# Probabilistic In-Plane Detection for Mid-Air Virtual Surface Interactions


Florian Fischer
Department of Engineering
University of Cambridge
Cambridge, United Kingdom
fjf33@cam.ac.uk

Zhuorui Yu
Department of Engineering
University of Cambridge
Cambridge, United Kingdom
zy353@cam.ac.uk

Christoph A. Johns
Department of Computer Science
Aarhus University
Aarhus, Denmark
cajohns@cs.au.dk

Per Ola Kristensson
Department of Engineering
University of Cambridge
Cambridge, United Kingdom
pok21@cam.ac.uk

John J Dudley
Department of Engineering
University of Cambridge
Cambridge, United Kingdom
jjd50@cam.ac.uk

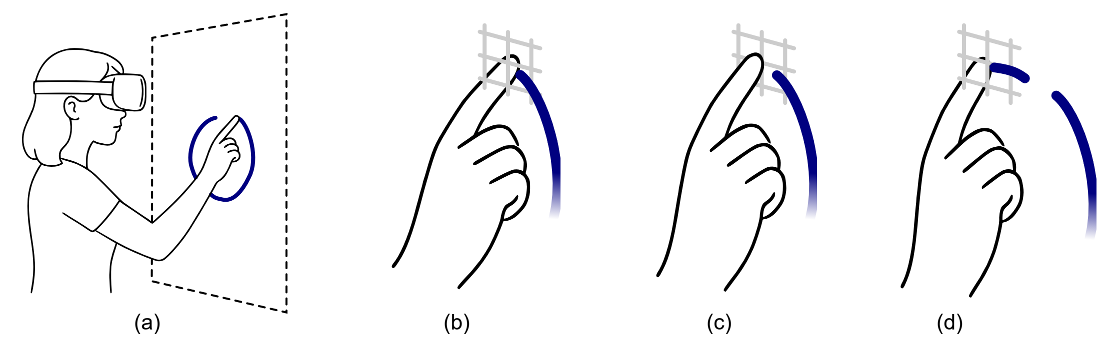




**Figure 1: In this paper we examine the challenge of supporting users with interactions performed on a virtual mid-air plane. In (a), the user is attempting to draw a circle on a 2D virtual plane. A conventional approach to supporting such interactions is to begin drawing when the user's fingertip enters the plane, and to terminate drawing when the user's fingertip leaves the plane. Provided the user's finger remained behind the plane, as in (b), the line will be continuous. If, however, the user inadvertently leaves the plane, as in (c), the line will stop drawing. The user may correct their depth and re-enter the plane, but under a conventional implementation there will now be a gap in the drawn line as shown in (d).**


## Abstract

In virtual and augmented reality, certain types of mid-air interaction, such as planar drawing or dragging interface elements, are most sensibly performed on a 2D surface. When this surface must be located in mid-air and without any form of passive haptic feedback, it can be very difficult to accurately determine the start and end of user input due to various sources of uncertainty. The conventional approach is to apply a simple depth threshold on the interaction plane and associate entry within this threshold as the start event and exit from this threshold as the end event. However, users struggle to maintain planar movement without any physical support, and unintended departures from the threshold are common, resulting in a frustrating user experience. In this work, we explore an approach that handles the exit events probabilistically so that the user interface can gracefully switch between operating modes, and avoid frustrating interruptions of user interactions. Our method is simple, interpretable, and can be easily deployed on commercial VR hardware. In a user study, we demonstrate that it reduces the number of unintended exits by a factor of 16 in a drag-and-drop task and to almost zero in slider and sketching tasks, compared to simple depth-based thresholding. Participants also found our method more usable and less demanding.



## CCS Concepts

• **Human-centered computing** → **Virtual reality**; **Interaction techniques**; **Empirical studies in interaction design**.

## Keywords

Virtual reality, virtual surface, mid-air interaction.

**ACM Reference Format:**
Florian Fischer, Zhuorui Yu, Christoph A. Johns, Per Ola Kristensson, and John J Dudley. 2025. Probabilistic In-Plane Detection for Mid-Air Virtual Surface Interactions. In *ACM Symposium on Spatial User Interaction (SUI '25), November 10–11, 2025, Montreal, QC, Canada.* ACM, New York, NY, USA, 11 pages. https://doi.org/10.1145/3694907.3765937




*SUI '25, Montreal, QC, Canada*



ACM ISBN 979-8-4007-1259-3/25/11
https://doi.org/10.1145/3694907.3765937

# 1 Introduction

Virtual reality (VR) and augmented reality (AR) do not only set new standards on how we perceive information from our environment, but also open up new possibilities for interacting with physical and virtual objects. Building on inside-out body tracking implemented in consumer-level head-mounted displays (HMDs) such as the Meta Quest or the Apple Vision Pro, controller-free input methods have gained much traction, as they enable more direct and natural interaction [37, 47]. However, using natural gestures and body motion to select and manipulate objects, or to switch between game modes, requires particular caution.

In a world where every movement could be interpreted as an attempt to create or modify virtual content, every bodily movement, whether deliberate or not, takes on new significance. This necessitates robust and reliable discrimination between intentional and erroneous body movements. Mode switching leaves it up to the user to signal intent to interact, usually through dominant- or non-dominant hand gestures such as pinch or push [24, 27, 56]. While such explicit delimiters considerably improve accuracy of user intent classification, they require additional time and effort from the user, and increase interaction complexity. For direct mid-air interactions with a virtual 2D surface, we suggest and demonstrate that this complexity can be dispensed with through simple and lightweight discrimination of intent based on the fingertip motion.

Building on the line of user intent recognition methods for hand gesture input [43], this work examines the problem of probabilistically detecting the end of mid-air surface interactions. In contrast to existing works that leverage contextual information from additional modalities such as speech, facial expressions, or eye gaze dynamics [11, 29, 45, 55, 57], we hypothesized that there are discriminative features in the user's 3D fingertip trace that can be exploited to infer when the user wishes to leave the interaction plane. To test this hypothesis and support development of the user intent inference capability, we collected a dataset ($n$=8) of hand-tracking motion data captured while users interact with three variants of mid-air surface interactions. These variants are: (i) drag and drop; (ii) manipulating a slider; and (iii) 2D sketching. The interaction for these tasks was based on a simple depth-based thresholding such that departure of the fingertip from the plane would be treated as a leave event.

We reviewed and labeled this dataset to build an understanding for the features in the fingertip trace indicative of intent to end planar interaction. We identified two core features indicative of the user's intent to withdraw their finger from the interaction plane: (i) the fingertip velocity along the axis perpendicular to the plane; and (ii) the maximum distance between the fingertip and plane. These features were then used as the basis for a Naïve Bayes classifier, with model parameters determined using a partition of the collected dataset. This classifier achieved an accuracy rate of 93.6% on the held-out partition of the dataset.

To assess whether a probabilistic treatment of the withdrawal event can enhance planar VR interaction in practice, we conducted an evaluation study ($n$=16). This study compared interaction with and without the intent classifier across the same three mid-air interaction tasks. We observed a significant reduction in undesired terminations of the planar interaction in all three tasks, with very few undesired terminations remaining in the slider and sketching task. Moreover, participants found planar interaction with the classifier more usable and experienced reduced perceived workload. Nevertheless, for high-precision tasks such as for high-resolution slider manipulation, small delays introduced by the classifier may negatively impact user experience.

In summary, this work demonstrates the untapped potential that simple probabilistic methods combined with (noisy) consumer-level motion tracking offer to free-hand planar interaction in VR. It makes the following contributions:

(1) Development of a method for probabilistically classifying intentional and unintentional departures from the plane for 2D virtual surface interactions in VR.
(2) Empirical evaluation of this method and demonstration of a significantly lower number of interaction errors and improved usability over the conventional depth-based thresholding approach.

# 2 Related Work

In this work, we investigate the potential of a Naïve Bayes classifier for enhancing free-hand planar interaction in VR. We therefore review use cases of planar interaction in commercial products and user interaction research, as well as related work using probabilistic classification to facilitate VR interaction.

## 2.1 Planar Interactions in VR

While VR and other immersive environments open up new opportunities for natural and direct interaction in 3D [61], traditional 2D surfaces and user interface elements still play an important role. Widgets, sliders, 2D windows and buttons, and info boxes are an integral part of modern VR interfaces. XR operating systems such as Meta Horizon OS or Apple's visionOS are fundamentally based on the same WIMP (windows, icons, menus, pointer) paradigm that was originally proposed for desktop interfaces and later transferred to mobile devices and ubiquitous computing [4, 61]. Creating an account, customizing one's avatar, or adjusting system settings all involve a series of menu selections, checkbox ticks, and slider manipulations. In addition, applications traditionally rooted in desktop or mobile contexts, such as web browsers, document editors, chat platforms, and video or image editing tools, are increasingly finding their way into VR. While VR interfaces build upon function-specific widgets that can be flexibly arranged and combined in 3D space [4, 59], these widgets still share the form of 2D 'planes' in otherwise 3D space. It is therefore surprising that direct interaction with mid-air planes, despite being an essential part of immersive user interfaces, has received relatively little attention in VR/AR research.

While planar interaction in VR shares many similarities with mobile touch devices or public displays [53, 61], the lack of haptic feedback, especially in free-hand interaction [8, 54], results in some unique characteristics. Crucially, the lack of tactile or force feedback when 'touching' a virtual object has been shown to negatively affect users' depth perception [16, 28, 36]. This can lead to an increased movement variability along the depth axis, a lower spatial awareness in VR, and reduced performance and accuracy rates in virtual pointing or typing tasks [8, 12, 13]. In addition, the lack of physical touch further complicates the delimitation problem. Without methods that can robustly detect switches between interaction modes (e.g., writing on a touchscreen versus hovering over the screen), interaction is prone to becoming unreliable [50] with negative impacts for efficiency and user experience.

The delimitation problem has been investigated for interfaces relying on body gestures [25, 41], pen-based input [27], and mid-air sketching [6]. Proposed mitigation strategies include ad-hoc approaches such as increasing the depth threshold that defines the 'input zone' in which user input is processed [25], or requiring the user to explicitly provide interaction delimiters, e.g., using a separate gesture such as a pinch or 'air push' [32, 50, 56], or an additional input modality such as eye gaze [35] or voice commands [45]. In this work, we investigate the potential of probabilistic classification to enhance robustness and experience of free-hand planar interaction in VR across a range of interaction modes.

### 2.2 Probabilistic Interaction Events in VR

Many interaction events in VR are inherently noisy and, therefore, difficult to detect [2, 3]. This holds especially true for free-hand interactions that need to be reliably recognized based on inaccurate and noisy hand and body tracking data [7]. One promising approach is *probabilistic classification*, which has emerged as an effective mechanism to deal with uncertainty.

Probabilistic classification refers to the task of predicting one or multiple class labels by estimating a probability distribution over a set of candidate classes [5]. It is a widely studied topic in machine learning and has been used to address diverse VR interaction tasks, including to robustly detect and classify gestures and poses [38, 42, 46], improve target selection [23, 30], and facilitate text entry [21, 39, 48]. Recent work has extended the scope of probabilistic inference methods to more creative and expressive interaction scenarios, including terrain generation and authoring [22], olfactory experiences where users can proactively interact with scented objects [26], and mixed reality music interfaces [19]. Moreover, sensing systems that can detect small amplitude hand and finger movements at a distance with high spatial accuracy have recently been demonstrated [1]. While these works highlight the potential of probabilistic sensor data processing and classifications for gesture-based interaction, the proposed methods usually require complex hardware and software setups tailored to specific use cases. In contrast, this paper proposes a novel approach for in-plane detection that is computationally efficient, conceptually simple and easily interpretable, and which applies to a range of standard HCI tasks.

One central theme in this line of research has been the disambiguation of user-intended interactions and unintended movement. For example, David-John *et al.* [10] used a logistic regression classifier based on eye-gaze features to predict whether users intended to select a virtual item; Nguyen *et al.* [33] employed linear discriminant analysis with physiological data to predict when a user intends to interact with the real or virtual environment; and Jo *et al.* [23] used eye gaze data to train a Support Vector Machine based on a Bayesian posterior probability distribution to infer a user's intention to select a given virtual target via gaze. Yi *et al.* [62] also tackle the problem of probabilistically detecting touches on a mid-air virtual plane. These works demonstrate the effectiveness of probabilistic methods to handle uncertain input data and decode user intent in VR contexts.

The most closely related work to our in-plane detection method comes from Bohari *et al.* [6], who trained a set of classifiers to distinguish between intended curves ('strokes') and non-drawing hand movements ('hovers') in mid-air 3D sketching tasks with a desktop display. Their best-performing classifier, a Random Forest-based model, achieved an accuracy of up to 80%. However, this classifier was implemented and tested in offline use, and it is unclear to what extent it generalizes to tasks other than sketching. Their work further focused on non-immersive sketching and therefore did not take into account VR specifics, such as consumer-level body tracking accuracy and variations in user depth perception. Finally, their classifier was trained on a large number of predefined input features related to the user's movement trajectories. In this work, we demonstrate that two features that can be computed in real time from HMD-based hand tracking data are sufficient for accurate classification and effective online performance.

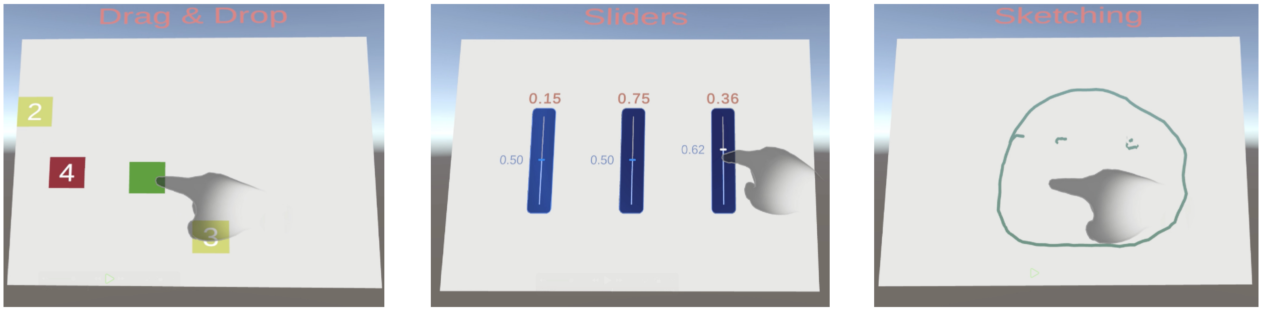
We use a Naïve Bayes classifier, a computationally efficient and often performant probabilistic learning algorithm derived from Bayes' theorem, which assumes conditionally independent input features [63], although this assumption can often be relaxed in practice [40]. It has previously been used in the context of VR interaction to detect and classify head, hand, arm, and body motions and poses [38]. This general approach was also used effectively by Foy *et al.* [17] on a distinct but related problem involving detecting unintended movement of fingers during 10 finger mid-air typing. In this work, we apply Naïve Bayes classification to distinguish intentional interactions (e.g., manipulating sliders, or drawing strokes) from incidental movements in 3D space. To our knowledge, this study represents the first application of probabilistic classification to the challenge of real-time detection for in-plane interaction in VR.

## 3 Data Collection

To assess the prevalence of unintentional departures from virtual planes during mid-air interaction and to identify associated user behavior patterns, we collected motion data for three common planar interaction tasks in VR.

### 3.1 Tasks

We developed an experimental application implementing three planar interaction tasks in VR as visualized in Figure 2. These are: (i) drag and drop; (ii) manipulating a slider; and (iii) 2D sketching. Each task was performed using free-hand interaction based on the body tracking feature implemented in the Meta Quest 3. While any deviation from the virtual plane towards the user resulted in an immediate drop, moving behind the plane did not affect the interaction. Prior to the actual data collection, participants were allowed to practice all three tasks in order to familiarize themselves with the requirements and interaction technique, thus reducing learning effects. At the beginning of each task, the virtual plane was placed 55 cm in front and 10 cm to the right of the tracked head position. The number of trials per task were chosen to ensure




**(a) Drag** **(b) Slide** **(c) Sketch**

**Figure 2: The three planar interaction tasks used for both data collection and evaluation of our classifier.**

comparable durations for each task block during the data collection. The three tasks are described in detail below.

*3.1.1 Drag and Drop.* Two-dimensional dragging constitutes a standard planar interaction task in HCI [49, 58]. Dragging involves carefully manipulating an object from an initial position to a desired target position while maintaining contact with the interaction plane. It is an ideal task for analyzing unintentional drops as interruptions are highly disruptive to the continuous object manipulation.

Our drag and drop task (for conciseness subsequently referred to as Drag) is shown in Figure 2a. The participant must enter the plane to 'pick up' the rectangular object at location 1 (green square). The object's style (transparent with outline only versus filled with green color) provides a visual indication of whether it is currently 'picked up'. Next, the participant has to drag this object through two intermediary targets (yellow squares labeled 2 and 3) in the prescribed order. Finally, the participant must drag the square to the 'target zone' (red square labeled 4) where it should be dropped. If the object is dropped outside the 'target zone', it can be picked up again to continue the task.

We include two intermediary targets in order to capture data on prolonged interactions that involve different movement directions along the plane and require the user to continuously integrate visual feedback. Notably, the resulting planar 'via-point task' has been the subject of significant interest in human motor control research [34, 51]. The core task also shares several characteristics with gesture typing, which is another widely studied mid-air interaction task in VR and AR [15, 44]. Both dragging and gesture typing require a continuously connected sequence of directed movements along a two-dimensional surface. We decided to omit gesture typing from this study due to its additional complexity and potential variability in participant familiarity, but view the implemented Drag task as a simpler, abstracted version of this interaction.

For every trial, the locations of all four squares were randomly sampled without replacement from a 10×8 equidistant grid of non-overlapping cells. Each participant completed 35 trials of this task.

*3.1.2 Manipulating a Slider.* Sliders are commonly used for selecting values on a discrete or continuous range by manipulating a draggable slider thumb [31, 56]. They form a core part of many graphical user interfaces and are implemented as standard components in almost all VR software development kits, including the Meta Interaction SDK[1] and the Microsoft MRTK3[2]. Based on the design of the MRTK3 sliders, our implementation allows us to study the challenges users face when trying to accurately position the slider thumb in the context of a mid-air user interface.

The slider manipulation task (Slide) is shown in Figure 2b. Participants are presented with three vertical sliders that need to be set to the target values shown above the respective slider tracks. The current value of each slider is determined from the handle position and is additionally displayed to the left of the handle. To match the displayed target value, the user must adjust the slider's vertical position along the track. Each slider can only be manipulated once per trial; if it is dropped, it becomes locked, indicated by the handle turning red. The three sliders differ in their step size (0.05, 0.25, and 0.01, with values on a range from 0 to 1), thus implementing different levels of slider resolution.

We defined five different sets of target values for the three sliders. Participants completed each set once, resulting in 15 trials of the slider task per participant.

*3.1.3 2D Sketching.* The third task is a sketching task, in which users are asked to draw a given figure onto a mid-air virtual plane. The challenges associated with mid-air 2D sketching have been widely studied [3, 6, 14]. Like handwriting or writing a signature, the task imposes only loose constraints on the drawn strokes without prescribing their exact size, shape, or order.

The 2D sketching task (Sketch) is shown in Figure 2c. Using the tip of their index finger, participants add strokes onto the canvas. The objective is to draw a 'smiley face' using as few strokes as possible (four is the minimum). One complete sketch represents a single trial. Participants completed three sketches in total, with the participant manually indicating the end of each sketch.

## 3.2 Participants

Eight participants (seven male and one female, all of them students or employees from our local university) were recruited to complete the data collection exercise. Participants were 29.38 years old on average, with a standard deviation of 5.63 years. All participants were right-handed and reported prior experience with VR.

[1] https://developers.meta.com/horizon/documentation/unity/unity-isdk-uiset#sliders
[2] https://learn.microsoft.com/en-us/windows/mixed-reality/mrtk-unity/mrtk3-uxcomponents/packages/uxcomponents/slider

Participants consented to their anonymized movement data being recorded, stored, and published, and were compensated with an Amazon voucher. The study was reviewed and approved by the Department's Research Ethics Committee.

## 3.3 Apparatus

Participants wore the Meta Quest 3[3] head-mounted display. The three tasks were implemented in a custom-built Unity application that allowed us to log body tracking data as well as object selection and drop events. Body tracking data was received from the Meta Movement SDK[4] at a frequency of 60 Hz. The application ran on a Windows 10 machine with an AMD Ryzen 7 and Nvidia RTX 4070 GPU. The headset was connected to the computer in Link mode.

## 3.4 Data Labeling

We reviewed the collected data, identified instances where the user departs from the interaction plane (i.e., a drop), and assigned each instance a class label indicating whether the drop was *intentional* or *unintentional*. We applied the following task-specific heuristics:

For Drag, all drops inside the target zone were assumed intentional, and all drops outside the target zone were assumed unintentional. For Sketch, each drop was labeled manually based on visual inspection of the motion trajectories. For example, sudden drops that occur before a circle has been closed or where the user later returns to continue drawing (see drops circled in red in Figure 5) were labeled unintentional. Drops followed by a clear movement towards another position (e.g., blue line in Figure 5 from left corner of mouth to right eye) were labeled as intentional.

For Slide, each drop received a score based on the distance between set slider value $s$ and target slider value $\hat{s}$:

$$1 - \frac{|s - \hat{s}|}{\max(\hat{s}, 1 - \hat{s})} \tag{1}$$

If the drop receives a score of 1, i.e., if the slider exactly reaches the target position, it is labeled as intentional. Drops with lower scores $s < 1$ are manually labeled, following a similar approach as in Sketch.

It is true that these heuristics may not fully capture the actual intention of the user. Nevertheless, they provide a good starting point for creating labeled training data. Notably, these labels are only used for training our classifier. For assessing classifier performance in use, we apply a more conservative and less subjective approach and calculate the number of *avoidable drops* in a given tasks as a proxy for the number of unintentional drops. Details on this calculation are provided later in Section 5.3.

## 3.5 Prevalence of Unintentional Departures

The observed frequency of unintentional drops per trial for each task is summarized in Table 1. It should be noted that drawing one face requires at least four strokes, so the number of unintentional drops on average for one stroke is $1.917/4 = 0.479$. From the table, it can be observed that people tend to have more unintentional drops for the drag and drop task, and less frequently for the slider task. This is in part a reflection of the fact that the slider task permitted only a single attempt and so the upper limit on unintentional drops for a trial is one.

**Table 1: Frequency of unintentional drops per trial across tasks. The Slide task allows for exactly one adjustment per trial, thus imposing an upper limit of 1. Each trial of the Sketch task requires at least four strokes.**

| Task | **Frequency (mean ± std)** |
|---|---|
| Drag | 0.8178 ± 0.7670 |
| Slide | 0.1167 ± 0.0927 |
| Sketch | 1.9167 ± 1.4117 |

## 4 Classification

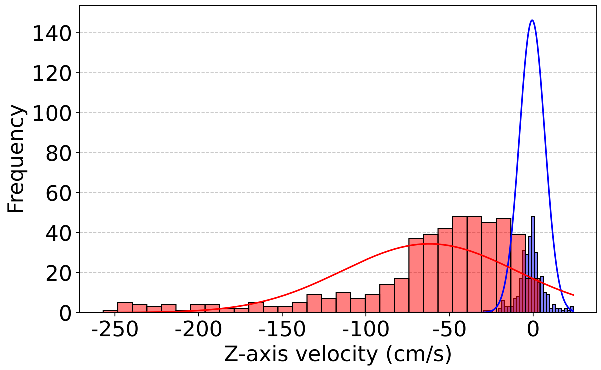
During mid-air interactions with the virtual plane, the $x$, $y$, and $z$ coordinates of the user's index finger are continuously tracked. From this spatial data, two key features are extracted within a 150 ms temporal window surrounding each drop event:

- Z-axis velocity ($v_z$): This represents the velocity of motion perpendicular to the virtual plane (in cm/s). It is calculated as the average of the instantaneous z-axis velocities between consecutive frames within the window. Each frame-wise velocity is computed as the z-axis displacement between adjacent frames divided by the elapsed time. Negative velocity is movement towards the user.
- Deviation ($d$): This is defined as the maximum absolute deviation of the fingertip from the plane (in the z-direction, with negative displacement being closer to the user) within the window (in cm).

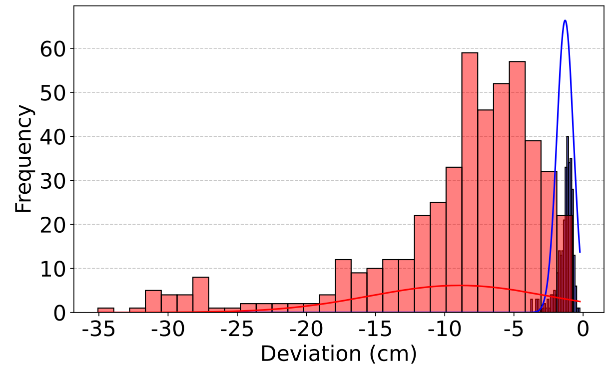
While we experimented with other features such as velocity tangential to the plane or using tracking data from different virtual markers, we found that the combination of (i) and (ii) yielded the highest success rate. Notably, adding feature (ii) resulted in a substantially improved classification accuracy. The 150 ms time window was chosen to include 40 ms before and 110 ms after a departure from the virtual plane was observed, resulting in a history-to-future ratio of approximately 1:3. Including frames after the drop allows the classifier to learn from post-drop behavior, which we found to be highly informative. Testing different window sizes (between 120 and 240 ms) and history-to-future ratios (1:3, 1:1, 3:1), we found this combination to be most successful, although it does effectively delay the decision time by 110 ms.

To differentiate between *intentional* and *unintentional* drops, we trained and evaluated a Naïve Bayes classifier [60] using the labeled dataset of 782 drops. We focused on this classifier, as it is easily interpretable and constitutes a conceptually and technically simple approach suitable for real-time usage on commercial VR hardware. Although Naïve Bayes assumes conditional independence of the included features, empirical evidence suggests its applicability to a broader range of use cases, especially when features are only weakly correlated [20]. The effectiveness of this 'naïve' approach was also confirmed by tests that replaced the classifier with a considerably more complicated model based on the Transformer architecture. The Transformer-based classifier did not substantially improve

[3]https://www.meta.com/quest/quest-3
[4]https://developers.meta.com/horizon/documentation/unity/move-body-tracking/




**(a) Velocity distribution.**




**(b) Deviation distribution.**

**Figure 3: Intentional (red) and unintentional (blue) drop histograms and fitted model distributions for z-axis velocity and deviation.**

performance, with overall accuracy within approximately ±1% of the Naïve Bayes classifier, regardless of the provided input features.

The Naïve Bayes classifier works as follows. Let $\{I, U\}$ denote the set of class labels, where I is a label for an intentional drop and U is a label for an unintentional drop. Let $\mathbf{x} = [v_z, d]^\top$ represent the feature vector consisting of the z-axis velocity $v_z$ and the deviation $d$ within a given window. The posterior probability of a drop belonging to class $C \in \{I, U\}$, given a vector of observed features $\mathbf{x}$, is then computed using Bayes' theorem:

$$P(C \mid \mathbf{x}) = \frac{P(C) \cdot P(\mathbf{x} \mid C)}{P(\mathbf{x})}, \quad \text{where } \mathbf{x} = [v_z, d]^\top,\ C \in \{I, U\}. \tag{2}$$

Here:

- $P(C)$ is the prior probability of class $C \in \{I, U\}$,
- $P(\mathbf{x} \mid C)$ is the class-conditional likelihood, following a multivariate Gaussian distribution,
- $P(\mathbf{x})$ is the marginal likelihood, ensuring normalization.

The dataset was partitioned into training and testing sets using an 80:20 split, using stratified sampling to preserve class distribution. The Gaussian assumption is justified by the approximately normal distributions observed for both $v_z$ and $d$, as discussed in Section 4.1.

After training, we can obtain a class-conditional probability for any test data point and use Bayes' decision rule (Equation 3) to determine its predicted class label:

$$\hat{C} = \arg\max_{C \in \{I, U\}} P(C) \cdot P(\mathbf{x} \mid C), \tag{3}$$

where, $\hat{C}$ denotes the predicted class.

### 4.1 Performance

The model parameters for the Naïve Bayes classifier determined from the training set are listed in Table 2. Intended drops exhibit substantially higher average velocities along the depth axis (-62.7 cm/s versus -1.4 cm/s) and occur much further from the plane (mean 9.0 cm versus 1.3 cm) than unintended drops over the 150 ms temporal window, suggesting good discriminative power for these features. Figure 3 shows good alignment between the raw data distribution and the model. Intended drops are more common (prior 0.619) than unintended drops (prior 0.381). Evaluated on the test set, the classifier achieves an overall accuracy of 93.6%, with high F1-scores of 0.95 and 0.92 for the intentional and unintentional drop classes on the held-out test set, respectively (see Table 3).

**Table 2: Naïve Bayes model parameters for each class: prior probabilities, means, and standard deviations for z-axis velocity ($v_z$, unit: cm/s) and deviation ($d$, unit: cm).**

| Class | Prior | Mean ($v_z$) | Std ($v_z$) | Mean ($d$) | Std ($d$) |
|---|---|---|---|---|---|
| I | 0.6190 | -62.6674 | 54.8915 | 8.98 | 6.53 |
| U | 0.3810 | -1.4303 | 7.6409 | 1.31 | 0.64 |

**Table 3: Classification performance of the Naïve Bayes model on the test set from our data collection.**

| Class | Precision | Recall | F1-score | Support |
|---|---|---|---|---|
| I | 0.99 | 0.91 | 0.95 | 96 |
| U | 0.87 | 0.98 | 0.92 | 60 |

## 5 Evaluation

Having established the classification performance of the Naïve Bayes classifier on the collected dataset, we evaluate its impact on interaction performance in a controlled user study. The goal of this study is to compare planar mid-air interaction with and without the classifier enabled. In the 'with' setting, the classifier is used to decide whether a departure from the virtual plane was intended or not. More precisely, whenever a deviation from the plane towards the user was detected, the system continued buffering data for another 110 ms and then computed features from the data obtained for the previous 150 ms. These features were then passed to the classifier, which decided whether to allow a drop. In the 'without' classifier setting, every departure from the plane immediately resulted in a drop. This variant is equivalent to the interaction configuration used for the data collection described in Section 3.

**Table 4: Unintentional versus avoidable drop counts in the initial dataset, using the labeling methods from Sections 3.5 and 5.3, respectively. For DRAG, both methods match by definition.**

| Task | Unintentional Drops | Avoidable Drops |
|---|---|---|
| DRAG | 229 | 229 |
| SLIDE | 14 | 3 |
| SKETCH | 46 | 56 |

## 5.1 Procedure

Participants performed the same three planar interaction tasks used to collect the training data (DRAG, SLIDE, SKETCH) both with and without the classifier. The order of the two interaction conditions was counterbalanced among participants. For each condition, each participant performed 35 DRAG trials, 15 SLIDE trials in batches of three, and six SKETCH trials in that order. The order of square locations in the DRAG task and of target values in the SLIDE task was fixed for participants and the two conditions.

After the three tasks were completed, we asked each participant to complete a System Usability Survey (SUS) and a NASA-TLX questionnaire. This allows us to assess the usability and perceived workload for each condition. Finally, we conducted a short semi-structured interview with each participant to obtain qualitative feedback on their perceptions of the two interaction conditions.

## 5.2 Participants

Sixteen participants were recruited from our local institution. Ten participants were male and six were female. The mean of age of the sample was 21.8 with a standard deviation of 1.28. All participants were right-handed and reported at least some previous experience in VR. None of the participants had taken part in the data collection exercise described in Section 3. The study was reviewed and approved by the Department's Research Ethics Committee.

## 5.3 Performance Metric

The key dependent variable used to evaluate the impact of introducing the classifier is the frequency of *avoidable drops*. The frequency of avoidable drops provides a proxy for unintentional drops by capturing the mean number of drops beyond the minimum required to complete a given trial. In the drag-and-drop task, each trial can be executed with a single entry and departure from the interaction plane. This means that observing more than one drop in a DRAG trial indicates some form of interaction error has occurred. Given that the task is very simple, it is reasonable to expect such errors are due to an unintentional drop. In the sketching task, participants must make a minimum of four strokes to draw a smiley face which requires a minimum of four entries and departures from the interaction plane. Therefore, observing more than four drops in a SKETCH trial indicates an interaction error may have occurred.

The calculation of the interaction performance for the SLIDE task is slightly different given that each slider allows only one attempt. In practice, a drop on a slider interface element that results a set value very far from the user's desired slider value would require the user to interact with the slider a second time, i.e., a potentially avoidable drop. Therefore, we treat trials in which the set slider value substantially differed from the target value (i.e., with score $s < 0.5$) as avoidable drops. Given that users were able to precisely control the slider value, this depicts a very conservative approach, which likely underestimates the number of true unintended drops.

To demonstrate this performance metric provides a reliable proxy measure for unintentional drops, we compute the number of avoidable drops for the dataset introduced in Section 3. Table 4 compares the automatically computed avoidable drops with the partially manually labeled unintentional drops. For the DRAG task, the counts are the same since unintentional and unavoidable drops are labeled in the same way. For the SLIDE task, the avoidable drops measure is more conservative since the acceptable final slider setting is very forgiving at 50% of the maximum range. For the SKETCH task, there are slightly more avoidable drops than unintentional drops. This is likely due to the fact that although a minimum of four strokes was required to produce the sketch, there was no explicit penalty on using more than four strokes.

**Table 5: Frequency of avoidable drops per trial across tasks in the evaluation (median and interquartile range) and statistical significance of classifier effects based on one-sided Wilcoxon signed rank tests. The SLIDE task allows for exactly one adjustment per trial, imposing an upper limit of 1.**

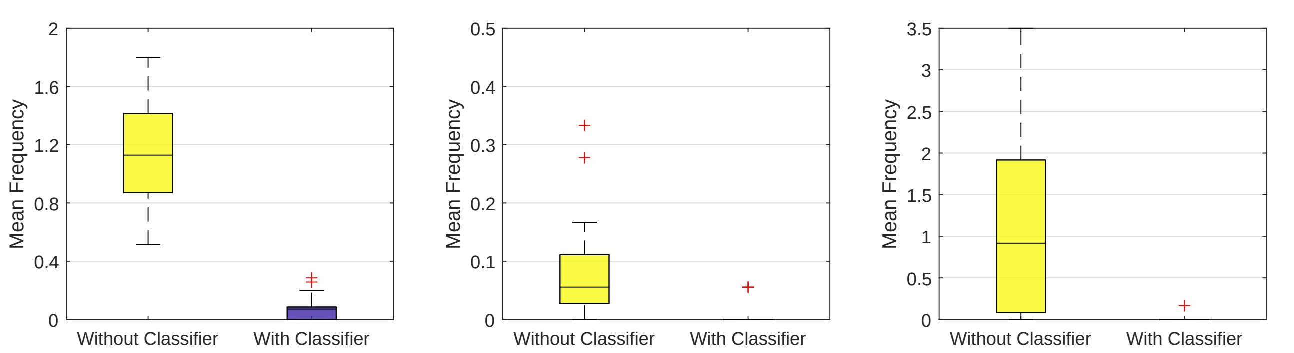
| | Frequency (median ± IQR) | | |
|---|---|---|---|
| Task | With Classifier | Without Classifier | $p$-Value |
| DRAG | 0.07 ± 0.09 | 1.13 ± 0.46 | 0.0002 |
| SLIDE | 0.00 ± 0.00 | 0.06 ± 0.07 | 0.0017 |
| SKETCH | 0.00 ± 0.00 | 0.92 ± 1.67 | 0.0012 |

## 5.4 Results

The impact of the classifier on participant performance in the three interaction tasks is summarized in Table 5 and Figure 4. In the DRAG task, the median frequency of avoidable drops was 0.07 with classifier and 1.13 without classifier. This means that one avoidable (and potentially unintended) drop occurred once every trial on average without the classifier, but only every 14th trial when the classifier was used. In the SLIDE task with classifier, only three participants performed a single avoidable drop each, resulting in a median frequency of 0. Without classifier, a median frequency of 0.06 was observed.

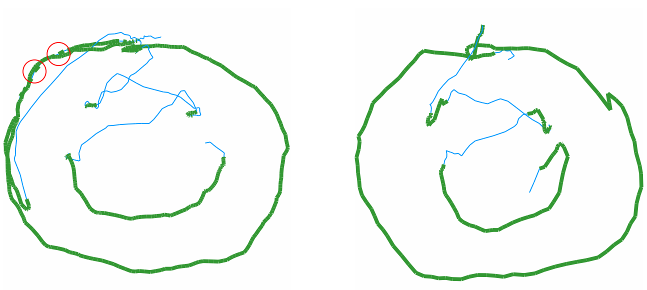
Similar results were obtained for the SKETCH task. Only a single smiley drawn with the classifier enabled required more than the minimum number of four strokes (out of the 96 sketches performed by participants in the study in this condition). This resulted in a median avoidable drops frequency of 0. In contrast, a median frequency of 0.92 was observed without classifier, corresponding to approximately one avoidable drop for every smiley drawn on average.

The with classifier distributions shown in Figure 4 are skewed due to the fact that the lower bound on the frequency measure is zero. We tested for normality using the Kolmogorov-Smirnov test and found that the data in the 'with classifier' condition was indeed not normally distributed, and so we employed one-sided Wilcoxon signed rank tests to analyze for statistical significance.




(a) Drag (b) Slide (c) Sketch

**Figure 4: Boxplots of the participants' mean frequency of avoidable drops per trial during the three planar interaction tasks with and without the Naïve Bayes plane departure classification.**



(a) Without classifier. (b) With classifier.

**Figure 5: Strokes (green) and transitions (blue) in the Sketch task. (a) Sketching without classifier is prone to unintended drops (red circles), which require the user to redraw this portion of the outline. (b) With our classifier enabled, unintended departures from the plane are automatically handled, resulting in a more stable and less disruptive workflow.**

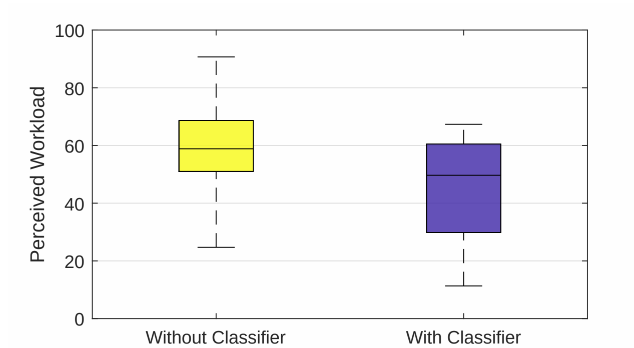



**Figure 6: Perceived Workload as measured by NASA-TLX for the evaluation tasks with and without the Naïve Bayes classifier.**

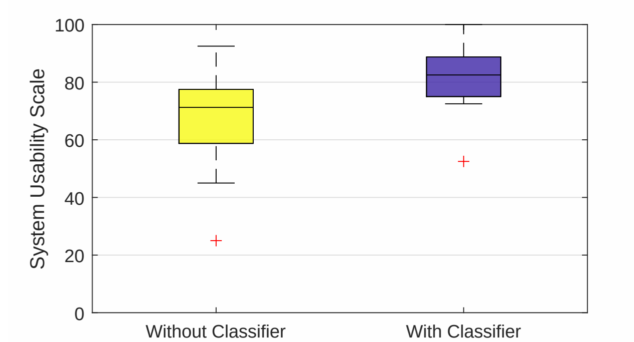



**Figure 7: SUS measure for the evaluation tasks with and without the Naïve Bayes classifier.**

This analysis revealed that the Naïve Bayes classifier significantly reduced the number of avoidable drops in all three tasks: Drag ($Z = 3.490$, $p = 0.0002$), Slide ($Z = 2.926$, $p = 0.0017$), and Sketch ($Z = 3.024$, $p = 0.0012$).

This severe difference in performance is also directly visible in the strokes made by participants in the Sketch task. Figure 5a shows an example sketch from one of the participants, created without classifier. The red circles highlight unintentional drops when drawing the outline of the smiley face, disrupting the interaction flow and requiring the user to return later to redraw this portion of the outline. Figure 5b conversely depicts a smiley face drawn with the classifier enabled. Here, the smiley can be completed with the minimum number of four strokes, resulting in a higher quality outcome.

Furthermore, repeated measures ANOVAs showed that using the classifier led to significantly lower perceived workload as measured by the NASA-TLX ratings ($F(1, 15) = 10.579, p = 0.005, \eta_p^2 = 0.414$) and significantly higher perceived usability according to the SUS scores ($F(1, 15) = 17.753, p < 0.001, \eta_p^2 = 0.542$).

*5.4.1 Qualitative Feedback.* Feedback from the interviews after the experiment revealed consistent themes. Across all participants, it was reported that the classifier made the drag-and-drop task and the sketching task more stable and reduced unintentional drops. For example, *P7* commented that, "with the classifier, [the system] is better to use, a lot less frustrating, and a bit more responsive." Several participants also described the interaction with the classifier as "more natural" (*P4*, *P8*) and "smoother" (*P9*, *P10*, *P11*). Moreover, it was observed that with the classifier enabled, less effort was required to maintain control and avoid unintentional drops. As *P1* noted, "while the condition without the classifier was too sensitive in terms of the control movement, the classifier condition seemed more user-friendly. There was no need to take more effort in dragging items, or to poke inside the plane too deep, therefore [it is] easier to retract and leave the plane."

However, the majority of participants found that the high-sensitivity slider task remained somewhat difficult to complete precisely with the classifier. *P10* suggested this was mainly because, "even when the hand has left the plane, the value still changes a bit because of lateral movement." *P6* hypothesized that, "when dropping [the slider thumb], the error of the value is higher [with classifier] because it is not released instantly." It is indeed possible that there was a short perceptible delay caused by calling the classifier with 110 ms of data collected *after* a departure from the virtual plane.

## 6 Discussion

We identified two spatiotemporal features that convey meaningful information about the user's intent to depart from a virtual 2D surface. These are the z-axis velocity and plane deviation of the user's fingertip as described in Section 4. These features show good discriminative power and can be easily calculated from the hand tracking data provided by commercial VR devices. As we have demonstrated in this work, a simple probabilistic method such as the Naïve Bayes classifier trained on these two features can considerably improve mid-air interaction by reducing the number of unintended drops.

Although we covered three relevant planar interaction tasks in the study, there are several limitations that should be noted when considering our findings. First, all conditions shared the same task setup with the plane oriented vertically and directly facing the user. Considering the complex, non-linear nature of human shoulder and arm movements [18, 52], it is unclear whether drops in other directions, e.g., moving the arm upwards to withdraw from a horizontal plane as common in virtual typing, can be analogously captured by modulating our features with respect to the plane's normal vector. Future work should therefore investigate the robustness of our approach in terms of spatial variability. Similarly, in tasks that require high end-point accuracy, such as slider manipulation, we foresee potential for more sophisticated classification methods that may help reduce latency and increase usability (e.g., early classification methods [9]).

Second, body movements may vary greatly across users. While our classifier was generally well received by all 16 participants, further evaluations on a more diverse set of users are needed. Those might include elderly users, children, or people with motor impairments. In addition, training task- and user-specific classifiers might further improve prediction quality. For example, we hypothesize that group-specific classifiers trained on movement data from Parkinson's patients could be key to increasing the accessibility of free-hand planar VR interaction. Alternatively, it might be worth investigating how domain and user knowledge can be effectively integrated into a prior distribution of intended and unintended drops.

Finally, we hope that this work encourages the community to reconsider the demands of tracking-based mid-air interaction on a more general level. While several classification methods have recently been proposed and applied to infer user intent across a range of tasks and techniques [6, 10, 23, 33], they typically require between 8 and 200 input features, most of which are overly complex and fairly difficult to interpret. Our work shows that good user outcomes can be achieved with much less: We find that two carefully chosen features are sufficient to significantly reduce the frequency of unintended drops and improve stability and subjective perception of free-hand planar interaction across a range of interaction tasks. Future work should investigate the robustness of the proposed classifier, e.g., with respect to temporary variations in the frame rate, or when data is noisy or incomplete. It should also compare Naïve Bayes with related methods such as Random Forest or Support Vector Machine (SVM) and explore practical trade-offs between these techniques. Moreover, it would be interesting to see how other forms of planar mid-air interaction, such as gesture typing, handwriting, or scrolling, may benefit from a simple probabilistic classification approach. Ultimately, this could lead to more natural and intuitive VR interactions that reduce cognitive load and computation by keeping things simple—both for the user and the system.

## 7 Conclusion

This paper addressed the challenge of providing robust 2D virtual surface interactions in VR. In this interaction setting, the lack of haptic feedback and noisy hand tracking can lead to unintentional departures from the interaction plane that cause frustrating interruptions for the user. We presented and evaluated a Naïve Bayes classifier using two readily available kinematic features—fingertip z-axis velocity and z-axis deviation from the plane—to probabilistically distinguish intentional planar departures from unintentional movements in 3D space. Our user study demonstrated that this approach can significantly reduce the frequency of unintended drops across drag-and-drop, slider manipulation, and sketching tasks, improving perceived usability and lowering workload compared to a conventional depth-threshold method. These findings highlight how probabilistic techniques can help leverage consumer-level tracking data to deliver more natural, reliable, and delimiter-free direct manipulation in VR, paving the way for richer free-hand experiences.

## Acknowledgments

Florian Fischer, Per Ola Kristensson and John J. Dudley were supported by the Engineering and Physical Sciences Research Council [grant number EP/W02456X/1]. Supporting data is available at https://doi.org/10.17863/CAM.121310.

## References

[1] Yuta Abe, Taishi Iriyama, Takashi Komuro, Kohei Shimasaki, and Idaku Ishii. 2024. A Mid-air Multi-touch Interface using an Ultrafast Pan-tilt Camera. In

*Extended Abstracts of the CHI Conference on Human Factors in Computing Systems*. 1–6.
[2] Ferran Argelaguet and Carlos Andujar. 2013. A Survey of 3D Object Selection Techniques for Virtual Environments. *Computers & Graphics* 37, 3 (May 2013), 121–136. doi:10.1016/j.cag.2012.12.003
[3] Rahul Arora, Rubaiat Habib Kazi, Fraser Anderson, Tovi Grossman, Karan Singh, and George Fitzmaurice. 2017. Experimental Evaluation of Sketching on Surfaces in VR. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems* (Denver, Colorado, USA) *(CHI '17)*. ACM, New York, NY, USA, 5643–5654. doi:10.1145/3025453.3025474
[4] Lonni Besançon, Anders Ynnerman, Daniel F Keefe, Lingyun Yu, and Tobias Isenberg. 2021. The state of the art of spatial interfaces for 3D visualization. In *Computer Graphics Forum*, Vol. 40. Wiley Online Library, 293–326.
[5] Christopher M. Bishop. 2006. *Pattern Recognition and Machine Learning*. Springer, New York.
[6] Umema Bohari, Ting-Ju Chen, and Vinayak. 2018. To Draw or Not to Draw: Recognizing Stroke-Hover Intent in Non-instrumented Gesture-free Mid-Air Sketching. In *23rd International Conference on Intelligent User Interfaces*. ACM, Tokyo Japan, 177–188. doi:10.1145/3172944.3172985
[7] Gavin Buckingham. 2021. Hand Tracking for Immersive Virtual Reality: Opportunities and Challenges. *Frontiers in Virtual Reality* 2 (Oct. 2021). doi:10.3389/frvir.2021.728461
[8] Li-Wei Chan, Hui-Shan Kao, Mike Y Chen, Ming-Sui Lee, Jane Hsu, and Yi-Ping Hung. 2010. Touching the void: Direct-touch interaction for intangible displays. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. 2625–2634.
[9] Asma Dachraoui, Alexis Bondu, and Antoine Cornuéjols. 2015. Early classification of time series as a non myopic sequential decision making problem. In *Machine Learning and Knowledge Discovery in Databases: European Conference, ECML PKDD 2015, Porto, Portugal, September 7-11, 2015, Proceedings, Part I 15*. Springer, 433–447.
[10] 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*. 1–7.
[11] Shiyu Duan, Ziyi Wang, Shixiao Wang, Mengmeng Chen, and Runsheng Zhang. 2024. Emotion-Aware Interaction Design in Intelligent User Interface Using Multi-Modal Deep Learning. In *2024 5th International Symposium on Computer Engineering and Intelligent Communications (ISCEIC)*. 110–114. doi:10.1109/ISCEIC63613.2024.10810240
[12] Tafadzwa Joseph Dube and Ahmed Sabbir Arif. 2023. Ultrasonic keyboard: A mid-air virtual qwerty with ultrasonic feedback for virtual reality. In *Proceedings of the Seventeenth International Conference on Tangible, Embedded, and Embodied Interaction*. 1–8.
[13] John Dudley, Hrvoje Benko, Daniel Wigdor, and Per Ola Kristensson. 2019. Performance envelopes of virtual keyboard text input strategies in virtual reality. In *2019 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*. IEEE, 289–300.
[14] John J. Dudley, Hendrik Schuff, and Per Ola Kristensson. 2018. Bare-Handed 3D Drawing in Augmented Reality. In *Proceedings of the 2018 Designing Interactive Systems Conference* (Hong Kong, China) *(DIS '18)*. ACM, New York, NY, USA, 241–252. doi:10.1145/3196709.3196737
[15] J. J. Dudley, J. Zheng, A. Gupta, H. Benko, M. Longest, R. Wang, and P. Kristensson. 2023. Evaluating the Performance of Hand-Based Probabilistic Text Input Methods on a Mid-Air Virtual Qwerty Keyboard. *IEEE Transactions on Visualization and Computer Graphics* 29, 11 (nov 2023), 4567–4577. doi:10.1109/TVCG.2023.3320238
[16] Fatima El Jamiy and Ronald Marsh. 2019. Survey on depth perception in head mounted displays: distance estimation in virtual reality, augmented reality, and mixed reality. *IET Image Processing* 13, 5 (2019), 707–712.
[17] Conor R. Foy, John J. Dudley, Aakar Gupta, Hrvoje Benko, and Per Ola Kristensson. 2021. Understanding, Detecting and Mitigating the Effects of Coactivations in Ten-Finger Mid-Air Typing in Virtual Reality. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems* (Yokohama, Japan) *(CHI '21)*. ACM, New York, NY, USA, Article 287, 11 pages. doi:10.1145/3411764.3445671
[18] Claude Ghez, James Gordon, Maria Felice Ghilardi, CN Christakos, and Scott E Cooper. 1990. Roles of proprioceptive input in the programming of arm trajectories. In *Cold spring harbor symposia on quantitative biology*, Vol. 55. Cold Spring Harbor Laboratory Press, 837–847.
[19] Max Graf and Mathieu Barthet. 2022. Mixed reality musical interface: Exploring ergonomics and adaptive hand pose recognition for gestural control. In *NIME 2022*.
[20] David J Hand and Keming Yu. 2001. Idiot's Bayes—not so stupid after all? *International statistical review* 69, 3 (2001), 385–398.
[21] Jinghui Hu, John J Dudley, and Per Ola Kristensson. 2024. Skimr: Dwell-free eye typing in mixed reality. In *2024 IEEE Conference Virtual Reality and 3D User Interfaces (VR)*. IEEE, 439–449.
[22] Yushen Hu, Keru Wang, Yuli Shao, Jan Plass, Zhu Wang, and Ken Perlin. 2024. Generative Terrain Authoring with Mid-air Hand Sketching in Virtual Reality. In *Proceedings of the 30th ACM Symposium on Virtual Reality Software and Technology*. 1–10.
[23] Taewoo Jo, Ho Jung Lee, Sulim Chun, and In-Kwon Lee. 2024. Predicting Selection Intention in Real-Time with Bayesian-based ML Model in Unimodal Gaze Interaction. *arXiv preprint arXiv:2411.06726* (2024).
[24] Jinwook Kim, Sangmin Park, Qiushi Zhou, Mar Gonzalez-Franco, Jeongmi Lee, and Ken Pfeuffer. 2025. PinchCatcher: Enabling Multi-selection for Gaze+ Pinch. In *Proceedings of the 2025 CHI Conference on Human Factors in Computing Systems*. 1–16.
[25] Per Ola Kristensson, Thomas Nicholson, and Aaron Quigley. 2012. Continuous Recognition of One-Handed and Two-Handed Gestures Using 3D Full-Body Motion Tracking Sensors. (2012).
[26] Junxian Li, Yanan Wang, Zhitong Cui, Jas Brooks, Yifan Yan, Zhengyu Lou, and Yucheng Li. 2025. Mid-Air Gestures for Proactive Olfactory Interactions in Virtual Reality. In *Proceedings of the 2025 CHI Conference on Human Factors in Computing Systems*. 1–18.
[27] Yang Li, Ken Hinckley, Zhiwei Guan, and James A Landay. 2005. Experimental analysis of mode switching techniques in pen-based user interfaces. In *Proceedings of the SIGCHI conference on Human factors in computing systems*. 461–470.
[28] Lawrence Makin, Gareth Barnaby, and Anne Roudaut. 2019. Tactile and kinesthetic feedbacks improve distance perception in virtual reality. In *Proceedings of the 31st Conference on l'Interaction Homme-Machine*. 1–9.
[29] Pedro Monteiro, Hugo Coelho, Guilherme Gonçalves, Miguel Melo, and Maximino Bessa. 2023. Exploring the user experience of hands-free VR interaction methods during a Fitts' task. *Computers & Graphics* 117 (2023), 1–12.
[30] Hee-Seung Moon, Yi-Chi Liao, Chenyu Li, Byungjoo Lee, and Antti Oulasvirta. 2024. Real-time 3d target inference via biomechanical simulation. In *Proceedings of the 2024 CHI Conference on Human Factors in Computing Systems*. 1–18.
[31] Tomer Moscovich. 2009. Contact area interaction with sliding widgets. In *Proceedings of the 22nd annual ACM symposium on User interface software and technology*. 13–22.
[32] Aunnoy K Mutasim, Anil Ufuk Batmaz, and Wolfgang Stuerzlinger. 2021. Pinch, click, or dwell: Comparing different selection techniques for eye-gaze-based pointing in virtual reality. In *ACM Symposium on Eye Tracking Research and Applications*. 1–7.
[33] Willy Nguyen, Klaus Gramann, and Lukas Gehrke. 2024. Modeling the Intent to Interact With VR Using Physiological Features. *IEEE Transactions on Visualization and Computer Graphics* 30, 8 (2024), 5893–5900. doi:10.1109/TVCG.2023.3308787
[34] Rieko Osu, Ken-ichi Morishige, Jun Nakanishi, Hiroyuki Miyamoto, and Mitsuo Kawato. 2015. Practice reduces task relevant variance modulation and forms nominal trajectory. *Scientific reports* 5, 1 (2015), 17659.
[35] Ken Pfeuffer, Benedikt Mayer, Diako Mardanbegi, and Hans Gellersen. 2017. Gaze + pinch interaction in virtual reality. In *Proceedings of the 5th Symposium on Spatial User Interaction*. ACM, 99–108. doi:10.1145/3131277.3132180
[36] Jiamin Ping, Yue Liu, and Dongdong Weng. 2019. Comparison in depth perception between virtual reality and augmented reality systems. In *2019 IEEE Conference on Virtual Reality and 3D User Interfaces (VR)*. IEEE, 1124–1125.
[37] Thammathip Piumsomboon, David Altimira, Hyungon Kim, Adrian Clark, Gun Lee, and Mark Billinghurst. 2014. Grasp-Shell vs gesture-speech: A comparison of direct and indirect natural interaction techniques in augmented reality. In *2014 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*. IEEE, 73–82.
[38] Paweł Pięta, Hubert Jegierski, Paweł Babiuch, Maciej Jegierski, Mirosław Płaza, Grzegorz Łukawski, Stanisław Deniziak, Artur Jasiński, Jacek Opałka, Paweł Węgrzyn, Magdalena Igras-Cybulska, and Adrian Łapczyński. 2024. Automated Classification of Virtual Reality User Motions Using a Motion Atlas and Machine Learning Approach. *IEEE Access* 12 (2024), 94584–94609. doi:10.1109/ACCESS.2024.3424930
[39] Mark Richardson, Matt Durasoff, and Robert Wang. 2020. Decoding surface touch typing from hand-tracking. In *Proceedings of the 33rd annual ACM symposium on user interface software and technology*. 686–696.
[40] Irina Rish et al. 2001. An empirical study of the naive Bayes classifier. In *IJCAI 2001 workshop on empirical methods in artificial intelligence*, Vol. 3. Seattle, USA, 41–46.
[41] Jaime Ruiz and Yang Li. 2011. DoubleFlip: a motion gesture delimiter for mobile interaction. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. 2717–2720.
[42] K Martin Sagayam and D Jude Hemanth. 2017. Hand posture and gesture recognition techniques for virtual reality applications: a survey. *Virtual Reality* 21 (2017), 91–107.
[43] 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*. 1–19.
[44] Junxiao Shen, John Dudley, and Per Ola Kristensson. 2023. Fast and Robust Mid-Air Gesture Typing for AR Headsets Using 3D Trajectory Decoding. *IEEE Transactions on Visualization and Computer Graphics* (2023).

[45] Jaisie Sin and Cosmin Munteanu. 2020. Let's go there: Voice and pointing together in vr. In *22nd International Conference on Human-Computer Interaction with Mobile Devices and Services*. 1–3.

[46] Zhaomou Song, John J Dudley, and Per Ola Kristensson. 2023. Hotgestures: Complementing command selection and use with delimiter-free gesture-based shortcuts in virtual reality. *IEEE Transactions on Visualization and Computer Graphics* 29, 11 (2023), 4600–4610.

[47] Becky Spittle, Maite Frutos-Pascual, Chris Creed, and Ian Williams. 2022. A review of interaction techniques for immersive environments. *IEEE Transactions on Visualization and Computer Graphics* 29, 9 (2022), 3900–3921.

[48] Paul Streli, Jiaxi Jiang, Andreas Rene Fender, Manuel Meier, Hugo Romat, and Christian Holz. 2022. TapType: Ten-finger text entry on everyday surfaces via Bayesian inference. In *Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems*. 1–16.

[49] S Shyam Sundar, Saraswathi Bellur, Jeeyun Oh, Qian Xu, and Haiyan Jia. 2014. User experience of on-screen interaction techniques: An experimental investigation of clicking, sliding, zooming, hovering, dragging, and flipping. *Human–Computer Interaction* 29, 2 (2014), 109–152.

[50] Hemant Bhaskar Surale, Fabrice Matulic, and Daniel Vogel. 2019. Experimental analysis of barehand mid-air mode-switching techniques in virtual reality. In *Proceedings of the 2019 CHI conference on human factors in computing systems*. 1–14.

[51] Emanuel Todorov. 2004. Optimality principles in sensorimotor control. *Nature neuroscience* 7, 9 (2004), 907–915.

[52] Robert J Van Beers, Patrick Haggard, and Daniel M Wolpert. 2004. The role of execution noise in movement variability. *Journal of neurophysiology* 91, 2 (2004), 1050–1063.

[53] Peter J Van de Broek, Clement Onime, James O Uhomobhi, and Mattia Santachiara. 2022. Evolution of user interface and user experience in mobile augmented and virtual reality applications. (2022).

[54] Tianyi Wang, Xun Qian, Fengming He, Xiyun Hu, Yuanzhi Cao, and Karthik Ramani. 2021. Gesturar: An authoring system for creating freehand interactive augmented reality applications. In *The 34th Annual ACM Symposium on User Interface Software and Technology*. 552–567.

[55] Colin Ware and Harutune H Mikaelian. 1986. An evaluation of an eye tracker as a device for computer input2. In *Proceedings of the SIGCHI/GI conference on Human factors in computing systems and graphics interface*. 183–188.

[56] Kieran Waugh, Mark McGill, and Euan Freeman. 2022. Push or Pinch? Exploring Slider Control Gestures for Touchless User Interfaces. In *Nordic Human-Computer Interaction Conference (NordiCHI '22)*. ACM, New York, NY, USA, 1–10. doi:10.1145/3546155.3546702

[57] Yushi Wei, Rongkai Shi, Difeng Yu, Yihong Wang, Yue Li, Lingyun Yu, and Hai-Ning Liang. 2023. Predicting gaze-based target selection in augmented reality headsets based on eye and head endpoint distributions. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems*. 1–14.

[58] Jason Wu, Rebecca Krosnick, Eldon Schoop, Amanda Swearngin, Jeffrey P Bigham, and Jeffrey Nichols. 2023. Never-ending learning of user interfaces. In *Proceedings of the 36th Annual ACM Symposium on User Interface Software and Technology*. 1–13.

[59] Yongkang Xing, Jethro Shell, Conor Fahy, Kexin Guan, Qian Zhang, and Tiande Xie. 2021. User interface research in web extended reality. In *2021 IEEE 7th International Conference on Virtual Reality (ICVR)*. IEEE, 76–81.

[60] Feng-Jen Yang. 2018. An implementation of naive bayes classifier. In *2018 International conference on computational science and computational intelligence (CSCI)*. IEEE, 301–306.

[61] Adriel Yeo, Benjamin WJ Kwok, Angelene Joshna, Kan Chen, and Jeannie SA Lee. 2024. Entering the next dimension: A review of 3d user interfaces for virtual reality. *Electronics* 13, 3 (2024), 600.

[62] Xin Yi, Chen Liang, Haozhan Chen, Jiuxu Song, Chun Yu, and Yuanchun Shil. 2022. From 2D to 3D: Facilitating Single-Finger Mid-Air Typing on Virtual Keyboards with Probabilistic Touch Modeling. In *2022 IEEE Conference on Virtual Reality and 3D User Interfaces Abstracts and Workshops (VRW)*. 694–695. doi:10.1109/VRW55335.2022.00198

[63] Harry Zhang. 2004. The Optimality of Naive Bayes. In *Proceedings of the Seventeenth International Florida Artificial Intelligence Research Society Conference (FLAIRS 2004)*. Miami Beach, FL, USA, 562–567.