RIDS: Implicit Detection of a Selection Gesture Using Hand Motion Dynamics During Freehand Pointing in Virtual Reality

Ting Zhang
tingzhang@fb.com
Reality Labs Research, Meta Inc
Redmond, WA, USA

Zhenhong Hu
zh0711@fb.com
Reality Labs Research, Meta Inc
Redmond, WA, USA

Aakar Gupta
aakarg@fb.com
Reality Labs Research, Meta Inc
Redmond, WA, USA

Chi-Hao Wu
eddywu@fb.com
Reality Labs Research, Meta Inc
Redmond, WA, USA

Hrvoje Benko
benko@fb.com
Reality Labs Research, Meta Inc
Redmond, WA, USA

Tanya R. Jonker
tanya.jonker@fb.com
Reality Labs Research, Meta Inc
Redmond, WA, USA

ABSTRACT

Freehand interactions with augmented and virtual reality are growing in popularity, but they lack reliability and robustness. Implicit behavior from users, such as hand or gaze movements, might provide additional signals to improve the reliability of input. In this paper, the primary goal is to improve the detection of a selection gesture in VR during point-and-click interaction. Thus, we propose and investigate the use of information contained within the hand motion dynamics that precede a selection gesture. We built two models that classified if a user is likely to perform a selection gesture at the current moment in time. We collected data during a pointing-and-selection task from 15 participants and trained two models with different architectures, i.e., a logistic regression classifier was trained using predefined hand motion features and a temporal convolutional network (TCN) classifier was trained using raw hand motion data. Leave-one-subject-out cross-validation PR-AUCs of 0.36 and 0.90 were obtained for each model respectively, demonstrating that the models performed well above chance (=0.13). The TCN model was found to improve the precision of a noisy selection gesture by 11.2% without sacrificing recall performance. An initial analysis of the generalizability of the models demonstrated above-chance performance, suggesting that this approach could be scaled to other interaction tasks in the future.

CCS CONCEPTS

• **Human-centered computing** → **HCI theory, concepts and models**; *Mixed / augmented reality*; *Virtual reality*.

KEYWORDS

selection, pointing, gesture detection, hand motion dynamics, interaction, virtual reality, temporal convolutional network

UIST '22, October 29-November 2, 2022, Bend, OR, USA

ACM ISBN 978-1-4503-9320-1/22/10…$15.00
https://doi.org/10.1145/3526113.3545701

ACM Reference Format:
Ting Zhang, Zhenhong Hu, Aakar Gupta, Chi-Hao Wu, Hrvoje Benko, and Tanya R. Jonker. 2022. RIDS: Implicit Detection of a Selection Gesture Using Hand Motion Dynamics During Freehand Pointing in Virtual Reality. In *The 35th Annual ACM Symposium on User Interface Software and Technology (UIST '22), October 29-November 2, 2022, Bend, OR, USA.* ACM, New York, NY, USA, 12 pages. https://doi.org/10.1145/3526113.3545701

1 INTRODUCTION

To support natural and immersive experiences in augmented reality (AR) or virtual reality (VR), systems have been leveraging freehand interactions, which rely on computer vision or wrist-based sensing technologies [25, 49, 59]. These systems recognize and track mid-air, freehand user input rather than relying on users to perform specific input actions using controllers or joysticks. Although these freehand technologies are intuitive and less encumbered, they introduce new challenges because they require highly robust input recognizers and effortless interaction techniques.

Unfortunately, mid-air gesture detection is relatively unreliable. Vision-based tracking systems tend to fail when occlusion occurs [6], while wrist-based sensing techniques are sensitive to motion artifacts and sensor placements [47]. The use of these techniques also often results in false positive and false negatives during recognition [6, 21, 38, 71], which can significantly impact user experiences [41]. To reduce the false positives that occur when detecting a thumb-finger pinch gesture using a smartwatch, Wen et al. [66] proposed using an activation gesture so that input events would only be detected when the system was activated. This is a non-optimal solution because it places an undue burden on the user to perform additional actions. In this paper, we propose an alternative approach to improve the detection of a selection gesture in VR during point-and-click interaction by harnessing natural user behaviors to *implicitly* infer whether a user intends to make a selection. To this end, we present our approach that does Real-time Implicit Detection of Selections (RIDS). RIDS leverages historical hand motion dynamics during freehand pointing to detect the probability of a user's selection gesture at any time, independently of, and agnostic to, the actual sensing of the gesture (e.g., a finger-thumb pinch) and the selection target. RIDS increases selection accuracy when the sensing of a selection gesture is noisy, which often occurs with wearable systems when freehand gestures are performed. As demonstrated in this work, fusing the output from RIDS with a gesture sensing model increases selection accuracy.

Prior research on implicit input detection has only explored using natural gaze behavior for point-and-select VR tasks [18]. The drawbacks to using gaze are that eye-tracking technology is not integrated within most consumer AR/VR devices and eye-trackers are influenced by variations in scene brightness, eyeglasses, and eye tracker biases [17]. As such, this research explores whether hand motion dynamics during pointing in VR might be useful for improving the detection of a selection gesture.

To this end, data was collected during a pointing and selection task that was representative of a VR game. The hand motion data was then used to build two different RIDS models, one using logistic regression and the other using a temporal convolutional network (TCN). During an evaluation of the models, Leave-one-subject-out cross-validation (LOSOCV) PR-AUC scores of 0.36 and 0.90, were reported (chance = 0.13). Further testing of the model was then conducted using an existing VR pointing dataset to explore the generalizability of the model. Finally, the model was fused with a noisy inertial pinch model and found to increase the selection PR-AUC by 11.19%.

The primary contribution of this research is the development and demonstration of two RIDS models that performed well above chance in a VR setting using users' natural hand motion dynamics. The secondary contributions are an analysis of the models' generalizability to other point-and-select task scenarios and the application of the TCN model to increase the precision of a noisy selection gesture.

2 RELATED WORK

This work builds upon prior research on hand motion dynamics in VR selection and explicit input prediction during pointing.

2.1 Hand Motion Dynamics During Pointing

It is generally assumed that hand motions while pointing consist of predictable patterns that can be modeled [5, 22, 51]. This assumption formed the basis of our motivation to use hand motion dynamics for RIDS. As there are several kinematic parameters from the experimental arm pointing literature that measure predictable patterns over the course of the arm's trajectory (i.e., velocity, peak velocity, time to peak velocity, index of velocity shape etc., [9]), these features were explored in our logistic regression model.

In terms of models, the most well-established model explaining hand motion while pointing is Meyer et al.'s hybrid OII model [50] which separated pointing motions into two distinct stages: a high velocity, large movement to bring the pointer close enough to a target without visual tracking (i.e., the ballistic phase), and a lower velocity, corrective movement to reach a target under feedback control (i.e., the corrective phase). A further assumption noted that this behavior was the result of humans trying to behave optimally according to a certain internalized cost function [22]. Flash and Hogan [23] proposed that this function took the form of a minimum jerk model, where humans aimed to minimize the jerk (i.e., derivative of acceleration) and generate smooth movements, at least during the ballistic phase of movement. Alternatively, other research proposed a minimum acceleration model [10]. Berret et al. argued that vertical arm movements in the air minimized absolute work, the energy consumption of the muscular forces [10]. One implication that results from these models explaining ballistic motion is that as a user gets closer to selecting a target, they would switch from ballistic to corrective motion, possibly resulting in more frequent instances of higher jerk, for example. The variation in these feature values could therefore be informative for RIDS so we explored the use of these features during our logistic regression model development. As described later, we performed feature selection on this set to arrive at the final features that were predictive of selection gestures across individuals.

2.2 Input Prediction During Pointing

There have been several types of input prediction models that have been developed to understand pointing, the most common being end-point prediction [36, 42, 69], hand trajectory prediction [24, 46], and target prediction [14]. Multiple techniques used hand and input device motion for predicting user-intended targets [11, 12, 52, 72]. In human-robot collaboration, existing work used trajectory matching [57] or neural networks [53, 58] to enable proactive robot assistance whenever a user's hand reached for objects. For 3D environments, there has been research predicting vehicular touchscreen input [1–4] and on using long short-term memory (LSTM) models to predict the probability of selecting candidate objects using hand-reach features like position and orientation [16]. A heuristic method was also developed to disambiguate the target object a user intends to grasp in a cluttered scene using hand action cues [56]. Further, existing work has also predicted future cursor positions in target-agnostic ways for mouse input [7, 43, 54], touchscreens [31, 45, 68], and controller input in VR [28, 29]. Gamage et al. [24] demonstrated continuous 3D hand trajectory prediction in VR using a kinematics-based prediction approach.

In addition to hand and input devices, eye tracking has also been used for prediction. For example, gaze scanpaths have been used to predict search targets [13, 60, 61] or anticipate user actions that a robot can perform [34, 35, 40, 63, 70]. Researchers have also explored target forecasting in VR (e.g., [33]), with some research taking advantage of gaze fixations to anticipate a user's hand movements while reaching for objects [15, 26].

Although these techniques have been shown to be useful, each of these projects focused on explicitly predicting the trajectory, the final target, or the final hand or cursor position. The present research focuses on a different problem, namely the implicit detection of a selection gesture in the current moment using contextual information that is independent of the sensing of gestures. The closest research in this space comes from David et al. [19] and Bednarik et al. [18]. Bednarik et al. used an SVM model based on hand-crafted features such as eye fixations and saccades. Bednarik et al's prediction model, however, incorporated gaze data up to one fixation after a click, which reduced its potential application to real-time scenarios. David et al. overcame this limitation but still only used feature-based regression models. In contrast, the present research used hand motion dynamics for a feature-based model, as well as a temporal convolutional network that took raw time-series data as input. It further demonstrated the model's application and investigated its generalizability to another task scenario.

3 DATA COLLECTION

To train and test the RIDS model, hand motion data was collected during a freehand pointing and selection task that invoked movement dynamics analogous to a real-world VR task [62]. While existing work has used controlled, prompt-based pointing tasks [29] for problems such as end-point prediction, those tasks yielded a narrow set of movement dynamics that were not representative of real-world use, making the problem seem easier than it is. Therefore, a VR game that contained target size and distance variations, as well as the unstructured behaviors inherent in real-world use, was developed.

3.1 Task Design

The task was a VR version of Yahtzee [67]. The participant's goal was to compete against the computer to collect as many points as possible within a specific time period (i.e., a three minute block). Each turn started by rolling five dice and the number of turns depended on how fast the participant played the game. A set of actions was displayed on a panel in front of the participant, indicating the points that they could collect if they rolled that combination with their dice (Figure 1). After each roll, the participant could "lock" any subset of the dice to try to aggregate the dice towards a desired combination. Possible actions included rolling the dice, "locking" a die, and selecting a combination to claim points. All actions were selected using a finger-thumb pinch gesture.

Figure 1: A competitive VR Dice Game task.

There were 12 trials within each block and each lasted 3 minutes. Before each block, participants were offered a voluntary break from using the HMD, if desired. After each block, participants completed subjective surveys (these data were not used for model development and were not analyzed within this paper). The entire data collection process lasted approximately one hour.

3.2 Apparatus

The task was built using the Unity game engine. Participants wore an HTC Vive Pro Eye HMD and used a hand tracker puck [32] whose raycast was used for pointing. The tracker's position and orientation provided the 6 DOF hand motion data that was used for the models. To sense the selection pinch gesture, wristwatch-IMU driven pinch sensing similar to Wen et al. [66] was used. The pinch sensing was usable, but not 100% accurate. Although the reported pinch detection had an F1-score of 83%, this score was artificially high because it did not include false positives from non-gestures, where in real world cases, they would be detected frequently. This sensing technique enabled for an investigation of how well the RIDS model could increase the precision of noisy selection gestures. The ground-truth of the pinch selection gesture was also collected using an approach similar to ElectroRing [39], which required the participant to wear rings on their thumb and the index finger proximal phalanges, ensuring near-perfect pinch detection accuracy.

The data from the ground truth rings were not used to drive the selection gesture because the IMU-driven pinch sensing enabled for the collection of data about a participant's hand motion dynamics in the event of false positives and negatives, which the RIDS model needed to account for.

3.3 Participants

For a safe data collection during the COVID-19 pandemic, seventeen right-handed participants were recruited remotely. The devices were mailed to each participant and the study was conducted through video calls. Participants' ages ranged from 23 to 42 with a mean age of 34. Participants included 6 females and 11 males. Two participants reported no experience with VR devices, while the rest had used VR devices in the past. Informed consent was obtained and protocols were approved by the Western Institutional Review Board. Two of the participants' data was removed from the study due to the data being incomplete.

4 MODELING FRAMEWORK

Simple regression models have lower power, processing, and memory costs, which are significant factors for wearable devices, however, a more complex convolutional model may offer better performance despite higher costs. Therefore, two RIDS models were investigated within this research, a logistic regression model and a temporal convolutional network (TCN) model. The two models shared the same modeling framework, producing probabilistic outputs which indicated how likely the participant performed a selection gesture, however they had key differences in terms of model input and architectures.

4.1 Data Processing

The time-series hand motion data was first resampled to 60 Hz to account for irregular data sampling during the real-time recording (i.e., approximately 90 Hz at the standard Unity frame rate). To mark the ground truth of a participant's selection gesture, the RIDS models utilized the onset of the pinch selection signalled by the ground truth pinch device. For each time frame, a class label, i.e., *True/Null*, was added, according to the pinch detection from the ground truth device in each time frame.

The time series hand motion data was then divided into two continuous datasets, with the first 70% of the data being used for training and the remaining 30% of the data being used for held-out testing within each participant. Five-fold cross validation was used on the 70% training data.

4.2 Sliding Window

A sliding window approach was used to enable the model to make an inference at every time frame. The sliding windows had two parameters, i.e., window size and step size. The window size defined the duration of the predictive window used for the model input, while the step size determined how many samples to move forward in time when generating the sliding windows. The models used a step size of 16.67ms (i.e., one data point in a time series with 60Hz sampling rate) and the window size was determined through hyperparameter tuning on the training set. Hyperparameters are typically specified heuristically and then tuned for a given machine learning problem. Tuning allows one to build a model for each combination of hyperparameter values and select the best hyperparameter value based on the one that provides the best performance on the validation set. Window sizes ranging from 83.33 to 2500.00 milliseconds were investigated. The class label for each window was determined by the class of the last sample in the window. All windows with a pinch gesture detected in the middle of the sample were discarded to ensure that the model considered only a single instance of a true pinch in the data. As training samples were generated through a sliding window over the time series data, cross-validation could not be performed using a randomized sampling strategy to ensure that there were no overlapping window segments between the training and testing sets nor the cross-validation folds. To maintain data independence, we first split the time-series into continuous data folds (i.e., Fold 1 from time 0 to t, Fold 2 from time t+1 to 2t+1..),and then applied the sliding window to each fold.

4.3 Model Evaluation Metric

Both the area under the curve of the Receiver Operator Characteristic (ROC-AUC) and the Precision Recall Curve (PR-AUC) were used to evaluate the models. ROC-AUC is a more commonly used metric in evaluating a model's performance, however, compared to ROC-AUCs assuming a chance rate of 0.5, PR-AUCs are more appropriate for unbalanced datasets [20, 64]. The PR-AUC metric is more sensitive to a large number of null classes that are misclassified as false positives and the chance rate of PR-AUC is derived from the percentage of positive examples among all samples, which varies by individual. To facilitate direct comparisons between participants and models, a standardized chance rate of 0.13 was created for each participant by resampling the data to a fixed ratio of 1:7 between positive and null classes. A much higher chance rate was used in this research to address the data unbalance problem for model training. In original data, the ratio between the number of true and null classes is 1:128.18 (chance rate = 0.0078). This would have had an impact on our real-time evaluation since the actual ratio would be heavily skewed in favor of null samples. Essentially, the resampling emphasized the detection of true positives more than true negatives. Section 6.3 further evaluated the model's performance using an adjusted metric which considered real-time application requirements. Although models evaluated on resampled data do not represent their real-time performance, such data enabled for model selection by facilitating parameter tuning and model comparisons. In contrast, the additional parameters used in the adjusted metric (Section 6.3) made the training space intractable.

5 FEATURE-BASED LOGISTIC REGRESSION MODEL

To implicitly detect selection gestures, a logistic regression model was developed using a set of hand motion features (Figure 2a). A recursive feature addition (RFA) approach, which is commonly used to select features that have meaningful independent contributions towards predicting the target value [27, 55], was used as part of the feature exploration pipeline (Figure 2b).

5.1 Feature Extraction

Based on the pointing model literature [5, 9, 51], 19 features were extracted from the triaxial hand positions and forward pointing direction vectors which the pointing raycast was also generated with (Table 1). The features included *hand position* and *forward pointing direction velocity*, *acceleration*, *jerk*, *time since peak velocity*, *time since peak acceleration*, *velocity shape*, *acceleration shape*, *absolute work*, and *hand position direction change* in *velocity* and *jerk*.

A Pearson correlation matrix was computed to examine whether these extracted 19 features were correlated. Given the similarity amongst some of the features, it was expected that collinearity would exist within the data, which is a problem for feature selection because highly correlated features could be selected interchangeably. As six features including *hand position velocity shape*, *forward pointing direction velocity shape*, *time since peak velocity of hand position*, *time since peak velocity of forward pointing direction*, *time since peak acceleration of hand position*, and *time since peak acceleration of forward pointing direction* were highly correlated (correlation coefficient $r > 0.58$; Figure 2c), *velocity shape of hand position* was selected to represent this set of correlated features. Other features were excluded as they were more dependent on absolute time, making them challenging to generalize across tasks.

5.2 Feature Selection

A recursive feature addition (RFA) approach was used to select features from the remaining set of possible features. A sliding window was used to extract input samples for model development. Because different window sizes impact results differently, this RFA process was repeated seventeen times for different window sizes ranging from 83.33 to 2500.00 milliseconds. The selected features reported below were generated using the window size of the highest PR-AUC (i.e., window size = 2166.67 milliseconds).

5.2.1 Model Description. Sklearn (version 1.0.1) was used for the logistic regression models. Due to the imbalanced dataset, the training parameter, *class weight*, was set to be inversely proportional to the number of samples for each class. Thus, a class with fewer samples was penalized more when it was wrongly classified.

5.2.2 Recursive Feature Addition (RFA). RFA was first performed for each participant (Figure 2b). Features were added using a randomized order to ensure the best features were selected irrespective of the order they were added. Features were retained if they increased the average PR-AUC across folds; otherwise, they were dropped. The features were then rank ordered by the percentage of participants that retained a given feature (Figure 3a). The resulting feature order served as the input order for the next-step recursive feature selection using the training data from all participants to

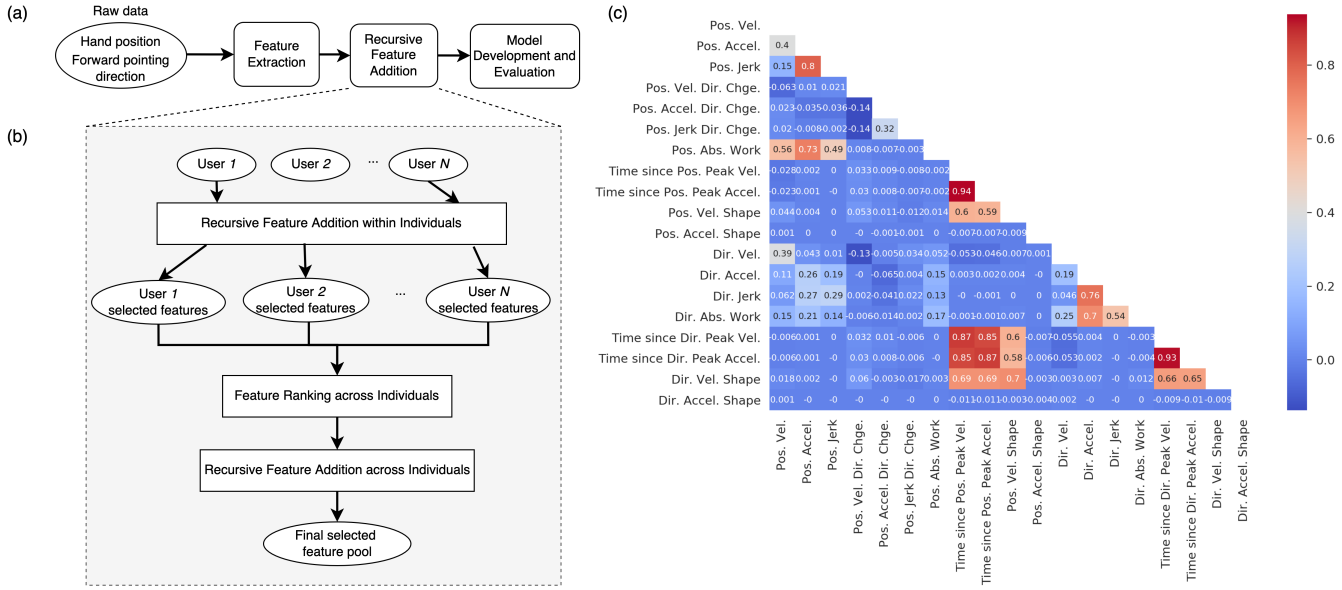

Figure 2: Feature-based logistic regression model development. (a) The feature exploration pipeline. Hand features were first extracted from the raw hand motion time series data. To identify a set of predictive features, training samples were generated using a sliding window approach followed by an RFA method. A logistic regression model leveraged this set of selected features. (b) A flowchart of the RFA method. (c) The feature correlation matrix.

ensure that noise did not eliminate a good feature and to remove features that were not consistently predictive across participants. Similarly, features that increased the PR-AUC were retained, otherwise they were dropped (Figure 3b). The final set of selected features are highlighted in Table 1.

While the velocity and acceleration features are self-explanatory, the other selected features may be less intuitive. *Absolute work*, for example, accounts for the assumption that a participant's goal was to minimize the work performed while pointing [10]. While kinematic models use joint torques to estimate this, the experimental setup did not afford access to such information so an approximation was used. The other features, including *velocity shape* and *direction change*, capture the relative dynamics of the hand motions. *Velocity shape*, for example, is a standard kinematic parameter for modeling arm movements [9] that tries to approximate a single value for the shape of a velocity curve as the hand approaches a target. Assuming a perfect ballistic-corrective motion, *velocity shape* will be at its highest at the time of selection. *Direction change features*, on the other hand, are intended to capture how the corrective phase includes adjustments around a target when a participant changed their motion direction multiple times to correct for overshooting or undershooting. Lastly, although different participants might have had different speed-accuracy trade-offs during pointing such that absolute features including *velocity*, *acceleration* and *absolute work* were not as generalizable as the selected relative features, there were consistent temporal patterns among them (e.g., participants slowed their hand motions down to a near-zero velocity or acceleration just before a selection).

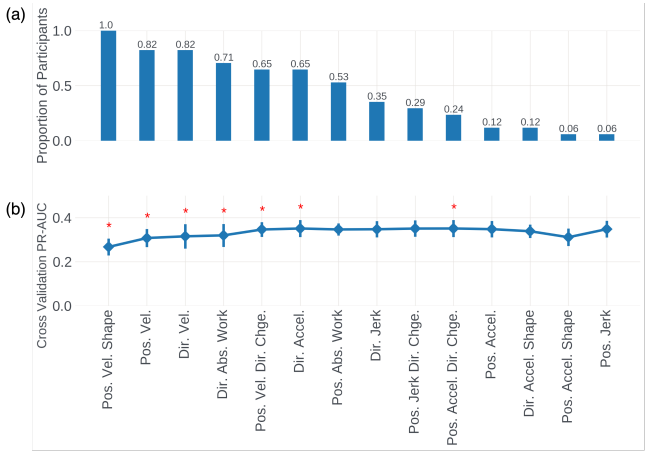

Figure 3: Results of the RFA. (a) The proportion of participants that retained a given feature. (b) Features that were retained from the RFA within participants were iteratively added, from the most retained to the least retained. Each point depicts the average PR-AUC from the 5-fold cross-validation. The error bars depict 95% confidence intervals. Asterisks correspond with features that increased the PR-AUC relative to the previous benchmark and were used in follow-up model evaluations.

5.3 Model Evaluation Results

From the RFA analysis, a set of 7 hand motion features that could implicitly detect selections during freehand pointing in VR was

Table 1: A description of the extracted features. The shaded rows denote the final features after feature selection.

Feature Name	Description	Mathematical Formula
Pos. Vel.	hand position velocity v	$v = \frac{\sqrt{(x_1-x_0)^2+(y_1-y_0)^2+(z_1-z_0)^2}}{\Delta t}$, where $\Delta t = t_1 - t_0$
Pos. Accel.	hand position acceleration a	$a = \frac{v_1-v_0}{\Delta t}$
Pos. Jerk	hand position jerk j	$j = \frac{a_1-a_0}{\Delta t}$
Pos. Vel. Dir. Chge.	velocity direction change $\Delta\alpha$	$\Delta v = \frac{\arccos[(\vec{p}_1\cdot\vec{p}_0)/(\lvert\vec{p}_1\rvert\lvert\vec{p}_0\rvert)]}{\Delta t}$
Pos. Accel. Dir. Chge.	acceleration direction change Δa	$\Delta a = \frac{\arccos[(\vec{v}_1\cdot\vec{v}_0)/(\lvert\vec{v}_1\rvert\lvert\vec{v}_0\rvert)]}{\Delta t}$
Pos. Jerk Dir. Chge.	jerk direction change Δj	$\Delta j = \frac{\arccos[(\vec{a}_1\cdot\vec{a}_0)/(\lvert\vec{a}_1\rvert\lvert\vec{a}_0\rvert)]}{\Delta t}$
Pos. Abs. Work	absolute work of hand position	$w = a * \lvert\vec{p}_1 - \vec{p}_0\rvert$, $\vec{p}_t$ is the position vector at time t
Time since Pos. Peak Vel.	time since peak hand position velocity t_{v_p}, reset the start time to 0 after each pinch selection event	$t_{v_p} = \begin{cases} 0 & v \le v_p \\ t - t_{v_p} & v_p \le v \end{cases}$, v_p is the rolling peak velocity
Time since Pos. Peak Accel.	time since peak hand position acceleration t_{a_p}, reset the start time to 0 after each pinch selection event	$t_{a_p} = \begin{cases} 0 & a \le a_p \\ t - t_{a_p} & a_p \le a \end{cases}$, a_p is the rolling peak acceleration
Pos. Vel. Shape	position velocity shape vs	$vs = \frac{v_p}{v_m}$, v_p, v_m is the rolling peak and mean velocity
Pos. Accel. Shape	position acceleration shape as	$as = \frac{a_p}{a_m}$, a_p, a_m is the rolling peak and mean acceleration
Dir. Vel.	hand pointing direction velocity v	$v = \frac{\arccos\left[(\vec{d}_1\cdot\vec{d}_0)/(\left\lvert\vec{d}_1\right\rvert\left\lvert\vec{d}_0\right\rvert)\right]}{\Delta t}$, $\vec{d}_t$ is pointing direction vector
Dir. Accel.	hand pointing direction acceleration a	$a = \frac{v_1-v_0}{\Delta t}$
Dir. Jerk	hand pointing direction jerk j	$j = \frac{a_1-a_0}{\Delta t}$
Dir. Abs. Work	pointing direction absolute work	$w = a * \lvert\vec{p}_1 - \vec{p}_0\rvert$, $\vec{p}_t$ is the position vector at time t
Time since Dir. Peak Vel.	time since peak velocity of pointing direction t_{v_p}, reset the start time to 0 after each pinch selection event	$t_{v_p} = \begin{cases} 0 & v \le v_p \\ t - t_{v_p} & v_p \le v \end{cases}$, v_p is the rolling peak velocity
Time since Dir. Peak Accel.	time since peak acceleration of pointing direction t_{a_p}, reset the start time to 0 after each pinch selection event	$t_{a_p} = \begin{cases} 0 & a \le a_p \\ t - t_{a_p} & a_p \le a \end{cases}$, a_p is the rolling peak acceleration
Dir. Vel. Shape	pointing direction velocity shape vs	$vs = \frac{v_p}{v_m}$, v_p, v_m is the rolling peak and mean velocity
Dir. Accel. Shape	pointing direction acceleration shape as	$as = \frac{a_p}{a_m}$, a_p, a_m is the rolling peak and mean acceleration

obtained. To further evaluate model performance using these features, the effect of different sliding window sizes and the model's generalizability across participants was evaluated. This evaluation demonstrated that the model performance grew as the window size increased from 83.33 to 2500 milliseconds (Figure 4). The best PR-AUC of 0.37 was achieved with a 2166.67 millisecond window.

A different type of cross-validation, Leave-one-subject-out cross-validation (LOSOCV), was also performed to evaluate the model's generalizability across participants. Figure 5 represents the model performance trained on 14 users' data and tested on the left out participant. An average ROC-AUC of 0.79 and PR-AUC of 0.36 were found to be 57.60% and 184% higher than chance, respectively. All participants' ROC-AUC and PR-AUC scores were also found to be higher than chance. This indicates that there is a set of hand motion features that can be used to implicitly detect selection gestures during freehand pointing in VR.

The above results also suggest that there is space for improvement. More complex model architectures are commonly explored when it comes to performance improvements with a simple model. In the next section, we present improved model performance by utilizing a Temporal Convolutional Network (TCN).

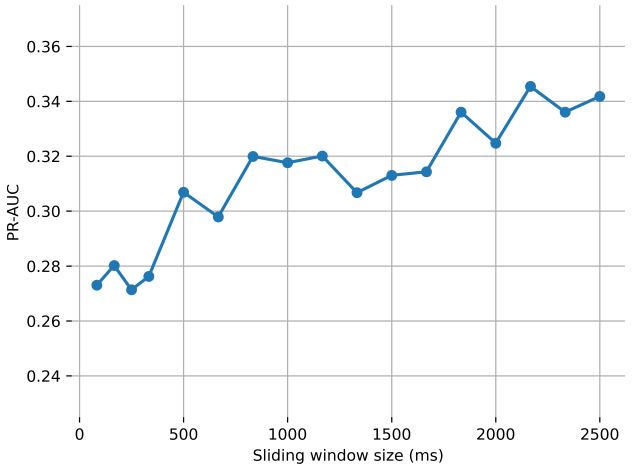

Figure 4: The effect of sliding window size on the Logistic Regression-based RIDS model.

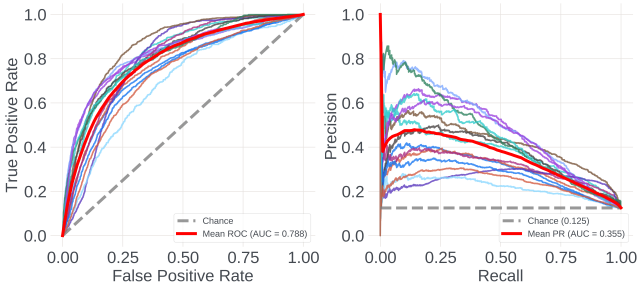

Figure 5: The logistic regression model LOSOCV performance. The ROC and PR curves are depicted on the left and right panels, respectively. Each colored curve represents each participants' performance. The thick red curve represents the averaged curve across participants.

6 TCN MODEL

To further improve the implicit detection of participants' selection gestures using hand motion dynamics, a temporal convolutional network (TCN) was also developed.

6.1 Model Description

Prior work has shown that TCNs can achieve good performance when learning spatio-temporal patterns from long term series data [44, 48, 65]. TCN model architectures are comprised of causal and dilated convolutional layers. The causal convolutions ensure that there is no leakage of information from the future to the past. The dilated convolutions help the model to learn longer historical information while maintaining a relatively simple architecture. This is an important property for the present research as the dataset might not be large enough such that a much more complex but also promising architecture (i.e., a FCN or a ResNet) might easily overfit it [37]. The TCN model implemented within this research followed the architecture defined in Bai et al. [8] with 4 layers.

Different from the logistic regression model, the TCN model used raw hand motion data as input rather than engineered features. Compared to linear models, a convolutional network has a higher capability to learn meaningful patterns from raw data with different convolution kernels. In this work, the model input was formulated as the first-order difference between two consecutive time frames of the hand position and forward pointing direction vectors.

6.2 Results

Similar to the logistic regression model evaluation, the TCN's model performance was investigated using different sliding window sizes and generalizability across participants.

The performance of the TCN model was first evaluated with different window sizes using the training data from all participants. The PR-AUC increased from 0.86 to 0.92 when the window size increased from 83.33 to 333.33 milliseconds and then continued fluctuating around 0.92 afterwards (Figure 6). The best PR-AUC, i.e., 0.93, was obtained using a 1333.33 millisecond sliding window, which was 644% higher than the chance rate of 0.13. This result demonstrates how a shorter duration of hand motion dynamics (i.e., 1333.33 milliseconds) could effectively infer selection gestures using the TCN model.

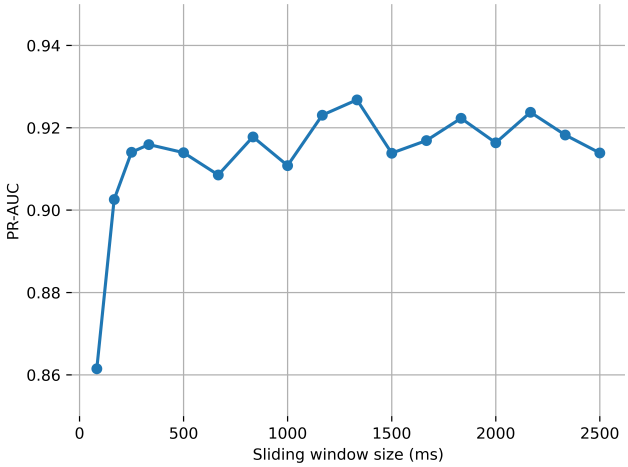

Figure 6: The effect of the sliding window size on the TCN model.

To investigate its generalizability across participants, the TCN model performance was then evaluated using Leave-one-subject-out cross-validation (LOSOCV; Figure 7). The averaged ROC-AUC of 0.97 and PR-AUC of 0.90 were 94% and 617.6% higher than chance, respectively. Each of the folds' performance was also higher than chance, with ROC-AUCs ranging from 0.91 to 0.99 and PR-AUCs ranging from 0.69 to 0.99. Compared to the feature-based logistic regression model performance (Figure 5), TCN improved the ROC-AUC by 23.10% and the PR-AUC by 152.68%. This indicates that the TCN model captured motion dynamics that were hidden in the raw data, whereas the engineered features used in the logistic regression model were not able to.

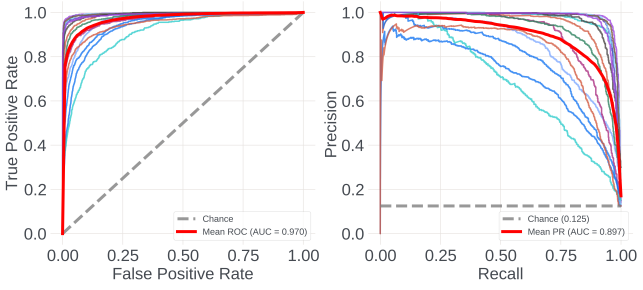

Figure 7: The TCN model LOSOCV performance. The ROC and PR curves are depicted on the left and right panel, respectively. Each colored curve represents each participants' performance. The thick red curve represents the averaged curve across participants.

6.3 Evaluation with an Adjusted Metric for Real-time Applications

As mentioned in 4.3, to facilitate model optimization and promote the occurrence of true samples during training, the data was resampled into a 1:7 ratio between the number of true and null samples (original ratio = 1:128.18). To investigate how this model might perform in a real-time application using a sliding window, a different metric was needed because overlapping windows over penalize the model's precision. When a sliding window was used for real-time applications without resampling, the overlapping windows were expected to be predicted as similar values, which over counted the number of false positives for windows that were overlapping with the true samples. This metric scanned the predicted time series using a debounce window and identified the predictions as true positive, false positives, or false negatives by searching within a time duration around the positive events and matching them with the ground truth. Debounce windows are a common technique used in real-time gesture classification algorithms [39] and they were also used in the pinch detector algorithm herein.

The two parameters, the *debounce window* and the *search length*, could be adjusted according to the requirements of the application. The *debounce window* was needed to reduce the frequency of signal onset due to overlapping windows. When debounce was applied to a time series, the first prediction that was above the onset threshold might not be perfectly synchronized with the ground truth. Therefore, a synchronization step was needed to find the matching events for evaluation. This was realized through a search around any onset event with a *search length*. Although the evaluation was not performed with the model deployed in real-time, this metric not only hinted at how the model might work in real-time, but also supported the exploration of different application requirements by adjusting the *debounce window*.

A debounce window of 300 milliseconds with a search length of 150 millisenconds was used to evaluate the TCN RIDS model. When evaluating the onset threshold from 0.1 to 1.0, the best performance was achieved with a precision of 0.58 and a recall of 0.70 when the threshold was set to 0.63. Although the real-time evaluation does not show near-ceiling performance, the RIDS model is expected to work well when used with noisy input recognizers to improve their performance (see Section 7.2). Also note that existing research on the implicit detection of selections did not evaluate real-time performance [18, 19].

7 ADDITIONAL EXPLORATIONS

Besides the development and evaluation of the two RIDS models, preliminary explorations were also performed to understand the models' generalizablity across other point-and-select task scenarios and to improve noisy input sensing models.

7.1 Model Generalizability Across Task Scenarios

Although the models might be representative of the specific task that they were trained for, the learnt behavior prior to the selection events should be generalizable to a certain extent to other pointing and selection tasks in VR. To explore the model's generalizability to other point-and-select task scenarios, the trained model was applied to an existing dataset from a reciprocal pointing task [30] in VR, wherein participants were prompted to point back and forth, in succession, between start and end targets that were rendered as spheres. The task consisted of controlled variations of the target angle, depth, and circular position of the spheres. The task used an Oculus Rift headset and handheld controllers for pointing. Not only was the task scenario different from this present research, but the selection events were also triggered through button presses on the controller instead of pinch gestures. The dataset consisted of 6 DOF motion data (i.e., triaxial controller positions and forward direction vectors) that was collected from the Oculus Rift controller. For more details on the task, refer to Henrikson et al.'s paper [30].

To ensure that fair comparisons were performed, for each task, the first 70% of the data for each participant was used as training data and the remaining 30% was used as testing data. Given that the reciprocal pointing task was far more constrained than the Dice Game task, we expected that training on the Dice Game task and testing on the reciprocal pointing task would be more effective than the reverse, because training on the Dice Game task could capture more variance in hand motions than the reciprocal pointing task.

Figure 8 shows the performance of the logistic regression model while Figure 9 shows the TCN model's performance. Given the reciprocal task simplicity, it is not surprising that both models have high PR-AUCs of .99 when trained and tested on this task. Also, as expected, models trained using the reciprocal pointing task produced above-chance performance on the Dice Game task, but the PR-AUCs were relatively poor (0.15 for logistic regression and 0.26 for TCN). This was not surprising as the reciprocal pointing task contained much simpler behaviors than the Dice Game task, such that the model might fail when interpreting such unseen patterns. The application of the Dice Game model to the reciprocal pointing task, on the other hand, was able to achieve a PR-AUC of 0.47 ($chance = 0.13$) using both models. This result, which is 276% higher than chance, indicates that the model captured task independent hand motion dynamics which were useful while detecting selection events on a novel task that the model had not been trained on. It is likely that these results could be improved further using a transfer learning approach to facilitate quick applications of this model to other novel tasks or by training the model on multiple distinct pointing-and-selection interaction tasks.

Comparing the two models, the TCN model had a much better performance than the logistic regression model when trained on the simple reciprocal task and tested on the more complex Dice Game task (i.e., 0.15 vs 0.26). This also indicates that when training data has less variance, TCN can capture more generalizable hidden features compared to engineered features.

7.2 Application: Improving Accuracy of a Noisy Pinch

The Real-time Implicit Detection of Selections (RIDS) could be used to increase selection accuracy when the sensing of a selection gesture is noisy. Since the collected data was comprised of the IMU-driven pinch detection and the pinch ground truth, it was possible to investigate how TCN RIDS could be used to improve the accuracy of the IMU-driven pinch detection.

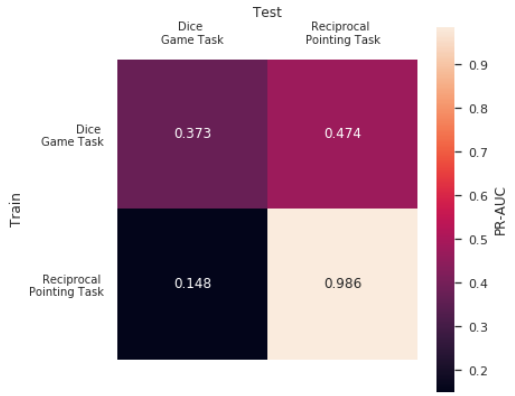

Figure 8: The results of the Logistic regression model generalizability cross-testing.

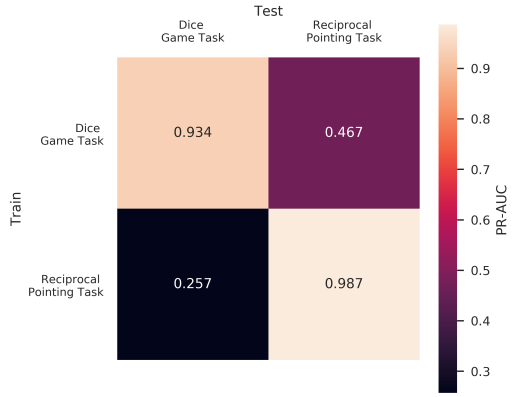

Figure 9: The results of the TCN model generalizability cross-testing.

Using the real-time evaluation metric described in Section 6.3, the IMU-driven pinch detection precision and recall were found to be 0.62 and 0.79, respectively. This evaluation contained pinches that did not trigger any UI events, thus participants might have experienced a more reliable pinch recognizer while interacting.

The IMU-driven pinch probabilities were then integrated with the probabilities generated by RIDS using a weighted sum algorithm (Eq. 1), where $0 < w < 1$.

$$P(selection) = w * P(RIDS) + (1 - w) * P(\text{pinch recognizer}) \quad (1)$$

Weight values, w, from 0.1 to 0.9 were used to calculate the improvement at every value. As studies have shown that false positives have more impact on user experience of a input recognizer than false negatives [41], this exploration focused on improving the precision performance of the pinch recognizer. At $w = 0.34$, the output demonstrated maximum performance, improving the selection precision by 11.19% (from 0.62 to 0.69) without sacrificing recall performance (from 0.79 to 0.80). This preliminary investigation shows that RIDS can be used to increase selection accuracies of noisy selection gestures.

8 DISCUSSION AND FUTURE WORK

Two models leveraging dynamic hand motions were built to implicitly detect a user's selection gesture: 1) a logistic regression model using engineered hand motion features, and 2) a TCN model using raw hand motion data. Through the development of the logistic regression model, we identified a consistent set of hand motion features across participants that were predictive of participant selection gestures. While both models perform well above chance, the TCN model demonstrated improved performance and task generalizability over the logistic regression model by learning discriminating hand motion dynamics using a much shorter duration of relative hand movements.

8.1 Limitations of the Experimental Design

In contrast to typical controlled tasks that prompt participants to select the highlighted target and therefore may not have realistic hand motion dynamics, this research used an unstructured task to represent real world hand motion dynamics. However, the models were still conditioned on the Dice Game task dynamics as seen in the generalizability results for the reciprocal pointing task. Sensing hardware is another factor that could affect the direct application of the models. For example, hand motions could be captured through headset cameras instead of the tracker puck used in this research. Although fatigue effects were minimal because each trial included a 3 minute break, the weight and mounting of the tracker puck might have affected participants' hand motions. The sensing capability of the hardware, such as low resolution cameras and IMUs with drift issues, may also impact the captured raw data as well as the extracted features.

The models developed in this research might not translate directly to other tasks due to the above factors, but the framework for building these RIDS models can be reproduced on other hardware and tasks.

8.2 Other Potential Applications

This research demonstrated a 11.2% improvement in pinch detection precision when a noisy pinch recognizer was combined with the TCN RIDS model. While this research focused primarily on the problem of pointing and selection, RIDS could be useful for other interactions that involve the use of noisy discrete gestures, such as scrolling, sliding, or typing. Another application area for such models could be to discriminate between multiple discrete gestures. For instance, if the user can pinch, double-pinch, or do a thumb-middle finger pinch at the end of their motion trajectory, detection accuracies may be even lower, in which case, a real-time discrimination model leveraging implicit behavioral patterns may be useful. Finally, given the high power and processing costs of gesture sensing algorithms for wearable devices, RIDS could act as a gatekeeper such that gesture sensing algorithms lay dormant until RIDS determines that a user might be trying to interact.

8.3 Early Fusion vs. Late Fusion

While the probabilities from the RIDS model were integrated with those from the IMU-driven model, an alternate approach could be to use both data streams to train a single classifier, thus performing an early fusion of the two streams and deploying the resulting stream.

The trade-off here though, is that while a pinch detection algorithm could be task- and context-agnostic, hand motion dynamics may vary and a single early fused model may not work for all tasks. Late fusion thus provides deployment flexibility and additional accuracy only when it is trained for and needed.

8.4 Detection vs. Prediction

RIDS focuses on the implicit detection of selection gestures in the current moment, which is a narrower slice of the larger problem of predicting when and what users will select after pointing. This narrower focus enables for the exploration of specific problems that occur at the moment of selection, pertaining to the improvement of noisy gesture detection, and can be extended to other applications such as gesture discrimination and gatekeeping. In contrast to this narrowly-scoped detection model, an interesting future direction would be the input anticipation problem in advance of user selection. This is a more complex problem than detection in the moment, but different from end-point or target prediction.

9 CONCLUSION

To improve input and interaction for mid-air, freehand technologies leveraging implicit user behavior, this research presented two models for the Real-time Implicit Detection of Selections (RIDS) for freehand pointing in virtual reality. The models used hand motion dynamics during pointing to classify selection events. The generalizability of the model to another task scenario was also investigated. This research also reported on an application of the models, where outputs were fused with a noisy pinch sensing system and improved the sensing system's accuracy.

As virtual and augmented reality devices become more popular, they will require more complex input techniques that do not encumber the hands and can be used anywhere. Such input techniques will be driven by wearable sensing modules that will not be as certain in their input detection as a mouse or touchscreen. In such scenarios, RIDS can offer a useful and independent signal of selection input that can be deployed flexibly depending on a system's needs. This research outlines the initial steps in what it believed to be an promising future for AR/VR input and interaction.

ACKNOWLEDGMENTS

The authors would like to thank Vivien Francis for the development of the data collection environment, along with Taylor Bunge, Chip Connor, and Nour Shoora for implementing the remote data collection infrastructure.

REFERENCES

[1] Bashar I. Ahmad, Patrick M. Langdon, Simon J. Godsill, Richard Donkor, Rebecca Wilde, and Lee Skrypchuk. 2016. You Do Not Have to Touch to Select: A Study on Predictive In-Car Touchscreen with Mid-Air Selection. In *Proceedings of the 8th International Conference on Automotive User Interfaces and Interactive Vehicular Applications* (Ann Arbor, MI, USA) *(Automotive'UI 16)*. Association for Computing Machinery, New York, NY, USA, 113–120. https://doi.org/10.1145/3003715.3005461

[2] Bashar I. Ahmad, Patrick M. Langdon, Simon J. Godsill, Robert Hardy, Eduardo Dias, and Lee Skrypchuk. 2014. Interactive Displays in Vehicles: Improving Usability with a Pointing Gesture Tracker and Bayesian Intent Predictors. In *Proceedings of the 6th International Conference on Automotive User Interfaces and Interactive Vehicular Applications* (Seattle, WA, USA) *(AutomotiveUI '14)*. Association for Computing Machinery, New York, NY, USA, 1–8. https://doi.org/10.1145/2667317.2667413

[3] Bashar I Ahmad, James Kevin Murphy, Simon Godsill, Patrick M Langdon, and Robery Hardy. 2017. Intelligent interactive displays in vehicles with intent prediction: A Bayesian framework. *IEEE Signal Processing Magazine* 34, 2 (2017), 82–94. https://doi.org/10.1109/MSP.2016.2638699

[4] Bashar I Ahmad, James K Murphy, Patrick M Langdon, Simon J Godsill, Robert Hardy, and Lee Skrypchuk. 2015. Intent inference for hand pointing gesture-based interactions in vehicles. *IEEE transactions on cybernetics* 46, 4 (2015), 878–889. https://doi.org/10.1109/TCYB.2015.2417053

[5] Stanislav Aranovskiy, Rosane Ushirobira, Denis Efimov, and Géry Casiez. 2020. A switched dynamic model for pointing tasks with a computer mouse. *Asian Journal of Control* 22, 4 (2020), 1387–1400. https://doi.org/10.1002/asjc.2063 arXiv:https://onlinelibrary.wiley.com/doi/pdf/10.1002/asjc.2063

[6] Kazuyuki Arimatsu and Hideki Mori. 2020. *Evaluation of Machine Learning Techniques for Hand Pose Estimation on Handheld Device with Proximity Sensor*. Association for Computing Machinery, New York, NY, USA, 1–13. https://doi.org/10.1145/3313831.3376712

[7] Takeshi Asano, Ehud Sharlin, Yoshifumi Kitamura, Kazuki Takashima, and Fumio Kishino. 2005. Predictive Interaction Using the Delphian Desktop. In *Proceedings of the 18th Annual ACM Symposium on User Interface Software and Technology* (Seattle, WA, USA) *(UIST '05)*. Association for Computing Machinery, New York, NY, USA, 133–141. https://doi.org/10.1145/1095034.1095058

[8] Shaojie Bai, J. Zico Kolter, and Vladlen Koltun. 2018. An Empirical Evaluation of Generic Convolutional and Recurrent Networks for Sequence Modeling. *CoRR* abs/1803.01271 (2018). arXiv:1803.01271 http://arxiv.org/abs/1803.01271

[9] Bastien Berret, Enrico Chiovetto, Francesco Nori, and Thierry Pozzo. 2011. Evidence for composite cost functions in arm movement planning: an inverse optimal control approach. *PLoS computational biology* 7, 10 (2011), e1002183.

[10] Bastien Berret, Christian Darlot, Frédéric Jean, Thierry Pozzo, Charalambos Papaxanthis, and Jean Paul Gauthier. 2008. The inactivation principle: mathematical solutions minimizing the absolute work and biological implications for the planning of arm movements. *PLoS computational biology* 4, 10 (2008), e1000194.

[11] Xiaojun Bi and Shumin Zhai. 2013. Bayesian Touch: A Statistical Criterion of Target Selection with Finger Touch. In *Proceedings of the 26th Annual ACM Symposium on User Interface Software and Technology* (St. Andrews, Scotland, United Kingdom) *(UIST '13)*. Association for Computing Machinery, New York, NY, USA, 51–60. https://doi.org/10.1145/2501988.2502058

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

[13] Ali Borji, Andreas Lennartz, and Marc Pomplun. 2015. What do eyes reveal about the mind?: Algorithmic inference of search targets from fixations. *Neurocomputing* 149 (2015), 788–799. https://doi.org/10.1016/j.neucom.2014.07.055

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

[15] Lung-Pan Cheng, Eyal Ofek, Christian Holz, Hrvoje Benko, and Andrew D. Wilson. 2017. Sparse Haptic Proxy: Touch Feedback in Virtual Environments Using a General Passive Prop. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems* (Denver, Colorado, USA) *(CHI '17)*. Association for Computing Machinery, New York, NY, USA, 3718–3728. https://doi.org/10.1145/3025453.3025753

[16] Aldrich Clarence, Jarrod Knibbe, Maxime Cordeil, and Michael Wybrow. 2021. Unscripted Retargeting: Reach Prediction for Haptic Retargeting in Virtual Reality. In *2021 IEEE Virtual Reality and 3D User Interfaces (VR)*. IEEE, 150–159. https://doi.org/10.1109/VR50410.2021.00036

[17] Viviane Clay, Peter König, and Sabine Koenig. 2019. Eye tracking in virtual reality. *Journal of eye movement research* 12, 1 (2019).

[18] Brendan David-John, Candace Peacock, Ting Zhang, T. Scott Murdison, Hrvoje Benko, and Tanya R. Jonker. 2021. Towards Gaze-Based Prediction of the Intent to Interact in Virtual Reality. In *ACM Symposium on Eye Tracking Research and Applications* (Virtual Event, Germany) *(ETRA '21 Short Papers)*. Association for Computing Machinery, New York, NY, USA, Article 2, 7 pages. https://doi.org/10.1145/3448018.3458008

[19] Brendan David-John, Candace Peacock, Ting Zhang, T. Scott Murdison, Hrvoje Benko, and Tanya R. Jonker. 2021. Towards Gaze-Based Prediction of the Intent to Interact in Virtual Reality. In *ACM Symposium on Eye Tracking Research and Applications* (Virtual Event, Germany) *(ETRA '21 Short Papers)*. Association for Computing Machinery, New York, NY, USA, Article 2, 7 pages. https://doi.org/10.1145/3448018.3458008

[20] Jesse Davis and Mark Goadrich. 2006. The Relationship between Precision-Recall and ROC Curves. In *Proceedings of the 23rd International Conference on Machine Learning* (Pittsburgh, Pennsylvania, USA) *(ICML '06)*. Association for Computing Machinery, New York, NY, USA, 233–240. https://doi.org/10.1145/1143844.1143874

[21] Artem Dementyev and Joseph A. Paradiso. 2014. WristFlex: Low-Power Gesture Input with Wrist-Worn Pressure Sensors. In *Proceedings of the 27th Annual ACM Symposium on User Interface Software and Technology* (Honolulu, Hawaii, USA) *(UIST '14)*. Association for Computing Machinery, New York, NY, USA, 161–166. https://doi.org/10.1145/2642918.2647396

[22] Florian Fischer, Arthur Fleig, Markus Klar, Lars Grüne, and Jörg Müller. 2020. An Optimal Control Model of Mouse Pointing Using the LQR. *arXiv preprint arXiv:2002.11596* (2020).

[23] Tamar Flash and Neville Hogan. 1985. The coordination of arm movements: an experimentally confirmed mathematical model. *Journal of neuroscience* 5, 7 (1985), 1688–1703.

[24] Nisal Menuka Gamage, Deepana Ishtaweera, Martin Weigel, and Anusha Withana. 2021. So Predictable! Continuous 3D Hand Trajectory Prediction in Virtual Reality. In *The 34th Annual ACM Symposium on User Interface Software and Technology* (Virtual Event, USA) *(UIST '21)*. Association for Computing Machinery, New York, NY, USA, 332–343. https://doi.org/10.1145/3472749.3474753

[25] Jun Gong, Aakar Gupta, and Hrvoje Benko. 2020. *Acustico: Surface Tap Detection and Localization Using Wrist-Based Acoustic TDOA Sensing*. Association for Computing Machinery, New York, NY, USA, 406–419. https://doi.org/10.1145/3379337.3415901

[26] Eric J. Gonzalez, Parastoo Abtahi, and Sean Follmer. 2020. REACH+: Extending the Reachability of Encountered-Type Haptics Devices through Dynamic Redirection in VR. In *Proceedings of the 33rd Annual ACM Symposium on User Interface Software and Technology* (Virtual Event, USA) *(UIST '20)*. Association for Computing Machinery, New York, NY, USA, 236–248. https://doi.org/10.1145/3379337.3415870

[27] Henry Heberle. 2019. *Computational methods in Biology: cancer biomarkers, protein networks and lateral gene transfer*. Doctoral Dissertation. University of São Paulo.

[28] Rorik Henrikson, Daniel Clarke, Thomas White, Frances Lai, Michael Glueck, Stephanie Santosa, Daniel Wigdor, Tovi Grossman, Sean Trowbridge, and Hrvoje Benko. 2020. Head-Coupled Kinematic Template Matching for Target Selection in Hangry Piggos. In *Extended Abstracts of the 2020 CHI Conference on Human Factors in Computing Systems* (Honolulu, HI, USA) *(CHI EA '20)*. Association for Computing Machinery, New York, NY, USA, 1–4. https://doi.org/10.1145/3334480.3383176

[29] 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*. Association for Computing Machinery, New York, NY, USA, 1–14. https://doi.org/10.1145/3313831.3376489

[30] 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*. Association for Computing Machinery, New York, NY, USA, 1–14. https://doi.org/10.1145/3313831.3376489

[31] Niels Henze, Markus Funk, and Alireza Sahami Shirazi. 2016. Software-Reduced Touchscreen Latency. In *Proceedings of the 18th International Conference on Human-Computer Interaction with Mobile Devices and Services* (Florence, Italy) *(MobileHCI '16)*. Association for Computing Machinery, New York, NY, USA, 434–441. https://doi.org/10.1145/2935334.2935381

[32] HTC. 2021. Vive Trcker. *https://www.vive.com/us/accessory/tracker3/* (2021).

[33] Zhiming Hu, Andreas Bulling, Sheng Li, and Guoping Wang. 2021. Fixationnet: Forecasting eye fixations in task-oriented virtual environments. *IEEE Transactions on Visualization and Computer Graphics* 27, 5 (2021), 2681–2690. https://doi.org/10.1109/TVCG.2021.3067779

[34] Chien-Ming Huang, Sean Andrist, Allison Sauppé, and Bilge Mutlu. 2015. Using gaze patterns to predict task intent in collaboration. *Frontiers in psychology* 6 (2015), 1049. https://doi.org/10.3389/fpsyg.2015.01049

[35] Chien-Ming Huang and Bilge Mutlu. 2016. Anticipatory robot control for efficient human-robot collaboration. In *2016 11th ACM/IEEE international conference on human-robot interaction (HRI)*. IEEE, 83–90. https://doi.org/10.1109/HRI.2016.7451737

[36] Netha Hussain, Katharina S Sunnerhagen, and Margit Alt Murphy. 2019. Endpoint kinematics using virtual reality explaining upper limb impairment and activity capacity in stroke. *Journal of neuroengineering and rehabilitation* 16, 1 (2019), 1–9.

[37] Hassan Ismail Fawaz, Germain Forestier, Jonathan Weber, Lhassane Idoumghar, and Pierre-Alain Muller. 2019. Deep learning for time series classification: a review. *Data mining and knowledge discovery* 33, 4 (2019), 917–963.

[38] Peter Ju, Leslie Pack Kaelbling, and Yoram Singer. 2000. State-based Classification of Finger Gestures from Electromyographic Signals. In *ICML*.

[39] Wolf Kienzle, Eric Whitmire, Chris Rittaler, and Hrvoje Benko. 2021. ElectroRing: Subtle Pinch and Touch Detection with a Ring. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems* (Yokohama, Japan) *(CHI '21)*. Association for Computing Machinery, New York, NY, USA, Article 3, 12 pages. https://doi.org/10.1145/3411764.3445094

[40] Fatemeh Koochaki and Laleh Najafizadeh. 2018. Predicting intention through eye gaze patterns. In *2018 IEEE Biomedical Circuits and Systems Conference (BioCAS)*. IEEE, 1–4. https://doi.org/10.1109/BIOCAS.2018.8584665

[41] Ben Lafreniere, Tanya R. Jonker, Stephanie Santosa, Mark Parent, Michael Glueck, Tovi Grossman, Hrvoje Benko, and Daniel Wigdor. 2021. False Positives vs. False Negatives: The Effects of Recovery Time and Cognitive Costs on Input Error Preference. In *The 34th Annual ACM Symposium on User Interface Software and Technology* (Virtual Event, USA) *(UIST '21)*. Association for Computing Machinery, New York, NY, USA, 54–68. https://doi.org/10.1145/3472749.3474735

[42] Edward Lank, Yi-Chun Nikko Cheng, and Jaime Ruiz. 2007. Endpoint Prediction Using Motion Kinematics. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (San Jose, California, USA) *(CHI '07)*. Association for Computing Machinery, New York, NY, USA, 637–646. https://doi.org/10.1145/1240624.1240724

[43] Edward Lank, Yi-Chun Nikko Cheng, and Jaime Ruiz. 2007. Endpoint Prediction Using Motion Kinematics. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (San Jose, California, USA) *(CHI '07)*. Association for Computing Machinery, New York, NY, USA, 637–646. https://doi.org/10.1145/1240624.1240724

[44] Pedro Lara-Benítez, Manuel Carranza-García, José M. Luna-Romera, and José C. Riquelme. 2020. Temporal Convolutional Networks Applied to Energy-Related Time Series Forecasting. *Applied Sciences* 10, 7 (2020). https://doi.org/10.3390/app10072322

[45] Huy Viet Le, Valentin Schwind, Philipp Göttlich, and Niels Henze. 2017. PredicTouch: A System to Reduce Touchscreen Latency Using Neural Networks and Inertial Measurement Units. In *Proceedings of the 2017 ACM International Conference on Interactive Surfaces and Spaces* (Brighton, United Kingdom) *(ISS '17)*. Association for Computing Machinery, New York, NY, USA, 230–239. https://doi.org/10.1145/3132272.3134138

[46] Flavien Lebrun, Sinan Haliyo, and Gilles Bailly. 2021. A Trajectory Model for Desktop-Scale Hand Redirection in Virtual Reality. In *IFIP Conference on Human-Computer Interaction*. Springer, 105–124.

[47] Yang LI, Jin HUANG, Feng TIAN, Hong-An WANG, and Guo-Zhong DAI. 2019. Gesture interaction in virtual reality. *Virtual Reality & Intelligent Hardware* 1, 1 (2019), 84–112. https://doi.org/10.3724/SP.J.2096-5796.2018.0006

[48] Yujie Liu, Hongbin Dong, Xingmei Wang, and Shuang Han. 2019. Time Series Prediction Based on Temporal Convolutional Network. In *2019 IEEE/ACIS 18th International Conference on Computer and Information Science (ICIS)*. 300–305. https://doi.org/10.1109/ICIS46139.2019.8940265

[49] Manuel Meier, Paul Streli, Andreas Fender, and Christian Holz. 2021. TapID: Rapid Touch Interaction in Virtual Reality using Wearable Sensing. In *2021 IEEE Virtual Reality and 3D User Interfaces (VR)*. 519–528. https://doi.org/10.1109/VR50410.2021.00076

[50] David E Meyer, Richard A Abrams, Sylvan Kornblum, Charles E Wright, and JE Keith Smith. 1988. Optimality in human motor performance: ideal control of rapid aimed movements. *Psychological review* 95, 3 (1988), 340.

[51] Jörg Müller, Antti Oulasvirta, and Roderick Murray-Smith. 2017. Control Theoretic Models of Pointing. *ACM Trans. Comput.-Hum. Interact.* 24, 4, Article 27 (aug 2017), 36 pages. https://doi.org/10.1145/3121431

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

[53] Davide Nicolis, Andrea Maria Zanchettin, and Paolo Rocco. 2018. Human Intention Estimation based on Neural Networks for Enhanced Collaboration with Robots. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. 1326–1333. https://doi.org/10.1109/IROS.2018.8594415

[54] Phillip T. Pasqual and Jacob O. Wobbrock. 2014. Mouse Pointing Endpoint Prediction Using Kinematic Template Matching. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (Toronto, Ontario, Canada) *(CHI '14)*. Association for Computing Machinery, New York, NY, USA, 743–752. https://doi.org/10.1145/2556288.2557406

[55] Candace E Peacock, Ting Zhang, Brendan David-John, T Scott Murdison, Matthew J Boring, Hrvoje Benko, and Tanya R Jonker. 2022. Gaze dynamics are sensitive to target orienting for working memory encoding in virtual reality. *Journal of vision* 22, 1 (2022), 2–2.

[56] Frol Periverzov and Horea Ilieş. 2015. IDS: The intent driven selection method for natural user interfaces. In *2015 IEEE Symposium on 3D User Interfaces (3DUI)*. 121–128. https://doi.org/10.1109/3DUI.2015.7131736

[57] Claudia Pérez-D'Arpino and Julie A. Shah. 2015. Fast target prediction of human reaching motion for cooperative human-robot manipulation tasks using time series classification. In *2015 IEEE International Conference on Robotics and Automation (ICRA)*. 6175–6182. https://doi.org/10.1109/ICRA.2015.7140066

[58] Harish Chaandar Ravichandar and Ashwin P. Dani. 2017. Human Intention Inference Using Expectation-Maximization Algorithm With Online Model Learning. *IEEE Transactions on Automation Science and Engineering* 14, 2 (2017), 855–868. https://doi.org/10.1109/TASE.2016.2624279

[59] Farshid Salemi Parizi, Wolf Kienzle, Eric Whitmire, Aakar Gupta, and Hrvoje Benko. 2021. RotoWrist: Continuous Infrared Wrist Angle Tracking Using a Wristband. In *Proceedings of the 27th ACM Symposium on Virtual Reality Software and Technology* (Osaka, Japan) *(VRST '21)*. Association for Computing Machinery, New York, NY, USA, Article 26, 11 pages. https://doi.org/10.1145/3489849.3489886

[60] Hosnieh Sattar, Mario Fritz, and Andreas Bulling. 2020. Deep gaze pooling: Inferring and visually decoding search intents from human gaze fixations. *Neurocomputing* 387 (2020), 369–382. https://doi.org/10.1016/j.neucom.2020.01.028

[61] Hosnieh Sattar, Sabine Muller, Mario Fritz, and Andreas Bulling. 2015. Prediction of search targets from fixations in open-world settings. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 981–990. https://doi.org/10.1109/CVPR.2015.7298700

[62] Naveen Sendhilnathan, Ting Zhang, Ben Lafreniere, Tovi Grossman, and Tanya R. Jonker. 2022. Detecting Input Recognition Errors and User Errors using Gaze Dynamics in Virtual Reality. In *Proceedings of the 35th Annual ACM Symposium on User Interface Software and Technology*. https://doi.org/10.1145/3526113.3545628

[63] Ronal Singh, Tim Miller, Joshua Newn, Liz Sonenberg, Eduardo Velloso, and Frank Vetere. 2018. Combining Planning with Gaze for Online Human Intention Recognition. In *Proceedings of the 17th International Conference on Autonomous Agents and MultiAgent Systems* (Stockholm, Sweden) *(AAMAS '18)*. International Foundation for Autonomous Agents and Multiagent Systems, Richland, SC, 488–496.

[64] Nesime Tatbul, Tae Jun Lee, Stan Zdonik, Mejbah Alam, and Justin Gottschlich. 2018. Precision and Recall for Time Series. In *Advances in Neural Information Processing Systems*, S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett (Eds.), Vol. 31. Curran Associates, Inc. https://proceedings.neurips.cc/paper/2018/file/8f468c873a32bb0619eaeb2050ba45d1-Paper.pdf

[65] Renzhuo Wan, Shuping Mei, Jun Wang, Min Liu, and Fan Yang. 2019. Multivariate Temporal Convolutional Network: A Deep Neural Networks Approach for Multivariate Time Series Forecasting. *Electronics* 8, 8 (2019). https://doi.org/10.3390/electronics8080876

[66] Hongyi Wen, Julian Ramos Rojas, and Anind K. Dey. 2016. Serendipity: Finger Gesture Recognition Using an Off-the-Shelf Smartwatch. 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, 3847–3851. https://doi.org/10.1145/2858036.2858466

[67] Wikipedia. 2022. Yahtzee. *https://en.wikipedia.org/wiki/Yahtzee* (2022).

[68] Haijun Xia, Ricardo Jota, Benjamin McCanny, Zhe Yu, Clifton Forlines, Karan Singh, and Daniel Wigdor. 2014. Zero-Latency Tapping: Using Hover Information to Predict Touch Locations and Eliminate Touchdown Latency. In *Proceedings of the 27th Annual ACM Symposium on User Interface Software and Technology* (Honolulu, Hawaii, USA) *(UIST '14)*. Association for Computing Machinery, New York, NY, USA, 205–214. https://doi.org/10.1145/2642918.2647348

[69] Difeng Yu, Hai-Ning Liang, Xueshi Lu, Kaixuan Fan, and Barrett Ens. 2019. Modeling Endpoint Distribution of Pointing Selection Tasks in Virtual Reality Environments. *ACM Trans. Graph.* 38, 6, Article 218 (Nov. 2019), 13 pages. https://doi.org/10.1145/3355089.3356544

[70] Gregory Zelinsky, Zhibo Yang, Lihan Huang, Yupei Chen, Seoyoung Ahn, Zijun Wei, Hossein Adeli, Dimitris Samaras, and Minh Hoai. 2019. Benchmarking gaze prediction for categorical visual search. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*. 0–0. https://doi.org/10.1109/CVPRW.2019.00111

[71] Yang Zhang and Chris Harrison. 2015. Tomo: Wearable, Low-Cost Electrical Impedance Tomography for Hand Gesture Recognition. *Proceedings of the 28th Annual ACM Symposium on User Interface Software & Technology* (2015).

[72] Brian Ziebart, Anind Dey, and J. Andrew Bagnell. 2012. Probabilistic Pointing Target Prediction via Inverse Optimal Control. In *Proceedings of the 2012 ACM International Conference on Intelligent User Interfaces* (Lisbon, Portugal) *(IUI '12)*. Association for Computing Machinery, New York, NY, USA, 1–10. https://doi.org/10.1145/2166966.2166968