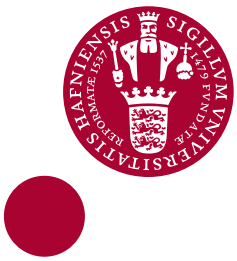


Modeling Pointing for 3D Target Selection in VR

Dalsgaard, Tor-Salve; Knibbe, Jarrod; Bergström, Joanna

Published in:
Proceedings of the 27th ACM Symposium on Virtual Reality Software and Technology

DOI:
10.1145/3489849.3489853

Publication date:
2021

Document version
Publisher's PDF, also known as Version of record

Modeling Pointing for 3D Target Selection in VR

Tor-Salve Dalsgaard
University of Copenhagen
Denmark
torsalve@di.ku.dk

Jarrod Knibbe
University of Melbourne
Australia
jarrod.knibbe@unimelb.edu.au

Joanna Bergström
University of Copenhagen
Denmark
joanna@di.ku.dk

ABSTRACT

Virtual reality (VR) allows users to interact similarly to how they do in the physical world, such as touching, moving, and pointing at objects. To select objects at a distance, most VR techniques rely on casting a ray through one or two points located on the user's body (e.g., on the head and a finger), and placing a cursor on that ray. However, previous studies show that such rays do not help users achieve optimal pointing accuracy nor correspond to how they would naturally point. We seek to find features, which would best describe natural pointing at distant targets. We collect motion data from seven locations on the hand, arm, and body, while participants point at 27 targets across a virtual room. We evaluate the features of pointing and analyse sets of those for predicting pointing targets. Our analysis shows an 87% classification accuracy between the 27 targets for the best feature set and a mean distance of 23.56 cm in predicting pointing targets across the room. The feature sets can inform the design of more natural and effective VR pointing techniques for distant object selection.

CCS CONCEPTS

• **Human-centered computing** → **Human computer interaction (HCI)**; *Pointing*; *User studies*.

KEYWORDS

Virtual reality, pointing, target selection

ACM Reference Format:
Tor-Salve Dalsgaard, Jarrod Knibbe, and Joanna Bergström. 2021. Modeling Pointing for 3D Target Selection in VR. In *27th ACM Symposium on Virtual Reality Software and Technology (VRST '21), December 8–10, 2021, Osaka, Japan.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3489849.3489853

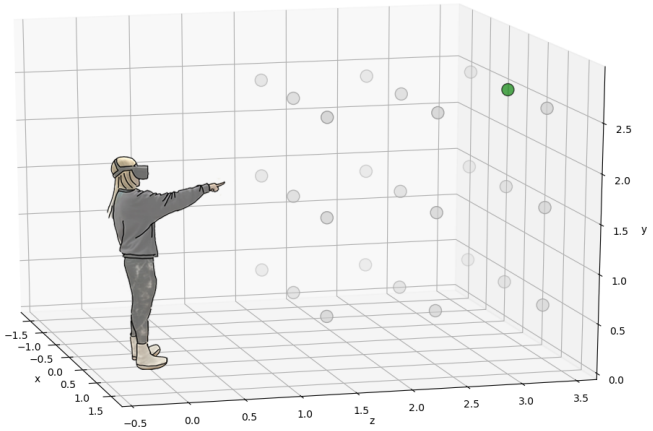


Figure 1: An illustration of our pointing study design. A user points at one distant target (green) out of 27 (grey) in a virtual room. The illustration shows the scale in meters.

Selecting objects is a key interaction with computers, whether using a mouse to select a file or tapping a touchscreen to select the next song. This is equally true in virtual reality (VR). In VR, we may select objects similarly to how we do in the physical world, for example, reaching out to touch, grab, or push them when nearby (e.g., whilst playing VR-Minecraft [3] or interacting with data visualisations [9]). Selecting objects located out-of-reach, however, has no direct counterpart in the real world.

People naturally use pointing gestures in communication to convey spatial relations and address distant objects. We learn this at a young age, for instance, to communicate desire, attention, and intention [7, 20]. Pointing can occur in a broad sense, in terms of 'that direction', or more specifically, akin to 'that one'. However, pointing forms just one part of multimodal communication, co-occurring primarily with language. As such, we are not spatially accurate in our use of pointing [33].

To select an object at a distance, most VR interfaces depart from assumptions of how people point and rely on casting a ray through two locations on the user's body (e.g., the head and a fingertip) or along one body part (e.g., along a forearm) and placing a cursor further away on that ray (e.g., [10, 11, 26, 30, 34]). These techniques may employ many forms of rays and cursors, such as projected lines and dots (e.g., [10, 26, 34]), or extended arms and hands (e.g., [11, 30]). However, by employing one or two points located on the body, these techniques force the user to align those body parts to move the cursor onto a target.

Prior studies show that when people get to choose the pointing motion that is natural to them, it does not correspond to the pointing motion performed with current ray casting techniques (e.g., [23, 29]). This is likely because the rays differ from how the human brain optimises movements for the best use of the body. Even when the closest ray casting model to natural, cursor-less pointing is employed, further offsets and targeting performance improvements can be found [23]. These results were based on models of cursorless pointing at 2D target planes and used partly real-world settings (e.g., [23, 29]). However, beyond menus and virtual displays, the

VRST '21, December 8–10, 2021, Osaka, Japan

ACM ISBN 978-1-4503-9092-7/21/12…$15.00
https://doi.org/10.1145/3489849.3489853

possible target objects in VR are often spread across wider distances, such as across rooms and other environments. The novelty in our study is to model cursorless pointing movements at targets positioned in 3D space.

We examine cursor-free pointing motions in order to derive features that best model how people would naturally point at a 3D target space in VR. To do this, we collect motion data from seven locations on the arm and body, in a study where participants point at 27 targets spaced one meter apart and at a distance across a virtual room (Figure 1). Our main contributions are:

- An analysis of features that best describe natural pointing movements. We derive two sets of features: (1) pose features and (2) motion features. The pose features are based on only the position data of the final pointing pose. This is akin to traditional interaction techniques, where the target is identified at the moment of selection (e.g., with a mouse, or ray casting). The motion features are based on data from the entire pointing movement, allowing for early prediction of intended targets.
- Estimates of how accurately intended targets can be interpreted from pointing movements. Our results show an 87% classification accuracy between the targets, and a mean distance of 23.56 cm in predicting pointing targets across the room. We present insights into the performance of the feature sets, analysing how they can generalise to inferring the user's targets in cursorless pointing with limited data from hand tracking, and with more challenging target locations.

These contributions are intended to inform development of 3D pointing techniques for distant targets in three ways. First, they can inform the design of ray casting techniques about which features could be best to use for natural and accurate ray-pointing. Second, they can inform the design of motion tracking about the points on the body that carry most information about the user's intended targets. Third, they can inform the design of cursorless pointing techniques around the classification accuracy of 3D targets in VR.

1 RELATED WORK

We examine pointing motions in order to derive features that characterise how people would naturally point at a 3D target space in virtual reality. Next we discuss literature on pointing techniques for targets at a distance in VR, background on characteristics of natural pointing, and prior work on modeling the features of pointing.

1.1 Pointing Techniques for Targets at a Distance

One of the first distant target selection techniques for human-computer interaction was in the seminal work of Put-That-There by Bolt [6]. In Put-That-There, the users could couple pointing and speech for interaction and selection of objects. This mirrored a natural way of conveying distant references in communication [7, 20] (i.e., deictic gestures [18]). Since Bolt's work, distant pointing has received much interest for interaction, especially on large displays [24], and more recently in VR [11]. Much of this work, however, has gone away from pointing as a part of multimodal communication (as employed by Bolt), and instead looked to employ pointing as a precise targeting technique.

Ray casting is one of the most common techniques to select distant targets. In ray casting, the ray follows a straight line originating from the user until it intersects a distant screen or target. Argelaguet et al. [2] categorises ray casting methods into two families: hand- and eye-rooted techniques. Prominent hand-rooted ray casting methods include Index Finger Ray Cast, which casts a ray extending the index finger [10], and Forearm Ray Cast, where the ray is an extension of the forearm [26]. Eye-rooted techniques are split further into two methods: Gaze Ray Cast, which uses the eyes' gaze to cast a ray [26] and Eye-Finger Ray Cast, which casts a ray based on the line between the eye and index fingertip [24]. The head and hand have been used together also in velocity-based models for predicting pointing targets [14]. Ray casting is often defined as 'natural' [17]; it is simple, can be used hands-free (does not require a controller), and aims to build upon every day pointing gestures [35]. Ray casting techniques, however, use a cursor. Thereby, they force the user to adapt their pointing gestures for some given pose features (e.g., by aligning the gaze and fingertip) based on the feedback they provide with the cursors.

As an adaptation of ray casting, Feuchtner et al. [11] proposed extending users' body parts to enable them to reach any object. On the one hand, this brings all objects into peri-personal space, allowing touch-based interaction. On the other hand, it proposes an 'unnatural' setting with the extended arm. Feuchtner et al.'s motivation for this are to facilitate direct interaction, where controls and effects are coupled onto the object [31]. This 'extended arm' technique draws on Poupyrev et al.'s [30] Go-Go Interaction technique, where the user could 'throw' their hand at a distance. Again, Go-Go Interaction was motivated in the pursuit of naturalness, allowing users to 'extend their arm' towards any object. However, these techniques also rely on a ray-based method with a cursor: the hand extended or thrown into the distance follows a ray, such as an elongated forearm between an elbow and the wrist (see e.g., the survey by Argelaguet and Andujar [1] of such techniques). Thereby, they leave open the question of whether unconstrained, cursorless pointing can support the design of better cursor-based techniques.

1.2 Characteristics of Natural Pointing

Pointing is frequently used in communication. The gesture is classified as deictic, used to add context to sentences and actions [7]. Pointing is learned already in early childhood, can be used imperatively (indicating desire), declaratively (directing attention), or epidemically (requesting information). In all purposes, it is used to address intention towards the objects of interest.

Kendon [18] describes different postures when pointing at different distances and objects. For instance, the extended index finger refers to a specific object, location, or human, while an open hand gesture indicates a more loose direction. Kendon also observes that different distances are indicated by different poses of the arm: When the arm is stretched, the object of interest is more distant, compared to when the arm is angled. Together these suggest that pointing gestures contain patterns of information about the intent of the person performing them.

Such features of natural pointing could be used in interpreting a user's intended targets. To date, however, natural pointing, as a

contextual part of multimodal communication (meant to convey information to someone else), has received little attention in HCI. Instead, 'natural' pointing is often used as a synonym for cursorless, or simply "hands-free" pointing, where the user seeks to select distant objects for themselves. For example, Cockburn et al. [8] use the latter form of 'natural' pointing, in their seminal paper on cursorless "Air Pointing" techniques. While the nature of pointing is changed, the same intention holds: to derive features that convey information about the intended target.

1.3 Modeling the Features of Pointing

Two recent papers suggest how to derive features with data collected from pointing at targets laid out on a 2D plane. Mayer et al. [23] used cursoless pointing to build a model for correcting offsets in ray casting, and Plaumann et al. [29] studied the effects of eye dominance and handedness on pointing accuracy for smart homes. In addition, Kopper et al. [19] uses a similar approach to Mayer et al., modeling pointing at a distance on a 2D plane. However, Kopper used a controller-based cursors on rays to improve Fitts' Law models, whereas Mayer used free-hand pointing to find how the rays, and thereby models of pointing movements, can be improved.

Mayer et al. [23] first collected data from participants pointing at a point on a 2D surface with no cursor. They then took the pose and fitted four types of rays on it (Index finger Ray Cast, Gaze Ray Cast, Eye-finger Ray Cast, and Forearm Ray Cast), and compared how close the point the ray showed was the point the participants actually pointed at (i.e., offset). Thereby, these offsets represent how close a natural pointing pose is to a pose that ray casting forces the user to adopt by showing a cursor in pointing tasks. Mayer et al. found that the average offset was smallest in Eye-finger Ray Cast, and built a model to correct that offset. Their model offers accuracy improvements of 33% over standard ray casting approaches. Prior to correction, they found an average pointing error of up to > 60 cm with Index finger Ray Cast when pointing at targets on a vertical diplay-like 2D surface from over a 3 meter distance to the targets without a cursor. With their model, they reduce this error to 38.4 cm. Plaumann et al. [29] further showed that embedding ray casting methods with information on ocular dominance and handedness can improve accuracy in cursorless pointing. Thereby, ray casting models can be improved with features captured from cursorless pointing movements so as to increase pointing performance and to allow users to point as they naturally would.

Both Mayer's and Plaumann's studies describe large errors in cursorless pointing at distant targets, but use targets laid on a 2D plane similar to a large wall-display. To date, cursorless pointing has not been modelled at a distant 3D target space in virtual reality. Our study is inspired by the studies of Mayer and Plaumann on cursorless pointing to investigate the features of pointing movements at distant targets in a 3D virtual environment. We look to identify features of pointing that can inform pointing techniques with cursors (to fit the rays) so as to embrace natural pointing poses, and also classification and prediction techniques for interpreting the user's intended targets without a cursor.

2 DATA COLLECTION

We conducted a data collection study to collect data on pointing movements in VR, that enables us to:

(1) Analyse the features that can be used for interpreting pointing targets.
(2) Estimate how accurately pointing targets can be classified and predicted.

In the study, 13 participants were asked to point at 27 targets spread across a large cube-shaped grid in a virtual room. The targets were out of arm's reach. In the virtual environment, the participants controlled an avatar from a first-person perspective, whose movements were tightly coupled to their own.

We designed the task to allow us to model natural pointing movements. We made no assumptions about the users' pointing, no cursor was presented, and no feedback of pointing performance was provided.

2.1 Participants

The study was conducted with 13 participants, with a mean age of 24.46 (SD±1.98) years. Five participants identified themselves as females, seven as males and one as another gender. The mean of the participants' forearm length (28.0 cm) fit within the standard deviation range of the anthropometric data (27.8±2.30 cm) reported in Gordon et al. [13], indicating that our sample can be considered representative. We recruited only right-handed participants due to the technical setup. 11 participants had no prior experience with VR. The participants were compensated for their time with gifts corresponding to approximately 15€ in value.

2.2 Study Design

In this study, we followed a within-subjects design, where each participant performed pointing movements at 27 targets. This number was chosen to include the same number of targets in all three dimensions. The 27 targets were presented one at a time, and repeated five times, resulting in 135 trials per participant. The number of repetitions was chosen to keep the study duration in 30 minutes, as otherwise pointing with an entire arm repeatedly could cause significant fatigue bias on the movement data. The order of the 135 pointing trials was randomised.

The targets were arranged in a 3 x 3 x 3 grid, spread over 2 m^3, and spaced 1 m apart. The closest targets were 150 cm from the participant's standing location. The lowest targets were 49 cm off the ground, the highest 249 cm above ground, and the middle targets were at 149 cm (the average human shoulder height [13][1]). The targets were spheres with a 15 cm diameter. We chose to use a single target size, as in this cursorless pointing, there are no errors, and thus no reason to vary the size as in Fitts' Law studies. Furthermore, we considered this size at the given distances to be small enough to not introduce extra noise due pointing at different locations within a target's surface or space. Figure 1 visualizes the target grid, in which the x-axis corresponds the left-right direction, the y-axis the vertical direction (height), and the z-axis the frontal direction (depth).

[1]Data collected from U.S. adults. See also: https://www.ergocenter.ncsu.edu/wp-content/uploads/sites/18/2017/09/Anthropometric-Summary-Data-Tables.pdf *(accessed: 10.121.20)*

We had three aims in designing the target grid. First, its size should cover a large space that is representative of possible targets in virtual rooms. Second, the target spread should provoke detectable changes in arm poses. Third, it should also present more challenging classification tasks, such as with targets directly in front of a user at different depths right at shoulder height.

2.3 Task

The participants were asked to point at a displayed target with the right index finger, with targets appearing one at a time. We chose to display the targets one at a time for the sake of experimental control and to not introduce distractor targets, following the guidelines for object selection studies in VR [5]. To promote natural pointing, the participants were instructed to point as they would to direct a friend's attention to the target.

The participants held a controller in their left (non-dominant) hand, leaving their dominant, pointing hand empty. They were instructed to rest their right hand over their belly in between the pointing trials. To begin a pointing trial, the participants pressed the controller's touchpad. When the touchpad was pressed, the next target appeared. The participants lifted their right hand to point at the target, and then pulled the controller's trigger (with their left hand) to indicate they were pointing at it. When pulling the trigger, the target disappeared, and the participants returned to the resting position with their hand over their belly.

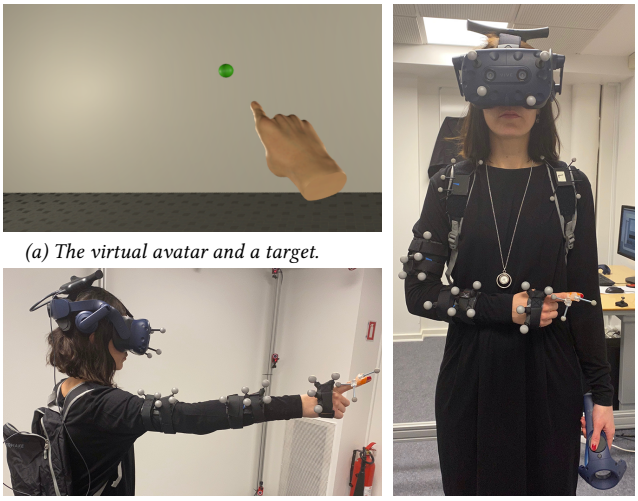
2.4 Procedure

Before starting the experiment, the participants were informed about the purpose of the study. We attached marker sets for motion tracking and measured their placement on the participant's arm. The participants put on the head-mounted display (HMD) for VR (a wireless HTC Vive Pro), were familiarised with the controller on their left hand and its touchpad and trigger, and had a chance to look around the virtual room to get accustomed to it and the virtual avatar (figure 2(a)). The participants were instructed to stand on a marked location on the virtual floor for the duration of the experiment.

Next, the participants went through a calibration task. For this, the participant assumed three positions: (1) pointing straight down, (2) pointing to the right, and (3) pointing straight ahead with the arm at shoulder level. This calibration task enabled the collection of soft biometric data from the participant, which enabled the avatar's size to be matched to the participant, and could be used later for data normalisation. After calibration, the participants performed the main part of the experiment, pointing at targets as described above.

2.5 Setup

The participant's movement was tracked at seven locations on the body (figure 2(b) and (c)). These were tracked with an OptiTrack motion capture system (24 cameras) and seven rigid bodies holding reflective markers. The rigid bodies were attached to the participant's left and right shoulder, upper arm, forearm, hand, index finger and the HMD. OptiTrack markers on the arm and hand were mounted using 3D printed models, based on the design by Mayer et al. [23][2]. The shoulder markers were attached on a backpack's straps, which also housed the wireless HMD's battery.



(a) The virtual avatar and a target.

(b) Side view. *(c) Front view.*

Figure 2: Figure (a) shows the virtual avatar pointing. Figures (b) and (c) show a user wearing the OptiTrack marker sets.

For each pointing trial (between the start touchpad press and the finishing trigger), we logged the movement of each of the seven rigid bodies and all their reflective markers with a 50 Hz sampling frequency. One sample consists of the position and orientation of each tracked rigid body and a timestamp. The orientational data is logged as Euler angles (a triplet of pitch, yaw and roll). The data is logged such that the positions of rigid bodies are in meters.

3 FEATURES OF POINTING

From the collected motion data we extract sets of features, describing the pointing movement, based on different interaction principles and hardware limitations. We analysed features across the entire pointing movement, starting by analysing a set of features derived from the last point of movement (the pointing pose at the moment of triggering selection) and then a set of features derived from the entire pointing gesture, including its motion profile and the temporal dimension, such as velocities. Generally, the movement pattern of all participants shows an initial fast and gross movement and ends with slow and precise movements to adjust for a specific target. This pattern is similarly described by Kendon [18].

3.1 Normalisation

As the position of each rigid body is partly dependent on the specific participants height, is the data normalised by their height (proportionally, along the vectors of their skeleton), as to be able to compare positional data between participants. This transformation does not directly capture other proportional differences between participants, but serves to approximate those as they correlate with height. Additionally, we translate the standing location of the

[2]https://github.com/interactionlab/htc-vive-marker-mount *(accessed: 13.07.21)*

Table 1: The 28 features selected from the pose data, grouped by their data type. Note that, for instance, the positions contain three features; the x, y, and z coordinates.

Group	Feature	Description
Position	Index finger (x,y,z)	Index finger spatial position
	HMD (x,y,z)	HMD spatial position
Combination	Shoulder abduction	Vertical arm movement angle
	Horizontal shoulder flexion	Horizontal arm movement angle
Categorical	Above hand	Indicates whether the index finger is above the hand
	Relative horizontal position	Indicates whether the index finger is above the head, below the shoulder height or in-between
	Relative vertical position	Indicates whether the index finger is to the left of the left shoulder, to the right of the right shoulder or in-between
Distance	Index finger to upper arm, left shoulder and right shoulder	
	Upper arm to left shoulder and right shoulder	
	Hand to upper arm, left shoulder, right shoulder and HMD	
	Forearm to left shoulder, right shoulder, HMD, index finger and upper arm	
	HMD to right shoulder, index finger and upper arm	

participants to a single point. This allows for direct comparison of all position data.

3.2 Feature engineering and selection

We selected features based on three ranking methods: correlation, mutual information, and χ^2-tests, in addition to handpicking features based on regression performance. We use correlation to get an overview of which recorded data is relevant for further investigation (we denote the correlation values with Pearson's r in the following subsections). We did not apply dimensionality reduction techniques such as principal component analysis (PCA) for feature engineering and selection, as initial exploration showed decrease model performance and as interpreting features is not obvious when aggregating in such ways.

3.3 Pose features

The first set of features is based on the position data at the moment of selection, i.e., the final pointing pose of the participant. In addition to analysing position coordinates as such, we also constructed new features from the pose data by using different techniques: combination, binning, and automatic feature engineering. Based on this analysis, we included a set of 28 features derived from the pose data. The resulting feature set is described on Table 1.

3.3.1 Position coordinates. Firstly, we analyse the positional correlations to focus our search on non-redundant features. Our analysis shows that the index finger position correlates strongly with hand (r = 0.99 for x, 0.99 for y, and 0.96 for z) and forearm (r = 0.97 for x, 0.97 for y, and 0.80 for z) positions, which is an expected result because of the kinematic chain between those. Thus, we include only the index finger position among these raw positions into our feature set so as to avoid introducing redundant data.

We observe a strong correlation between the index finger position and the target position in the horizontal (r = 0.95) and the vertical (r = 0.93) dimension. Therefore, we include all coordinates of the index finger position. Of the collected data, we also include the HMD position (r = 0.60 for x, 0.66 for y, and 0.22 for z), because previous work on ray casting has often suggested the importance of head and gaze directions in pointing (e.g., [26, 29, 32]). The orientation coordinates of the rigid bodies show little correlation with the target position and are therefore not added to these position coordinates.

3.3.2 Combinations. We combined right shoulder, upper arm, and forearm positions to compute elbow flexion (r = −0.01 for target x, 0 for target y, and 0 for target z), shoulder abduction (r = −0.11, 0.70, and 0.06), and horizontal shoulder flexion (r = −0.03, 0.01, and −0.01). These flexions were computed by using the collected body measurements of the participants. We explored elbow flexion as a feature under the assumption that the users arm stretch would correlate positively with target depth (e.g., Kendon's work suggested [18]), but found that users tend to stretch their arm fully, no matter the depth. Thus we excluded this feature. The horizontal shoulder flexion is included since it improves regression performance as per our feature tests on those.

3.3.3 Binning. As noted before, the index finger provides information about the target position. To put an emphasis on the horizontal position, we threshold it, such that the horizontal feature indicates whether the index finger is positioned to the left of the body (i.e., beyond the left shoulder), to the right of the body, or in-between (0.95 for target x). Similarly, we threshold the vertical position to indicated whether the index finger is above the participants head, below shoulder height, or in-between (0.71 for target y), and the depth is threshold according to the participants stretched arm (0.18 for target z). This last feature does not improve regression performance and is thus not included. Additionally, we compute a feature that indicates whether the index finger is vertically positioned above the hand (-0.01, 0.72, 0.13), thus revealing whether the participant points up- or downwards. These are therefore categorical features, which can gain weight in regression and boost the classification performance.

3.3.4 Automatic Feature Engineering. We use automatic feature engineering to construct features using a distance operator. This means that we compute distances between all combinations of marker sets, to obtain 21 features. Of these features, the distances between the HMD and the upper arm, forearm, hand, and index finger, show the most overall correlation, with horizontal correlation values ranging between 0.21 and 0.31 and vertical values between -0.64 and -0.71. These distances can describe flexions of joints in pointing poses. The selected distance features are presented in Table 1.

Table 2: The 18 features selected from the motion data, grouped by their data type.

Group	Feature	Description
Polynomial	Index finger (x,y,z)	3rd order polynomial coefficients of the index finger movement, fitted on each dimension
Velocity	Absolute index finger middle	Mean velocity in the time frame between 0.5 and 0.75 seconds after movement start
	Relative index finger start	Mean velocity in the first 20% of samples
	Relative index finger middle	Mean velocity in the time frame between 40% and 60% of samples
	Relative index finger end	Mean velocity in the last 20% of samples
Acceleration	Absolute index finger middle	Mean acceleration in the time frame between .5 and .75 seconds after movement start
	Relative index finger middle	Mean velocity in the time frame between 40% and 60% of samples

3.4 Motion features

Next, we construct a set of features derived from the motion data throughout the pointing movement. The selected set of 18 motion features is described on Table 2.

We plotted the motion profiles (velocity and acceleration) of the raw data, which showed samples up until 15 seconds. We visually recognised a consistent pattern only at the beginning of the movement, and noticed that it aligned with the average duration of pointing (between the start and finish clicks) across all trials. This average duration was 1.8 seconds, and therefore we removed the data beyond that as outliers, which likely resulted from the participants forgetting to trigger selection in some of the trials.

3.4.1 Motion Profile. To analyse the motion profile (i.e., the position of the finger across the frames of motion) we fit a 3rd order polynomial on each positional dimension of the index finger for each trial. For each polynomial, we gather four coefficients, which we use as features. The first three coefficients of the horizontal polynomial correlate well with with horizontal target position (-0.33, 0.65, and -0.38, respectively). Additionally, the first and second coefficients of both the vertical (0.38 and -0.25, respectively) and depth (-0.23 and 0.26, respectively) polynomial correlate with the vertical target position.

3.4.2 Velocity and Acceleration. We also look at velocity and acceleration of the index finger while pointing. We removed the values that were larger than 1.5 times the interquartile range as outliers. To keep the sample size, we interpolated those values by the mean of the surrounding values. To compute the velocity and acceleration, we split the data into bins, and take the mean of these measures (the traveled distance and time and the difference in those) as velocity and acceleration. We compute both absolute and relative bins for the beginning, middle, and end of the movement. The absolute bins are 0-0.3 seconds, 0.5-0.75 seconds, and the final 0.3 seconds of the movement. The relative bins are 0-20%, 40-60%, and 80-100% of the movement. We can generally see that the middle bin has most correlation with the target position, if any. Of these features, the velocity in middle absolute bin carries the most correlation (0.23, 0.27, 0.04). The selected features are presented on Table 2.

4 INTERPRETING A POINTING TARGET

The second purpose for modelling the collected pointing data is to estimate target prediction accuracies. Here, we present the results across three scenarios. These scenarios are inspired by the different ways in which classification and regression may be implemented in practice for target selection techniques.

In the first scenario - pose features - we use only pose data from the final pointing frame (i.e., when the selection is triggered). This is similar to most target selection techniques - taking the position of a *cursor* at the moment of selection.

In the second scenario - motion features - we consider features that could be determined 'en-route' to the target (and so support early prediction), or when it is desirable to have no selection interaction (i.e., no 'click').

Finally, the third scenario - mobile features - uses features that could be gathered through current VR technologies, without the need for an additional tracking setup, namely basic hand, and HMD position.

We test the scenarios on three models. For classification, we test Naive Bayes, Random Forests, and Support-Vector Machines (SVM), and for regression, we use Linear regression, Random Forests, and SVMs. We split the data 80:20 and validate our models using 5-fold cross-validation. We perform a grid search on the training data to find the best hyperparameters for both the classification and regression models, and run validation on the training data. At last the we compute a score based on the test set. These results are presented in Table 3.

4.1 Scenario 1: Pose features

We first tested a set of features based on only the position data from the pose at the moment of selection. The motivation for using only pose data is that most target selection techniques function this way: taking the position of a cursor or a ray from the moment when the selection is triggered, such as with a button click on a mouse. Using the pose-only features is simplest to implement, and does not require buffering multiple frames of data for interpreting pointing targets.

With these pose features we obtain a classification F_1-score of 0.87 using an SVM (which performs well above chance level of 0.04), and a mean prediction distance of 23.56 ± 18.77 cm when using Random Forest regression. Figure 3 depicts the classification results and Figure 4 the regression results across each individual target. Table 3 shows all results of validation and test sets.

4.2 Scenario 2: Motion features

Next, we tested a set of motion-based features from throughout the pointing movement. The motivation for exploring motion features is to enable early prediction of target selection (i.e., whilst the user is still moving towards the target), and to enable continuous target

Table 3: Overview of machine learning model performances across the three different scenarios. Best performing models for each scenario highlighted in bold.

	Classification				Regression			
	Model	*5-fold CV accuracy*	*Test accuracy*	*Test F_1-score*	*Model*	*5-fold CV RMSE*	*Test RMSE*	*Test distance (m)*
Pose features	Naive Bayes	49.00%	39.89%	0.3442	Linear Regression	0.4625	0.4130	0.5253
	Random Forest	74.29%	73.50%	0.7312	**Random Forest**	**0.2649**	**0.2789**	**0.2356**
	SVM	**83.90%**	**87.18%**	**0.8655**	SVM	0.3135	0.3060	0.2788
Motion features	Naive Bayes	18.52%	18.23%	0.1718	Linear Regression	1.0483	1.2084	1.8403
	Random Forest	33.76%	31.05%	0.3031	Random Forest	0.5288	0.5325	0.7237
	SVM	**50.08%**	**50.43%**	**0.4973**	**SVM**	**0.5397**	**0.5210**	**0.6170**
Mobile features	Naive Bayes	45.94%	40.46%	0.3364	Linear Regression	0.4224	0.4191	0.5223
	Random Forest	73.72%	72.65%	0.7232	**Random Forest**	**0.2631**	**0.2860**	**0.2430**
	SVM	**75.57%**	**76.92%**	**0.7609**	SVM	0.3034	0.3082	0.2892

selection without the use of a trigger (i.e., when no 'final pose' can be determined).

With these motion features we obtain a test classification F_1-score of 0.50 and a mean prediction distance of 61.70 ± 34.77 cm when using SVMs (see table 3). This shows that motion features by themselves are inferior to pose features for interpreting pointing targets.

4.3 Scenario 3: Mobile features

The final feature set we test is a subset of the pose features in scenario one. We include features that can be derived from fewer tracking points, including position data only from the HMD, the hand, the index finger, and the lower forearm, thereby excluding all data from the shoulders and the arm. The reason for testing this set is to explore the applicability of the features in interpreting pointing targets with simpler motion-tracking setups. The OptiTrack tracking setup we used consists of multiple cameras and markers, is expensive, wall-mounted, and non-portable, and therefore not applicable for commercial, wireless, and mobile use of virtual or augmented reality headsets. The built-in tracking solutions in many current HMDs, such as Oculus Quest, as well as smaller external tracking devices, such as LeapMotion (which can be easily used with a VR headset), typically only track the hand and, perhaps, the lower forearm, in addition to the headset position.

Table 3 shows that the classification performance suffers from the restricted set of *mobile features*, while the regression performance still predicts the targets with a similar error to the spatial features. We achieve a classification F_1-score of 0.76 and a mean distance of 24.30 ± 19.00 cm. These results are promising for the usage of the mobile feature set to predict target positions with built-in tracking.

5 CHALLENGING TARGETS

The classification and regression performance varies dramatically among the targets. To validate our feature set for more challenging target spaces in virtual environments, we analysed the largest sources of errors among the targets.

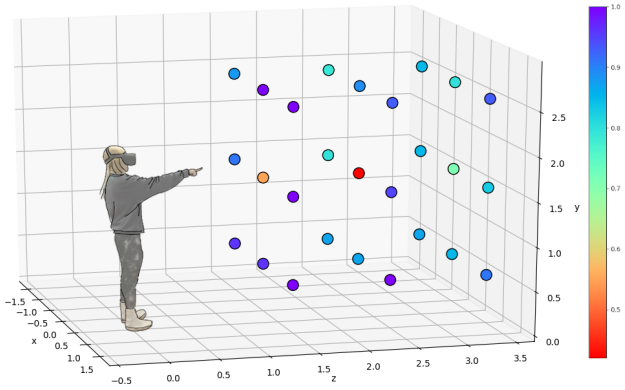


Figure 3: A visualisation of the classification accuracy (F_1-score) using an an SVM trained on a pose-based feature set derived from our study data. The color coding represents the score.

We found the largest errors in the center aisle of targets, that is, the targets at the shoulder height directly in front of the user. This is also visible on Figure 3 for classification, and on Figure 4 for regression. With the pose features (Scenario 1), for instance, 82% of the classification errors come from confusing the targets at different depths but with correctly predicted x and y coordinates. The reason for this is that here the pointing poses "look" very similar. For example, the participants appear to always extend their arm fully when pointing at the distant targets regardless of the differences in their distances. We can generally see little correlation between constructed features and the target depth. Therefore, when there is little variation in x and y coordinates of pointing poses for these particular targets, separating their depth is challenging.

A deeper look into the performance of the SVM with the Pose feature set reveals that the maximum regression error (distance from the actual target) was 152.33 cm for one of the central aisle targets. In fact, the central aisle targets always contained the target with

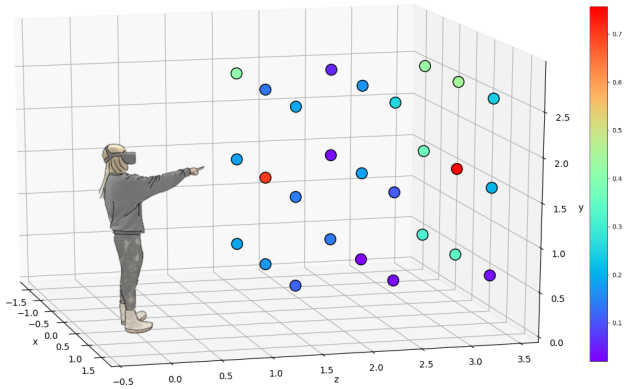


Figure 4: The tested targets, colored by mean distance between predicted and actual targets. The prediction is obtained by training a Random Forest on the pose feature set.

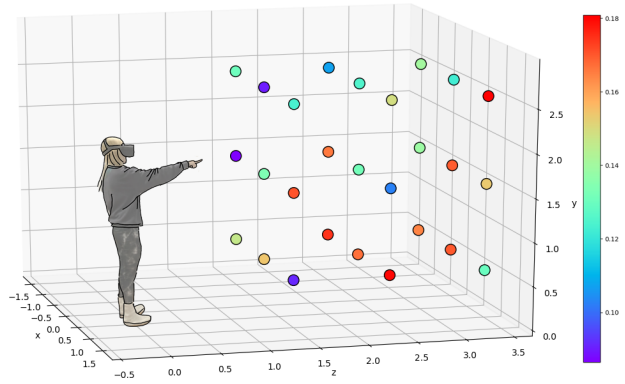


Figure 5: The tested targets, colored by mean distance between predicted and actual targets. The prediction is obtained by training an LSTM network on the collected positional data.

largest error. Therefore, the mean error of 27.88 cm across all targets is a conservative estimate. Because the depth of a target appears hard to estimate with the Pose feature set, we also investigated if that maximum prediction error could be dragged down by adding features. To this end, we combined the Pose and Motion feature sets, and ran the regression tests. We found, that with an SVM the combination of the two features sets dragged the maximum error down to 92.73 cm, and with a Random Forest to 71.36 cm. This reduced the error below the one meter spacing of the targets on the grid. This combination of pose and motion features is not thorough, as it simply combines the existing feature sets. However, it does highlight the promise of engineering new features for such challenging target spaces.

6 BLACK BOX FEATURE SELECTION

In previous sections, we discussed feature selection for classifying and predicting pointing targets. The process found the features that best describe pointing, but was subject to our interpretation and analysis of the data. To uncover opportunities for future work on modeling 3D pointing, we train a black box neural network to learn and solve the regression problem. We construct a long short-term memory (LSTM) neural network [16], that is able to learn from the collected movements.

We train the network, using data collected 0.2 seconds before the participant stopped pointing. This data is chosen as it contains the small adjustments participants made at the end of the pointing motion. Similarly to before, the data is split 80:20 and validated by 5-fold cross-validation. The network yields an average regression error of 13.25 ± 1.10 cm across all targets, a 40.75% decrease compared to the Random Forest regression described before. The network yields a consistent prediction error. Where the Random Forest yielded a maximum average error of 71.36 cm, the network has a maximum average error of 18.09 cm. This is reflected in figure 5, which shows min- and maximum closer together compared to results reported before.

This shows that a complex model has the potential to improve regression results; it can help with informed guesses about the upper boundaries of prediction accuracy. Unfortunately, the network cannot inform design choices of pointing techniques similarly than the classical algorithms described previously do. This is because it is not clear from the internal workings of a neural network or other techniques employing dimensional reduction (e.g., PCA), which features are important or how they interact with each other. Such information can be helpful, for instance, in deciding which body parts to track and which can be left out.

7 DISCUSSION

Pointing-based interaction techniques have become popular in HCI for selecting targets at a distance. Those are seen as simple, intuitive, and well-understood, and have become easy to implement for free hands with recent motion-tracking technologies. These properties lead some to refer to distant pointing as 'natural.' However, outside of technology use, our pointing is a rough, indicative, communication tool [18], that is most often supported by other modalities and lacks spatial accuracy [24]. We argue that the pointing which we use with technology, whether at large displays [34] or in virtual reality, is not natural. To date, distant-pointing as an interaction technique, is most often paired with a cursor. In VR, this cursor is often driven by ray casting, where the cursor is placed on ray based on a vector created by one or two body parts (e.g., the index finger and the elbow). This inherently changes the very way that we point, aligning different body parts to collide the cursor with our chosen targets.

We analysed cursor-free pointing, with a view to informing the design of future cursors for pointing in VR. We sought to reveal the features that best describe user pointing behaviour, by training machine learning techniques to classify and predict the users' intended targets. From only the final pose of pointing, our results show a classification accuracy of 87% among 27 targets, and a mean prediction distance of 23.56 cm for those targets one meter apart, and up to 3.5 meters away. These results show the promise of a machine learning-based approach to facilitating natural pointing in VR. However, our results also reveal a number of further questions.

First, we chose to examine *pose* and *motion* features. Pose features enabled a more direct comparison with existing cursor-based

approaches, where the machine identifies the target at the moment of 'selection'. Motion features enabled us to examine opportunities for forward prediction of targeting gestures. Our pose feature set (F_1-score of 0.87) outperformed our motion feature set (F_1-score of 0.5). In a conservative calculation, pose and motion features in concert performed similarly to pose features alone (an F_1-score of 0.83 and mean distance of 24.70 cm). We believe that pose-and-motion features show promise for future techniques. Additional features could also be drawn from eyes, such as eye dominance as Plaumann et al. [29] suggest, or gaze.

We found the largest prediction errors in the targets on the central row. In our study design, only one target was visible at a time. Should all the spheres have been visible (with the target highlighted), we might expect the pose-and-motion features to reveal further accuracy as participants point 'around' occluding targets. For example, in a real VR scenario we might expect the user to take a small step to the side to reveal the occluded target. This could help disambiguate the targets in the central row directly in front of the participant, where our classification performs the worst. Occluded targets and consequent pointing features remain an avenue for further work.

Second, part of our intended contribution is around revealing the features that best support machine learning for natural pointing. To achieve this, we instrumented our participants with multiple tracking markers along their arms (similar to Mayer et al. [23]). Whilst this allows us to analyse the influence and unique contributions of different joints, it is not indicative of the more limited tracking achievable with current state-of-the-art headsets. The built-in tracking solutions in many current HMDs, such as the Oculus Quest, as well as smaller external tracking devices, such as the LeapMotion, typically only track the hand and lower forearm, in addition to the headset position. With only the tracking facilitated by current commercial technologies, then, we suggest to achieve an F_1-score of 0.76 and a mean error of 24.30 cm.

We are not the first to have considered improving the accuracy of pointing. Mayer et al. [24], however, is one of the few who used cursor-free pointing to develop a model for improving accuracy in ray casting. Prior to applying their model, they report an average ray cast error of 61.3 cm for targets on a 2D plane at a $2 - 3$ m distance, reducing this to 38.4 cm with their model. Over a similar distance, our machine learning approach reduces this error to 23.56 cm, whilst maintaining natural pointing, and working in a target space with an additional dimension: the depth. This demonstrates the promise of our feature-based approach for facilitating natural pointing at a 3D target space in VR.

Third, in our study design the participants performed selections with their non-dominant hand, by pulling the trigger of a controller. Our work can inform design of target selection techniques based on pointing, but does not address how the selection should be triggered. Vogel and Balakrishnan [34], for example, explored how 'clicking' can be done freehand at a distance. However, the act of mid-air clicking simultaneously introduces movement noise into the targeting. Therefore, how to simply indicate selection remains a challenge. One possibility is early prediction similar to our Scenario 2 using features from early motions, and simply confirming the predicted target instead of inducing noise to the final pose.

Pointing movements are considered inaccurate in their nature [12]. We understand little about the noise they may contain. Ours and previous work (e.g., [24, 29]) have found patterns from pointing movements, but leave unclear how much of classification and regression errors are due to inherent noise in movements, and how much due suboptimal set of selected features. In addition to inherent noise in motor control, research shows that using distant-pointing for frequent interactions leads to fatigue [15]. Numerous ways of combatting fatigue have been explored (e.g., [21, 25]). Due to the this and other possible sources of noise, and the unclear extent to which it may occur in natural pointing, we suggest our prediction distance of 23.56 cm with the proposed features is a conservative estimate. Indeed, our black-box test with a neural network showed a prediction distance of 13.25 cm, which can inform about the upper boundaries of the amount of noise.

Further steps could be taken to better pursue pointing in its full natural, communicative form [18]. Primarily, in communication, we point for someone else, not for ourselves - whether indicating something for a friend to look at, or requesting a distant item from the clerk in a shop. Applications for improving communicative pointing have been studied in VR [22]. However, pointing studies, as they appear in HCI, typically disregard this aspect of pointing, though it *would* serve to improve naturalness, and *may* serve to improve target classification. For example, the study design could place a secondary avatar in the scene, and ask participants to communicate a given target to them.

Our study focused on target selection in virtual reality, but the constructed feature sets can also be useful for improved pointing techniques with other user interfaces, such as with augmented reality or smart homes [29]. However, people point differently in VR compared to the real world [23], have more depth perception issues in VR [4], the perception of object locations deviate between virtual and physical surroundings [27], and distance perception of objects directly in front of the user is more accurate than of those on the sides in VR [28]. Therefore, the applicability of the feature sets for user interfaces beyond VR also remains further work.

8 CONCLUSION

Virtual reality allows users to interact with objects in a natural way, similarly to how they do in the physical world. Pointing is a natural way to refer to objects out of the arm's reach. However, without direct touch or the use of cursors, interpreting the user's intended target is challenging. We collected motion data to analyse features of cursor-free pointing, in which users could point as they choose. Based on those, we derive two feature sets and show that the features can be used for predicting pointing targets with promising accuracy. Our feature sets and results can help in designing target selection techniques for virtual reality. These show the feasibility of modeling pointing and open up further investigations into the possibilities of using natural gestures for human-computer interaction.

REFERENCES

[1] Ferran Argelaguet and Carlos Andujar. 2013. A survey of 3D object selection techniques for virtual environments. *Computers & Graphics* 37, 3 (2013), 121–136.
[2] Ferran Argelaguet, Carlos Andujar, and Ramon Trueba. 2008. Overcoming eye-hand visibility mismatch in 3D pointing selection. *Proceedings of the ACM*

Symposium on Virtual Reality Software and Technology, VRST (2008), 43–46. https://doi.org/10.1145/1450579.1450588

[3] Mahdi Azmandian, Mark Hancock, Hrvoje Benko, Eyal Ofek, and Andrew D. Wilson. 2016. Haptic Retargeting: Dynamic Repurposing of Passive Haptics for Enhanced Virtual Reality Experiences. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems* (San Jose, California, USA) *(CHI '16)*. Association for Computing Machinery, New York, NY, USA, 1968–1979. https://doi.org/10.1145/2858036.2858226

[4] Anil Ufuk Batmaz, Mayra Donaji Barrera Machuca, Duc Minh Pham, and Wolfgang Stuerzlinger. 2019. Do head-mounted display stereo deficiencies affect 3d pointing tasks in AR and VR?. In *26th IEEE Conference on Virtual Reality and 3D User Interfaces, VR 2019 - Proceedings*. 585–592. https://doi.org/10.1109/VR.2019.8797975

[5] Joanna Bergström, Tor-Salve Dalsgaard, Jason Alexander, and Kasper Hornbæk. 2021. How to Evaluate Object Selection and Manipulation in VR? Guidelines from 20 Years of Studies. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*. 1–20.

[6] Richard A. Bolt. 1980. "Put-that-there": Voice and gesture at the graphics interface. *Proceedings of the 7th Annual Conference on Computer Graphics and Interactive Techniques, SIGGRAPH 1980* (1980), 262–270. https://doi.org/10.1145/800250.807503

[7] Hélène Cochet and Jacques Vauclair. 2010. Pointing gesture in young children. *Gesture* 10, 2-3 (2010), 129–149. https://doi.org/10.1075/gest.10.2-3.02coc

[8] Andy Cockburn, Philip Quinn, Carl Gutwin, Gonzalo Ramos, and Julian Looser. 2011. Air pointing: Design and evaluation of spatial target acquisition with and without visual feedback. *International Journal of Human-Computer Studies* 69, 6 (2011), 401–414.

[9] Maxime Cordeil, Andrew Cunningham, Tim Dwyer, Bruce H. Thomas, and Kim Marriott. 2017. ImAxes: Immersive Axes as Embodied Affordances for Interactive Multivariate Data Visualisation. In *Proceedings of the 30th Annual ACM Symposium on User Interface Software and Technology* (Québec City, QC, Canada) *(UIST '17)*. Association for Computing Machinery, New York, NY, USA, 71–83. https://doi.org/10.1145/3126594.3126613

[10] Andrea Corradini and Philip R. Cohen. 2002. Multimodal speech-gesture interface for handfree painting on a virtual paper using partial recurrent neural networks as gesture recognizer. *Proceedings of the International Joint Conference on Neural Networks* 3, August 2002 (2002), 2293–2298. https://doi.org/10.1109/IJCNN.2002.1007499

[11] Tiare Feuchtner and Jörg Müller. 2018. Ownershift: Facilitating overhead interaction in virtual reality with an ownership-preserving hand space shift. *UIST 2018 - Proceedings of the 31st Annual ACM Symposium on User Interface Software and Technology* (2018), 31–43. https://doi.org/10.1145/3242587.3242594

[12] J. M. Foley and Richard Held. 1972. Visually directed pointing as a function of target distance, direction, and available cues. *Perception & Psychophysics* 12, 3 (1972), 263–268. https://doi.org/10.3758/BF03207201

[13] Claire C. Gordon, Cynthia L. Blackwell, Bruce Bradtmiller, Joseph L. Parham, Patricia Barrientos, Stephen P. Paquette, Brian D. Corner, Jeremy M. Carson, Joseph C. Venezia, Belva M. Rockwell, Michael Mucher, and Shirley Kristensen. 2014. 2012 Anthropometric Survey of U.S. Army Personnel: Methods and Summary Statistics. *Security* December 2014 (2014), 640.

[14] 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*. 1–14.

[15] Juan David Hincapié-Ramos, Xiang Guo, Paymahn Moghadasian, and Pourang Irani. 2014. Consumed Endurance: A Metric to Quantify Arm Fatigue of Mid-Air Interactions. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (Toronto, Ontario, Canada) *(CHI '14)*. Association for Computing Machinery, New York, NY, USA, 1063–1072. https://doi.org/10.1145/2556288.2557130

[16] Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long Short-Term Memory. *Neural Computation* 9, 8 (1997), 1735–1780. https://doi.org/10.1162/neco.1997.9.8.1735 arXiv:https://doi.org/10.1162/neco.1997.9.8.1735

[17] 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* (Ottawa, Ontario, Canada) *(GI '10)*. Canadian Information Processing Society, CAN, 269–276.

[18] Adam Kendon. 2015. *Gesture: Visible action as utterance*. Cambridge University Press. 1–388 pages.

[19] 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.

[20] Ágnes Melinda Kovács, Tibor Tauzin, Erno Téglás, György Gergely, and Gergely Csibra. 2014. Pointing as epistemic request: 12-month-olds point to receive new information. *Infancy* 19, 6 (2014), 543–557. https://doi.org/10.1111/infa.12060

[21] Mingyu Liu, Mathieu Nancel, and Daniel Vogel. 2015. Gunslinger: Subtle arms-down mid-air interaction. In *Proceedings of the 28th Annual ACM Symposium on User Interface Software & Technology*. 63–71.

[22] Sven Mayer, Jens Reinhardt, Robin Schweigert, Brighten Jelke, Valentin Schwind, Katrin Wolf, and Niels Henze. 2020. Improving Humans' Ability to Interpret Deictic Gestures in Virtual Reality. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*. 1–14.

[23] Sven Mayer, Valentin Schwind, Robin Schweigert, and Niels Henze. 2018. The effect of offset correction and cursor on mid-air Pointing in real and virtual environments. *Conference on Human Factors in Computing Systems - Proceedings* 2018-April (2018). https://doi.org/10.1145/3173574.3174227

[24] Sven Mayer, Katrin Wolf, Stefan Schneegass, and Niels Henze. 2015. Modeling distant pointing for compensating systematic displacements. *Conference on Human Factors in Computing Systems - Proceedings* 2015-April (2015), 4165–4168. https://doi.org/10.1145/2702123.2702332

[25] Roberto A Montano Murillo, Sriram Subramanian, and Diego Martinez Plasencia. 2017. Erg-O: ergonomic optimization of immersive virtual environments. In *Proceedings of the 30th Annual ACM Symposium on User Interface Software and Technology*. 759–771.

[26] Kai Nickel and Rainer Stiefelhagen. 2003. Pointing gesture recognition based on 3D-tracking of face, hands and head orientation. *ICMI'03: Fifth International Conference on Multimodal Interfaces* (2003), 140–146.

[27] Alex Peer and Kevin Ponto. 2019. Mitigating Incorrect Perception of Distance in Virtual Reality through Personalized Rendering Manipulation. In *2019 IEEE Conference on Virtual Reality and 3D User Interfaces (VR)*. IEEE, 244–250.

[28] Etienne Peillard, Thomas Thebaud, Jean-Marie Normand, Ferran Argelaguet, Guillaume Moreau, and Anatole Lécuyer. 2019. Virtual Objects Look Farther on the Sides: The Anisotropy of Distance Perception in Virtual Reality. In *2019 IEEE Conference on Virtual Reality and 3D User Interfaces (VR)*. IEEE, 227–236.

[29] Katrin Plaumann, Matthias Weing, Christian Winkler, Michael Müller, and Enrico Rukzio. 2018. Towards accurate cursorless pointing: the effects of ocular dominance and handedness. *Personal and Ubiquitous Computing* 22, 4 (2018), 633–646. https://doi.org/10.1007/s00779-017-1100-7

[30] Ivan Poupyrev, Mark Billinghurst, Suzanne Weghorst, and Tadao Ichikawa. 1996. The go-go interaction technique. In *UIST (User Interface Software and Technology): Proceedings of the ACM Symposium*. ACM, 79–80. https://doi.org/10.1145/237091.237102

[31] Ben Shneiderman. 1997. Direct manipulation for comprehensible, predictable and controllable user interfaces. In *Proceedings of the 2nd international conference on Intelligent user interfaces*. 33–39.

[32] Ludwig Sidenmark and Hans Gellersen. 2019. Eye, Head and Torso Coordination During Gaze Shifts in Virtual Reality. *ACM Transactions on Computer-Human Interaction (TOCHI)* 27, 1 (2019), 1–40.

[33] J. F. Soechting, S. I. H. Tillery, and M. Flanders. 1990. Transformation from Head- to Shoulder-Centered Representation of Target Direction in Arm Movements. *J. Cognitive Neuroscience* 2, 1 (Jan. 1990), 32–43. https://doi.org/10.1162/jocn.1990.2.1.32

[34] Daniel Vogel and Ravin Balakrishnan. 2005. Distant Freehand Pointing and Clicking on Very Large, High Resolution Displays. In *Proceedings of the 18th Annual ACM Symposium on User Interface Software and Technology* (Seattle, WA, USA) *(UIST '05)*. Association for Computing Machinery, New York, NY, USA, 33–42. https://doi.org/10.1145/1095034.1095041

[35] Stephen Voida, Mark Podlaseck, Rick Kjeldsen, and Claudio Pinhanez. 2005. A Study on the Manipulation of 2D Objects in a Projector/Camera-Based Augmented Reality Environment. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (Portland, Oregon, USA) *(CHI '05)*. Association for Computing Machinery, New York, NY, USA, 611–620. https://doi.org/10.1145/1054972.1055056