between two subsequent matrices R_{t+1} and R_t. The matrix was then converted to angle-axis configuration to gain angular velocity Ω.

$$Q = R_{t+1}^{-1} R_t = e^{\Omega}$$

Examining yaw and pitch velocity provided additional means to differentiate symmetric angular placements of targets in horizontal or vertical setups, by introducing additional degrees of freedom along the x and y axes. Roll velocity on the other hand, generally displayed redundant information with more variance and noise for all data types.

To assess the reaction time to the start audio cue, we calculated the overall time for the projection velocities to reach 0.1° /sec. HMD angular velocity (M=141ms, SD=15ms) was the fastest in response to the cue. This was followed by the index fingertip (M=100ms, SD=8ms), and the right hand (M=125ms, SD=18ms). Eyes and EFRC vector velocities were rooted on the head movement and displayed exactly the same response time of the HMD. It was notable that no eye movements were detected prior to HMD movement, even when accounting for temporal latency of 58ms for Vive Pro Eye [26], when calculating their relative angular velocity with respect to the HMD.

From the velocity profiles, 5 target placement samples from the right eye showed significant lags in response time and higher peaks in magnitude. These samples were spread over all distances, and showed no recognizable patterns related to horizontal, vertical, or oblique target placements. Therefore, we considered these graphs as outliers and removed them, attributing their unique profiles to high noise and subject variance, before calculating mean fixation time for the right eye.

For projection velocities to decay to 5° /sec from their peaks, the right hand took the most time to stabilize (M=1946ms, SD=331ms). This was followed by the right EFRC vector (M=1333ms, SD=138ms), HMD (M=1249ms, SD=162ms), and finally the right eye (M=1177ms, SD=287ms). The eye shows faster fixation trend in the progression plot, but its high speed and noise levels hinder the calculation of exact fixation time.

Visually distinguishing individual graphs for different target distances is challenging due to overlaps and noise levels in projection error. In case of angular target placements, it is equally difficult to visually separate target graphs corresponding to the same line of sight for all data types.

D. DISCUSSION ON DIFFERENT DATA TYPES

Eye-tracking offers much lower projection error than HMD projection in terms of accuracy and displays the fastest fixation among the tested data types. The application of eye-tracker can significantly improve the performance of an angular prediction model. However, accuracy of the the Vive eye-tracker drops drastically as users gaze at vertically placed targets. For targets in oblique visual fields, the user's tendency to look short of a target is strong, resulting in a larger measurement error of the eye-tracker.

The temporal velocity profile of the eyes indicates significant variances in fixation time on a target, attributed to randomly generated saccades, blinking, or intentional eye movements during fixation across different subjects. Additional variance might be added due to the quicker movement speed of eyes during fixation and saccades. Although the eye profiles exhibits higher velocity compared to other data profiles, it still struggles to distinctly differentiate between different targets. Considering the larger measurement errors in eye deviation from the central field, this lack of differentiation can introduce errors at the initiation phase of eye movement. This suggests an opportunity where the integration of sensors with different modalities can provide more accurate data at the onset of pointing, contributing to a more reliable prediction model.

HMD and ray-casting features like EFRC show stable velocity profiles for differentiating between different angular displacements of targets. They also show good response time to start audio cue, which helps in providing pose information from the early stage of pointing. The index finger shows more varied velocity profiles when compared to other features, but can still differentiate between different target placement angles, and provides essential data for calculating ray-casting vectors. Its flexing behavior relative to the hand can also provide information on the detection of the user's pointing intention.

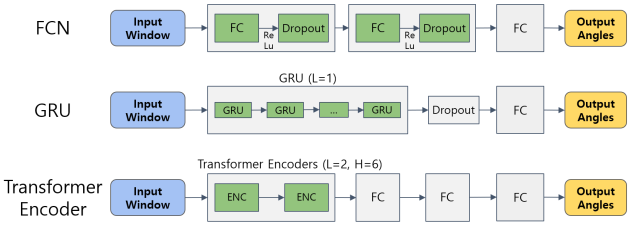


FIGURE 9. We construct a Fully-Connected-Network (FCN), and a Transformer-encoder network for comparison with our GRU model.

V. OVERVIEW OF PREDICTION MODEL

Based on the previous data analysis, we build a target angle prediction model using a recurrent neural network architecture (Fig. 9). Our model receives an input window of multi-frame features to track the changing trends in inputs for predicting angular directions toward the target of interest. We assume an extended arm pointing gesture with an extended index finger, following the data collection process.

To construct our model, we implemented a recurrent layer structure of GRU cells due to their comparable performance on long memory retention of LSTM cells, with lower computational cost [30]. The GRU model consists of single-stacked GRU cells, connected to a dropout layer and a Fully Connected (FC) linear layer.

A. MODEL INPUT WINDOW

To continuously operate during pointing, our model needs to adapt to input features from many different time periods. For the early phase of pointing where data offers limited clues to accurately predict the target, a time series input of multiple frames can provide changing trends in input features and improve the prediction accuracy. Following this assumption, we processed our input as a sequence of sliding windows. The window includes the history of past and current data from both the motion capture system and eye-tracker. Our model input takes the form of a 2-dimensional matrix where each row represents a single frame, stacked up to form a single input window. For each row, features corresponding to the same frame are all concatenated and collapsed into a 1d vector for stacking.

Results from the velocity progression analysis suggest that choosing an appropriate window length is crucial to capture the divergence of feature values for different target placements. We believe a model trained with sliding window inputs can be robust against random noise and slight data variances between subjects or within subjects.

B. INPUT FEATURES

Input features within each frame of the input window belong to three categories: motion features, eye-tracker

TABLE 2. Input feature categories.

Category	Features	Form
Motion Features	HMD linear velocity HMD angular velocity Dominant hand transform Dominant hand linear velocity Dominant hand angular rotation Index fingertip position Index fingertip linear velocity	Transformation matrix, reduced (3x3) Calculated w.r.t HMD transformation
Ray-Casting Features	*HFRC* (HMD) *HFRC* delta (HMD) *EFRC* (left eye, right eye) *EFRC* delta (left eye, right eye) *EFRC* · *HFRC* *EFRC* · *HFRC* delta	3x1 vector Ray starts from HMD root, or from each eye.
Eye-Tracker Features	Gaze ray (left eye, right eye) Gaze ray delta (left eye, right eye)	3x1 vector Ray starts from each eye.

features, and ray-casting features (Table. 2). For all features, we also calculate the difference between subsequent frames (subsequently called delta) for representing rate of change.

a: MOTION FEATURE

Motion features take the form of a transformation matrix, holding position and rotational information. The motion features holds information of HMD, dominant hand, and its index finger.

We remove HMD transformation to prevent information leakage on global subject pose, and only use its linear and rotational delta values. Dominant hand provide information on relative movement of the index finger point, its position, rotation, and delta values included as input. For the index finger, we only use its position and linear velocity, due to instabilities in rotation data.

We did not use joint data from the arms, legs, and pelvis because their inclusion introduced additional subject variances in inputs, due to different physical dimensions and movements. All transformations are expressed with respect to HMD pose because of the stated importance of head movement in static pointing scenarios.

The transformation matrices for the motion features were trimmed down to a size of (3 × 3), 6 for rotation and 3 for translation.

b: EYE-TRACKER FEATURE

Raw eye-tracking data consists of blink detection tags, pupil position values for both eyes, pupil forward vectors for both left and right eyes, and combined gaze vector, all streamed from the Tobii eye-tracker callback thread.

Among these data, we only include the gaze forward vectors for both eyes into our input. We do not use the combined gaze vector due to the undisclosed internal calculation algorithm within the released software. Pupil root values are used for calculating the ray-casting features but are not utilized as independent features in our input. For providing rate of change information, we compute delta values for gaze vector elements.

c: RAY-CASTING FEATURE

Ray-casting features are included to consider the directional relationships between the joints. Head-Ray-Cast (HRC) assumes a *cyclops eye*, where the eye gaze is represented as a single vector, pointing from the center point of the head. We employ a EFRC vector, featured in works on target angle prediction [7], [16]. We use an additional Head-Finger-Ray-Cast (HFRC) vector, drawn between HMD root and index fingertip. We also feed inner products between the HFRC and EFRC vectors, which serve as a metric to detect convergence of rays to the visual center point of HMD. To represent rate of changes in values between subsequent frames, we calculated delta values between vector elements.

C. MODEL OUTPUT

Our model's output are azimuth and elevation angles to the target in relation to the HMD orientation for the current frame. Target distance prediction was excluded after pre-testing a model that exhibited unstable oscillating behavior in distance prediction values.

With the realistic eye-tracker accuracy of 1.1° at the center, the edges of the calculated error cone began to form a parallel line under approximately 1.69 meters for the eye-tracker data, occasionally predicting the target distance at infinity, or instead being fixed to a position under this distance. Based on these observations, we believe the employed eye-tracker lacks the accuracy for reliably inferring the distance to pointing targets with parallax effect, thus constraining the scope of our work to a 2D pointing prediction problem.

D. MODEL TRAINING

We divided the 20 available subjects into groups of 14, 4, and 2 for training, testing, and validation purposes. This allocation ensures that our model encounters unseen subject data during the validation and testing phases. Due to the limited number of testing subjects, we created three sets of subject combinations for training, testing, and validation. While creating datasets, we downsampled our data from 120 FPS to 30 FPS to reduce computation cost and memory requirements. Each set was used to train an individual model, and the test results were averaged to produce the final results for evaluation.

VI. EVALUATION OF POINTING PREDICTION MODEL

We demonstrate the accuracy of our model in three aspects: 1) an ablation study of input features, 2) an analysis for different input window lengths, and 3) performance comparisons for different neural network structures. Our comparison includes overall RMS prediction errors and progression plots illustrating errors over the pointing sequence.

For comparing performance for different models, we produce overall RMSE by averaging all angular deviance values in all frames for all output sequence. We then plot

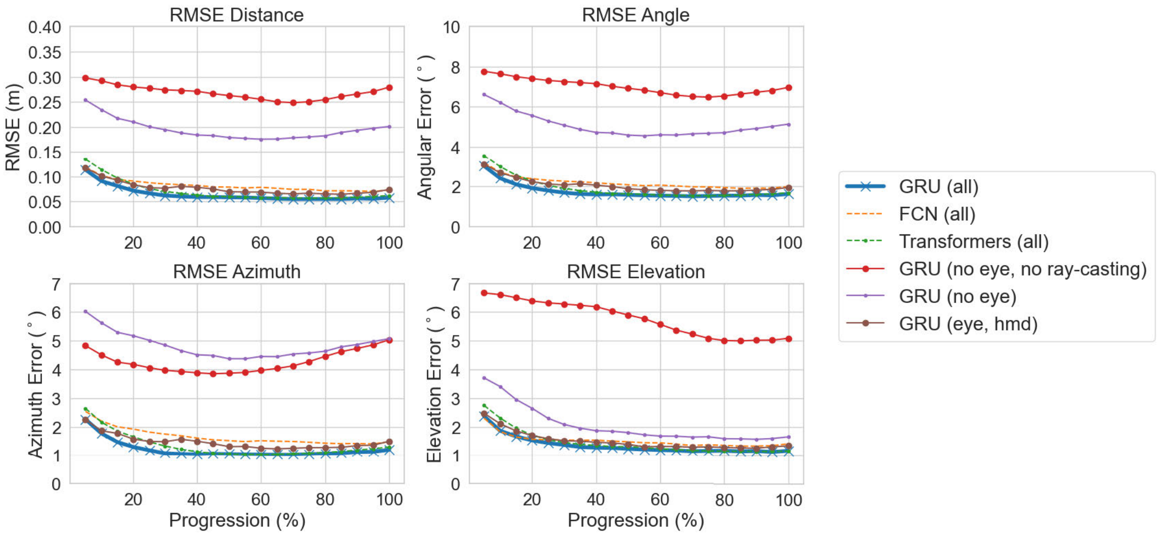


FIGURE 10. Output error progression plots for a normalized pointing sequence.

TABLE 3. Overall RMS error and its standard deviation (in parenthesis) for the ablated models, FCN and transformer models.

Models	Azimuth Error (°)	Elevation Error (°)	Deviation Angle (°)	Euclidean Error (cm)
GRU (no gaze, no raycast)	3.35 (2.75)	4.65 (3.64)	6.16 (3.45)	22.8 (15.8)
GRU (eye, HMD)	1.10 (1.03)	1.14 (1.01)	1.71 (1.16)	6.1 (4.6)
GRU (no gaze)	3.77 (3.11)	1.62 (1.44)	4.20 (2.86)	15.6 (12.5)
GRU (all)	**0.92 (0.86)**	**1.03 (0.90)**	**1.48 (0.99)**	**5.3 (0.04)**
FCN (all)	1.31 (1.14)	1.22 (0.96)	1.91 (1.16)	7.0 (4.9)
Transformer (all)	1.00 (0.98)	1.14 (1.00)	1.63 (1.15)	5.9 (4.7)

progression error plots to represent temporal performance for various setups. We start by resampling all prediction sequence data and error data series to a consistent 20-frame duration based on the trained model's results. Subsequently, we calculate the RMSE series for all output sequences and average them to generate a single, representative, progression plot (Fig. 10). Our model requires a historical input of 20 frames or 0.67 seconds (including the current frame) for each sequence. Consequently, all progression outputs from the model throughout each pointing sequence start from 20 frames after the initiation of pointing.

A. INPUT FEATURE ABLATION BASELINES

We conducted an ablation study encompassing four different input feature combinations to investigate the impact of eye-tracker data on prediction performance (Table. 3).

Our initial baseline excluded any eye-tracker derived data, which removes pupil vectors, ray-casting vectors, and vector products between ray-casting features. This model achieved an overall deviance of 6.16° (SD=3.45°) or minimum 5.0° at 80% progression, showing lower accuracy than the mean error deviance reported by a previous study [13], which reports a deviation range of 3.2°-4.0° for distances between 2.8m and 3.8m. The decrease in performance could be attributed to using inputs from varied progression points and randomized placement data during collection.

The second baseline assumed an HMD equipped with eye-tracking, but lacking controller or motion tracker support. With only HMD transforms and pupil vectors available, the model delivered a significant performance leap, achieving an RMSE of 1.71° (SD=1.16°).

The third baseline excluded only the gaze vectors, considering configurations where manual acquisition of eye root position data is feasible even without eye tracking, thus including ray-casting features and their derivatives. This setup attained an angular deviance of 4.20° (SD=2.86°).

We utilized all previously defined features in our final case, achieving overall angular error of 1.48° (SD=0.99°). The fully featured model also demonstrated a faster decrease in error values over the initial pointing period, and exhibited improved maximum accuracy.

The error progression plots of our recurrent model revealed distinct patterns between models using gaze-vectors and those without them. Baselines excluding eye-tracker gaze vector showed decreasing errors in azimuth angle prediction during progression, only to exhibit noticeable increases at the pointing termination, resulting in a concave profile. In contrast, input setups featuring eye-tracking data showed continuously decreasing trends, leading to points of stabilization where errors remained relatively unchanged for the rest of the progression.

B. NEURAL NETWORK BASELINES

We pick two widely used neural network architectures for comparison with our GRU network (Fig.9): a) a simple

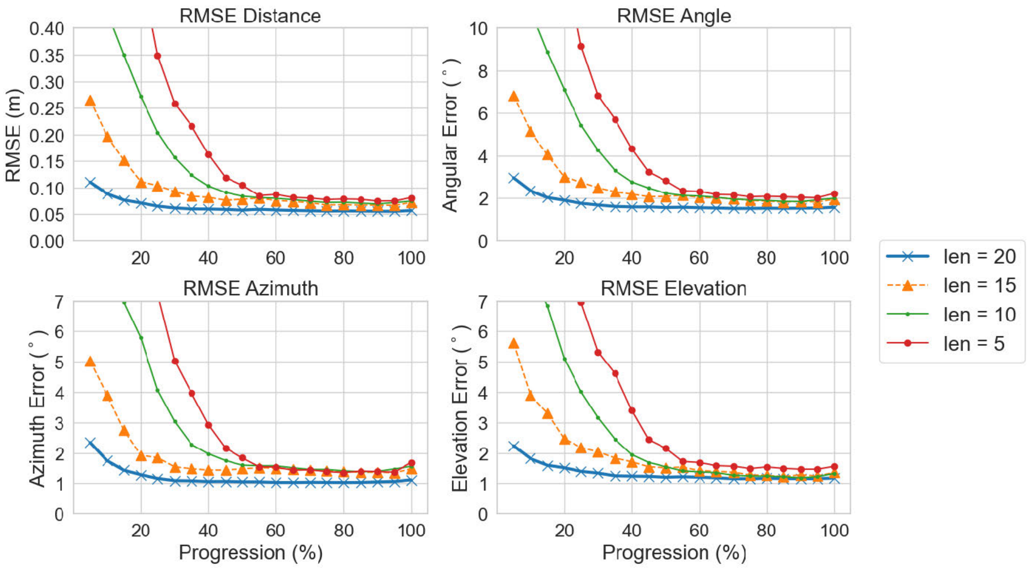


FIGURE 11. Output error progression plots for different input window lengths.

Fully-Connected-Network (FCN) with dropouts and b) a Transformer encoder model [31].

The FCN model consists of a double module structure with a linear output layer, where each module consists of an activated linear layer, connecting to a dropout layer. For the transformer model, we use two 1D transformer encoder layers from Pytorch, connected to a sequence of linear layers.

All three models, sharing the same input and output dimensions, use the same datasets for training and testing, with input window length of 20 frames. Our GRU neural network outperformed the other two models, achieving the lowest overall RMSE value.

C. WINDOW LENGTH ON MODEL PERFORMANCES

In analyzing the impact of window length on model performance, we observed error progression plots for window lengths of 0.67s, 0.41s, 0.33s, and 0.17s, corresponding to 20, 15, 10, and 5 frames at a 30 FPS input sampling rate (Fig. 11). We trimmed off starting frames of the test sequence data for all testing models, so that prediction results for all comparison cases start from the same time.

The model with the 20 frames input window achieved the fastest saturation to max performance, and the lowest RMSE. Models trained with 15 frame and 10 frame windows achieved errors smaller than target radius of 7.6cm at 50% and 80% output progression, respectively. The 5-frame window model failed to produce overall errors that fits inside our target radius throughout the entire output progression.

VII. DISCUSSION

Our model, incorporating eye-tracker data, achieves approximately 2.5 times greater accuracy than baselines that lack eye-tracker gaze vector data in overall RMSE. The maximum mean accuracy of approximately 1.48° is achieved when the input window length is sufficiently long for capturing key progression trends in the input window.

Feature ablation results indicate that eye-tracker gaze vectors significantly enhance the accuracy of target angle predictions. According to raw data analysis, eye ray projection plot shows much lower error values compared to head ray. The stabilization of eye was also the fastest in the angular velocity profile analysis. We believe the eyes' high projection accuracy, coupled with fast stabilization time, attributed to significant improvements in the model performance. The following model evaluation shows that overall eye-tracker projection errors set a theoretical performance ceiling for our neural network model. It is worth pointing out that information from the HMD, index finger, and local eye root positions is still essential to reach this ceiling. Comparatively, models using only HMD transform and eye-tracker ray inputs report lower performance than a fully-featured model.

Previous studies on eye-tracker performance have shown that its measurement data can produce errors exceeding 10 times the specified maximum target deviation angle range [26]. In such cases, eye-tracker data may potentially offer less accuracy compared to data from other features, especially during the initial phase of pointing. This can provide explanation on why our fully featured model performed better over the HMD and eye-tracker features baseline. Improvements in measurement accuracy for visual field periphery in HMD integrated eye-trackers can improve early target prediction capabilities for our target prediction approach.

A. LIMITATIONS AND FUTURE WORK

During training and evaluating our model, we identify three challenges in acquiring accurate prediction values: a) subject variances in the pointing data, 2) small sample sizes and follow up bias effects that hinder efficient training and testing, and 3) dataset scales that can result in inefficient training for larger neural network structures.

Inter-subject and within-subject variances in the data present challenges in identifying systematic human behaviors for the pointing features. User data typically exhibits a positively skewed distribution for both task completion and projection errors. These behaviors are due to occasional slow and inaccurate movements during pointing. Inefficient pointing cases can vary among different individuals, and make it difficult for our model to predict fixation time or final fixation pose of the user.

A large sample size can counter the effect of subject variance by providing coverage for all potential human behaviors in pointing. Our initial sample size (N=20 after post processing) proved to be too small to account for all potential variances in pointing. Lack of sample sizes was exacerbated in neural network model training and evaluation, where samples had to be split for training and testing purposes, further lowering diversity of subjects for each of the dataset. Furthermore, existence of only right-handed subjects in the data introduced bias for right-handed subjects. This is also true for eye dominance, where our data indicated prevalence for right eye dominant subjects in the sample [32]. Collection of larger samples should be followed by applications of sufficient data augmentation techniques, such as mirroring or noise augmentations.

Our recurrent model provides the optimal solution for smaller sized datasets, but its efficacy for training with much larger sample sizes or for predicting outputs with higher dimensions, has to undergo further comparisons with scale-appropriate model structures in the future.

Although we only considered static standing cases in our research, human can also use deictic gestures during walking. Variations in walking speed affect target prediction errors and selection time [28], while peripheral vision also plays a significant role in object exploration [33]. While our work focused on static pointing with identical standing pose, approaches to integrate dynamic pointing scenarios with different body pose configurations in the future will better reflect real life use for human pointing.

VIII. CONCLUSION

We present a prediction study on pointing target direction during pointing progression, employing motion and eye-tracker-based features. Our recurrent network model operates throughout the pointing process and significantly improves accuracy by incorporating eye-tracking data along with motion and ray-casting features. The addition of motion-based features helps reduce errors at start of pointing and maximizes the overall accuracy.

During the initial phase of pointing, eye-trackers may provide less accurate readings due to uneven performance over the visual field. However, sensors from other modalities consistently offering accurate data during pointing can partially compensate for the eye-tracker's shortcomings. This is evident in faster saturation of RMSE values for our whole-feature model compared to the model with limited input features. Improvements in HMD-integrated eye-tracker performance and calibration techniques can enhance maximum accuracy and earlier prediction of target positions in future works.

REFERENCES

[1] Y. Wei, R. Shi, D. Yu, Y. Wang, Y. Li, L. Yu, and H.-N. Liang, "Predicting gaze-based target selection in augmented reality headsets based on eye and head endpoint distributions," in *Proc. CHI Conf. Human Factors Comput. Syst.*, Apr. 2023, pp. 1–14.

[2] N. M. Gamage, D. Ishtaweera, M. Weigel, and A. Withana, "So predictable! Continuous 3D hand trajectory prediction in virtual reality," in *Proc. 34th Annu. ACM Symp. User Interface Softw. Technol.*, Oct. 2021, pp. 332–343.

[3] A. Clarence, J. Knibbe, M. Cordeil, and M. Wybrow, "Unscripted retargeting: Reach prediction for haptic retargeting in virtual reality," in *Proc. IEEE Virtual Reality 3D User Interfaces (VR)*, Mar. 2021, pp. 150–159.

[4] R. Henrikson, T. Grossman, S. Trowbridge, D. Wigdor, and H. Benko, "Head-coupled kinematic template matching: A prediction model for ray pointing in VR," in *Proc. CHI Conf. Human Factors Comput. Syst.*, Apr. 2020, pp. 1–14.

[5] T. W. H. Koehler, M. Pruzinec, T. Feldmann, and A. Worner, "Automatic human model parametrization from 3D marker data for motion recognition," in *Proc. WSCG Commun. Papers*, 2008, pp. 211–216. [Online]. Available: http://wscg.zcu.cz/DL/wscg_DL.htm

[6] M. Sousa, R. K. dos Anjos, D. Mendes, M. Billinghurst, and J. Jorge, "Warping deixis: Distorting gestures to enhance collaboration," in *Proc. CHI Conf. Human Factors Comput. Syst.*, May 2019, pp. 1–12.

[7] S. Mayer, J. Reinhardt, R. Schweigert, B. Jelke, V. Schwind, K. Wolf, and N. Henze, "Improving humans' ability to interpret deictic gestures in virtual reality," in *Proc. CHI Conf. Human Factors Comput. Syst.*, Apr. 2020, pp. 1–14.

[8] T.-S. Dalsgaard, J. Knibbe, and J. Bergström, "Modeling pointing for 3D target selection in VR," in *Proc. 27th ACM Symp. Virtual Reality Softw. Technol.*, Dec. 2021, pp. 1–10.

[9] M. R. Mine, "Virtual environment interaction techniques," UNC Chapel Hill CS Dept., Univ. North Carolina, Chapel HillChapel Hill, NC, USA, Tech. Rep., 1995, doi: 10.5555/897820.

[10] M. R. Mine, F. P. Brooks, and C. H. Sequin, "Moving objects in space: Exploiting proprioception in virtual-environment interaction," in *Proc. 24th Annu. Conf. Comput. Graph. Interact. Techn.*, 1997, pp. 19–26.

[11] J. S. Pierce, A. S. Forsberg, M. J. Conway, S. Hong, R. C. Zeleznik, and M. R. Mine, "Image plane interaction techniques in 3D immersive environments," in *Proc. Symp. Interact. 3D Graph.*, 1997, p. 39.

[12] F. Argelaguet, C. Andujar, and R. Trueba, "Overcoming eye-hand visibility mismatch in 3D pointing selection," in *Proc. ACM Symp. Virtual Reality Softw. Technol.*, Oct. 2008, pp. 43–46.

[13] K. Plaumann, M. Weing, C. Winkler, M. Müller, and E. Rukzio, "Towards accurate cursorless pointing: The effects of ocular dominance and handedness," *Pers. Ubiquitous Comput.*, vol. 22, no. 4, pp. 633–646, Aug. 2018.

[14] J. M. Foley and R. Held, "Visually directed pointing as a function of target distance, direction, and available cues," *Perception Psychophysics*, vol. 12, no. 3, pp. 263–268, May 1972.

[15] S. Mayer, K. Wolf, S. Schneegass, and N. Henze, "Modeling distant pointing for compensating systematic displacements," in *Proc. 33rd Annu. ACM Conf. Human Factors Comput. Syst.*, Apr. 2015, pp. 4165–4168.

[16] S. Mayer, V. Schwind, R. Schweigert, and N. Henze, "The effect of offset correction and cursor on mid-air pointing in real and virtual environments," in *Proc. CHI Conf. Human Factors Comput. Syst.*, Apr. 2018, pp. 1–13.

[17] D. Purves, G. Augustine, D. Fitzpatrick, L. Katz, A. LaMantia, J. McNamara, and S. Williams, "Neuroscience. types of eye movements and their functions," Sinauer Associates, Sunderland, MA, USA, Tech. Rep., 20, Mar. 2001.

[18] R. J. Krauzlis, L. Goffart, and Z. M. Hafed, "Neuronal control of fixation and fixational eye movements," *Phil. Trans. Roy. Soc. B, Biol. Sci.*, vol. 372, no. 1718, Apr. 2017, Art. no. 20160205.

[19] S. Martinez-Conde, J. Otero-Millan, and S. L. Macknik, "The impact of microsaccades on vision: Towards a unified theory of saccadic function," *Nature Rev. Neurosci.*, vol. 14, no. 2, pp. 83–96, Feb. 2013.

[20] M. Spering, "Eye movements as a window into decision-making," *Annu. Rev. Vis. Sci.*, vol. 8, no. 1, pp. 427–448, Sep. 2022.

[21] D. Miniotas, "Application of fitts' law to eye gaze interaction," in *Proc. CHI Extended Abstr. Human Factors Comput. Syst.*, Apr. 2000, pp. 339–340.

[22] L. Sidenmark, D. Mardanbegi, A. R. Gomez, C. Clarke, and H. Gellersen, "BimodalGaze: Seamlessly refined pointing with gaze and filtered gestural head movement," in *Proc. ACM Symp. Eye Tracking Res. Appl.*, Jun. 2020, pp. 1–9.

[23] L. Sidenmark, C. Clarke, X. Zhang, J. Phu, and H. Gellersen, "Outline pursuits: Gaze-assisted selection of occluded objects in virtual reality," in *Proc. CHI Conf. Human Factors Comput. Syst.*, Apr. 2020, pp. 1–13.

[24] K. Yamada, Y. Sugano, T. Okabe, Y. Sato, A. Sugimoto, and K. Hiraki, "Attention prediction in egocentric video using motion and visual saliency," in *Proc. Pacific-Rim Symp. Image Video Technol.* Cham, Switzerland: Springer, 2011, pp. 277–288.

[25] L. Sidenmark and H. Gellersen, "Eye&head: Synergetic eye and head movement for gaze pointing and selection," in *Proc. 32nd Annu. ACM Symp. User Interface Softw. Technol.*, Oct. 2019, pp. 1161–1174.

[26] A. Sipatchin, S. Wahl, and K. Rifai, "Eye-tracking for clinical ophthalmology with virtual reality (VR): A case study of the htc vive pro eye's usability," *Healthcare*, vol. 9, no. 2, p. 180, 2021.

[27] R. Yao, T. Heath, A. Davies, T. Forsyth, N. Mitchell, and P. Hoberman, "Oculus VR best practices guide," *Oculus VR*, vol. 4, pp. 27–35, Jul. 2014.

[28] Y. Lu, B. Gao, H. Tu, H. Wu, W. Xin, H. Cui, W. Luo, and H. B.-L. Duh, "Effects of physical walking on eyes-engaged target selection with ray-casting pointing in virtual reality," *Virtual Reality*, vol. 27, no. 2, pp. 603–625, Jun. 2023.

[29] H. Rambold, A. Sprenger, and C. Helmchen, "Effects of voluntary blinks on saccades, vergence eye movements, and saccade-vergence interactions in humans," *J. Neurophysiol.*, vol. 88, no. 3, pp. 1220–1233, Sep. 2002.

[30] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, "Empirical evaluation of gated recurrent neural networks on sequence modeling," 2014, *arXiv:1412.3555*.

[31] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *Proc. Adv. Neural Inf. Process. Syst.*, 30, 2017, pp. 1–11.

[32] J. Aswathappa, K. Kutty, and N. Annamalai, "Relationship between handedness and ocular dominance in healthy young adults—A study," *Int. J. Pharm. Biomed. Res.*, vol. 2, no. 2, pp. 76–78, 2011.

[33] Q. Zhou, D. Yu, M. N. Reinoso, J. Newn, J. Goncalves, and E. Velloso, "Eyes-free target acquisition during walking in immersive mixed reality," *IEEE Trans. Vis. Comput. Graphics*, vol. 26, no. 12, pp. 3423–3433, Dec. 2020.

CHOONGHO CHUNG (Student Member, IEEE) received the B.S. degree in mechanical engineering and the M.S. degree in culture technology from KAIST, Daejeon, South Korea, in 2014 and 2018, respectively, where he is currently pursuing the Ph.D. degree with the Graduate School of Culture Technology. His research interests include telepresence applications, human attention recognition, and the generation of character animation.

SUNG-HEE LEE (Member, IEEE) received the B.S. and M.S. degrees in mechanical engineering from Seoul National University, Seoul, South Korea, in 1996 and 2000, respectively, and the Ph.D. degree in computer science from The University of California, Los Angeles, CA, USA, in 2008. He is currently a Professor with the Graduate School of Culture Technology, KAIST. His research interests include autonomous human animation, avatar motion generation, and human modeling. He is an Associate Editor of IEEE Transactions on Visualization and Computer Graphics.