Understanding, Detecting and Mitigating the Effects of Coactivations in Ten-Finger Mid-Air Typing in Virtual Reality

Conor R. Foy, John J. Dudley
Department of Engineering
University of Cambridge
Cambridge, United Kingdom

Aakar Gupta, Hrvoje Benko
Facebook Reality Labs
Seattle, Washington, USA

Per Ola Kristensson
Department of Engineering
University of Cambridge
Cambridge, United Kingdom

ABSTRACT

Typing with ten fingers on a virtual keyboard in virtual or augmented reality exposes a challenging input interpretation problem. There are many sources of noise in this interaction context and these exacerbate the challenge of accurately translating human actions into text. A particularly challenging input noise source arises from the physiology of the hand. Intentional finger movements can produce unintentional coactivations in other fingers. On a physical keyboard, the resistance of the keys alleviates this issue. On a virtual keyboard, coactivations are likely to introduce spurious input events under a naïve solution to input detection. In this paper we examine the features that discriminate intentional activations from coactivations. Based on this analysis, we demonstrate three alternative coactivation detection strategies with high discrimination power. Finally, we integrate coactivation detection into a probabilistic decoder and demonstrate its ability to further reduce uncorrected character error rates by approximately 10% relative and 0.9% absolute.

CCS CONCEPTS

• **Human-centered computing** → **Virtual reality**; **Text input**.

KEYWORDS

virtual reality, text entry

ACM Reference Format:
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 *CHI Conference on Human Factors in Computing Systems (CHI '21), May 8–13, 2021, Yokohama, Japan.* ACM, New York, NY, USA, 11 pages. https://doi.org/10.1145/3411764.3445671

1 INTRODUCTION

The freedom afforded by modern virtual and augmented reality headsets providing egocentric hand tracking encourages the exploration of effective bare-handed mid-air text entry methods. Ten finger typing is a familiar and efficient text entry strategy in conventional computing environments and with further hand tracking improvements may become feasible on virtual and augmented reality headsets. Mid-air typing with ten fingers suffers, however, from the lack of a fixed physical reference frame and passive resistance to key presses. This fundamental distinction from typing on a physical keyboard exposes the nontrivial challenge of discriminating intended finger movements from those produced by coupled motion of the hand and fingers. In this paper, we examine the features that distinguish intentional finger movements from unintentional finger movements in mid-air typing. For conciseness, we subsequently refer to these unintentional finger movements as *coactivations*.

A simple and highly extensible strategy for providing a virtual keyboard is the virtualization of the input surface. Intersections between the tracked fingertips and this virtual surface can thus be treated as input observations. Under this detection strategy, however, coactivations that intersect with the input surface introduce spurious observations. Unintentional observations that insert corresponding characters into the input field are frustrating to users and likely to yield high error rates.

The input detection strategy based on intersections with a virtual surface (described above) is simple and effective but suffers more from coactivations than other potential strategies. A deep neural network might, for example, be trained to decode tracked finger movements directly into characters or words and learn to ignore finger coactivations. This approach is, however, less extensible than the virtualized input surface and would necessitate large amounts of user data which is currently difficult to collect and not widely available. In this paper, we present the design of a novel software module capable of detecting coactivations based on interpretable dynamic features of finger motion while typing. This design strategy ensures that the proposed technique has utility beyond the confines of the specific keyboard configuration examined. Further, the module is readily integrated into a probabilistic decoding framework based on the virtual input surface paradigm to complement deletion assessments for input observations in the decoder.

To better understand coactivations and evaluate detection strategies, we leverage a dataset previously collected by Dudley et al. [6] of 24 users typing on a mid-air virtual keyboard in virtual reality (VR) with high-precision finger tracking. We mine this dataset to develop an understanding of the causal factors and distributions related to coactivations. This understanding supports the identification of the dominant features that distinguish coactivations from intended finger movements. The dataset is subsequently partitioned to evaluate the performance of three alternative detection strategies in terms of detection accuracy. Finally, to demonstrate the utility of the developed technique we integrate coactivation detection functionality into a probabilistic input decoding framework. The key contributions of this paper are therefore:

CHI '21, May 8–13, 2021, Yokohama, Japan

ACM ISBN 978-1-4503-8096-6/21/05.
https://doi.org/10.1145/3411764.3445671

(1) A characterization of coactivations as encountered in ten finger mid-air typing in virtual reality.
(2) An exposition of the features that distinguish coactivations from intentional finger movements in this setting.
(3) Three coactivation detection strategies based on the identified features, evaluated on the partitioned dataset.
(4) An integration of coactivation detection into a statistical decoder and a demonstration of the benefit coactivation detection can provide in reducing character error rates.

2 RELATED WORK

This work is informed by prior research efforts exploring mid-air text entry and other more general studies examining synergistic movements of the hand. These two areas are briefly reviewed in the following subsections.

2.1 Mid-Air Text Entry

There have been many mid-air text entry strategies proposed and evaluated but no dominant method has emerged [4, 7, 10, 12]. Bowman et al. [4] evaluated four different text input techniques for immersive virtual environments: speech, glove, pen and chording keyboard approaches achieving 13 wpm[1], 6 wpm, 10 wpm and 4 wpm respectively. Bowman et al. [4]'s work serves to highlight that different techniques offer certain trade-offs in terms of speed and accuracy, and that the most appropriate choice may depend on context.

The virtualization of the keyboard opens up new opportunities in terms of exploring exotic key layouts [22, 31] and interactions [3, 14, 19, 21, 27]. However, the Qwerty layout and touch interaction remain the de-facto standard in modern computing and it is highly desirable to leverage this familiarity where possible. ATK [30] demonstrates the potential (29 wpm) of ten-finger mid-air typing with fingers tracked using a leap motion. Dudley et al. [6] investigated the performance limits of typing on a virtual keyboard in VR. Users were observed to have significant difficulty in effectively utilizing all ten fingers in this setting [6]. Nevertheless, participants achieved notional entry rates in the order of 30 to 35 wpm on a mid-air virtual keyboard with ten fingers, albeit with relatively high error rates [6]. These results suggest that traditional designs fail to fully support the user in ten finger typing on a virtual keyboard. Others have tackled related mid-air text entry scenarios, such as supporting input on large wall displays [18, 19, 25]. However, such contexts rarely demand prolonged or high rate text input.

The new capabilities afforded by virtual keyboards also expose new challenges. For example, mid-air text entry can quickly become uncomfortable due to the need to maintain the arm cantilevered in space. Yanagihara et al. [29] describe a curved virtual keyboard for word-gesture text entry, in part designed to reduce fatigue. Another important consideration exposed in VR is how best to represent hands and fingertips to support text entry [11, 17]. These studies and the broader related work covered in this section highlight that supporting efficient and enjoyable text entry in VR is nontrivial. An effective solution, particularly when all ten fingers are engaged, hinges on being able to mitigate the aspects that make a virtual keyboard distinct from a physical keyboard.

[1] wpm = words per minute, with a word defined as five consecutive characters including space.

2.2 Unintended Finger Movements in Typing

Our hands are capable of an extremely wide range of both dynamic and fine motion. This is achieved by a complicated structure consisting of several types of tissue including tendons, ligaments, and muscles [16, 20, 28]. The variety of tissues connecting different components within the hand also imposes certain constraints. For example, the middle and ring digits share ligaments and other connective tissues which prevents one from moving through its full range of motion without movement of the other.

As well as being physically constrained, the movement of our muscles is also constrained by our neuromuscular system. These constraints are used to reduce the number of degrees of freedom that need to be independently controlled, making it easier to control a group of muscles at once [24]. These physical and neuromuscular constraints combine to make it difficult to control one muscle without unintentionally moving other ones. This coupled motion of the human digits is fundamental to the mechanisms that cause coactivations. Understanding these mechanisms may provide useful insights into how best to detect them. Soechting and Flanders [26] investigated the 22 degrees of freedom of the hand and fingers and found hierarchical relationships between these degrees of freedom producing consistent repeatability of given keystrokes. This indicates that the nervous system imposes constraints to reduce the number of independent degrees of freedom of the hand. Baker et al. [1, 2] explore the angles, velocities and accelerations of the finger joints during typing on standard and ergonomic keyboards. Baker et al. [2] found that user hand kinematics adapt to the environment imposed by the keyboard configuration. This work provides a useful reference regarding the dynamic movements occurring during typing and suggests that users of a virtual keyboard will also adjust their behavior based on the keyboard shape. Particularly informative to this work, Fish and Soechting [9] investigated the correlations between finger lengths and between finger orientations during typing tasks. Fish and Soechting [9] found that in most cases there were significant correlations between digits and that the highest was between neighboring digits. They also found that while digits move in coordinated motions, once one was needed to complete a task, it was decoupled from the group and the correlation dropped substantially. This work was however performed on a physical keyboard with passive resistance provided by the keys. This differs from the setup used in this paper, but it is expected that these observations will be relevant because up to the moment a digit contacts a key, its motion is unrestricted, which is true for both setups.

Häger-Ross and Schieber [15] investigated the degree to which human fingers can be moved independently. The objective of this study was to determine whether humans move other digits when attempting to move just one and, if so, quantify this movement. Häger-Ross and Schieber [15] showed that there was significant motion of the digits that were intended to be held stationary. These results agree with the findings of Fish and Soechting [9]. Individuation is used as a metric to indicate how much other digits move when the digit of focus moves while stationarity measures how

little the digit of focus moves when each of the other digits move. The thumb and index finger were found to have the greatest individuation and stationarity from the other digits, followed by the pinky finger, with the middle and ring fingers having the least [9]. This suggest that the middle, ring and pinky digits are more likely to cause coactivations. Feit et al. [8] provide a complementary perspective by identifying common typing behaviors as patterns from an interaction rather than kinematic point of view. Although pursuing a fundamentally different approach to text, Sridhar et al. [27] highlight the issue of coactivations in mid-air text entry. In summary, although finger coactivations is a well recognized issue in text entry there has been limited specific exploration of its characteristics in mid-air virtual typing. Further, there are no established methods for detecting or mitigating spurious observations arising from coactivations.

3 APPROACH

As briefly outlined in section 1, we first seek to understand when and how coactivations occur before attempting to detect and eliminate them. This approach sits in contrast to, for example, a deep-learning strategy that might circumvent the need for understanding the mechanisms that produce coactivations. The motivation behind this strategy is: i) to provide a generalizable characterization of coactivations in mid-air ten finger typing; ii) to avoid the requirement for very large amounts of training data; and iii) to deliver a detection strategy that can be readily adapted to alternative keyboard configurations. This results in an understanding of coactivations that is relevant beyond the specific confines of this study and an approach to detection that is easily realized by others. This investigation therefore involves two distinct stages: Stage 1) understanding when and how coactivations occur; and Stage 2) implementing and evaluating strategies for coactivation detection and elimination.

In Stage 1, we begin by studying the occurrence of coactivations in a recorded dataset [6] of 24 users typing on a mid-air virtual keyboard in VR. The dataset contains precision finger tracking data captured while users transcribe set phrases (170 phrases per user). Users were not given the facility to correct errors when encountered, however, an oracle strategy (detailed in [6]) was used to simulate the effect of typing with high quality auto-corrections. This configuration was used to capture representative data of how users would type under ideal conditions. The dataset was manually reviewed to identify possible characteristics of coactivations. This analysis is reported in section 4. The outcome of Stage 1 is an improved understanding of the nature of coactivations encountered in ten finger mid-air typing.

In Stage 2, the discriminative power of each of the possible characteristics identified in Stage 1 is evaluated to determine their potential as features in a feature vector. The promising features are then used to construct three alternative coactivation detection techniques: naïve Bayes, support vector machine, and neural network. The careful approach taken in Stage 1 ensures that these detectors achieve high accuracy with minimal features. Finally, the best performing model, the neural network, is incorporated into an existing statistical decoder. This provides a demonstration of how coactivation detection can provide reduced error rates in mid-air ten finger typing.

4 UNDERSTANDING COACTIVATIONS

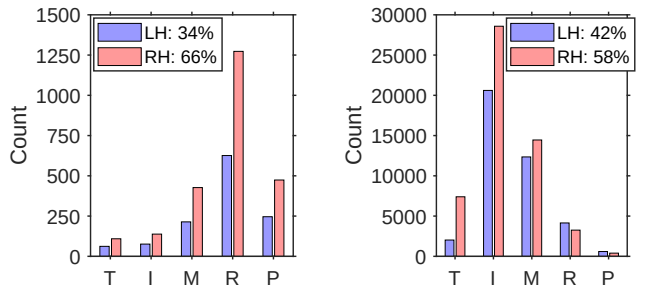
The mid-air ten finger typing dataset was manually reviewed to identify the circumstances in which coactivations occur. We limited this analysis to coactivations specifically causing unintended input observations. This review also provided an initial set of features distinguishing a coactivation from an intended finger movement.

4.1 When and Where do Coactivations Occur?

Coactivations can be divided into two types characterized by the order of the intended and unwanted key selection. If the intended key is selected first, the unwanted key is a post-intended-coactivation and if the unwanted key is selected first is it a pre-intended-coactivation. Post-intended-coactivations made up 3.4% of touch events recorded in the dataset and the majority (92%) of all coactivations. Subsequent references to coactivations imply the combined group of post- and pre-intended-coactivations. However, we limit the scope of coactivation detection and mitigation to post-intended-coactivations since: i) these are the overwhelming majority of coactivations; and ii) post-intended-coactivations can be dealt with immediately in a relatively simple online detection implementation.

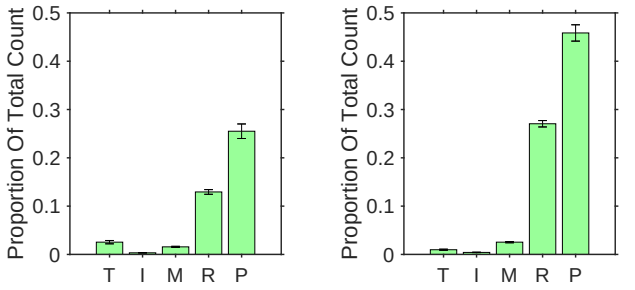
As outlined in section 2, coactivations are caused by the coupled motion of a pair of digits. We hypothesized that the occurrence of coactivations would vary depending on which finger is involved in a key press and, related to this, which key on the layout this press is targeting. The distribution of coactivation counts per digit is shown in Figure 1(a). In Figure 1(a) and throughout we use the following initials for conciseness: (T)humb, (I)ndex, (M)iddle, (R)ing, (P)inky, right hand (RH), left hand (LH). This figure shows that the majority of coactivations observed were caused by the ring digit and not the pinky finger as expected. This figure also shows that the right hand was responsible for two thirds of the coactivations which is consistent with a right handed bias among participants. Figure 1(b) shows a similar hand split for intended counts, which was equated to the underlying hand usage. Normalizing the coactivation count per finger by hand usage yields an approximately equal distribution of coactivations across the fingers for each hand. This indicates that neither hand is more prone to coactivations. The prior probabilities of each digit to cause a coactivation were also calculated and are shown in Figure 2. This figure shows that the ring and pinky fingers produce a much higher proportion of coactivations than the other digits.

The normalized proportion of ordered pairs of digits containing a post-intended-coactivation is shown in Figure 3 along with their standard errors. There are 40 pairs in total, each digit followed by another digit on the same hand. Coactivations, by definition, cannot be caused by the same digit although this case is included in the graph for clarity of presentation. Figure 3 shows that the majority of post-intended-coactivations are caused by a digit less dominant to the intended one. For example, coactivations of the middle, ring and pinky occur after the intended use of the index digit, coactivations of the ring and pinky occur after intended use of the middle digit, and so on. It also shows that the intended use of the middle digit followed by a post-intended-coactivation of the ring digit makes up a significant proportion of the times that pair is used, consistent with observations made by Häger-Ross and Schieber [15].



(a) Count of coactivations. (b) Count of intended touch events.

Figure 1: Distribution of intended and coactivated touch events by each digit of each hand.



(a) Left hand. (b) Right hand.

Figure 2: Proportion of all touch events for a given digit that were coactivations.

The distribution of coactivations across the layout was analyzed by plotting the raw coactivation counts and the proportion of touches on a key that were coactivations as heat-maps, shown in Figures 4(a) and 4(b), respectively. These figures reveal that coactivations occur most frequently at the sides of the keyboard closest to the ring and pinky digits. Also, there are higher counts on the right side, consistent with the hand usage shown in Figure 1. Figure 4(b) shows the majority of the edge key events were coactivations. These results offer an insight into the prior probability for each key to be coactivated.

4.2 Features Indicative of a Coactivation

Four features hypothesized to be indicative of a coactivation were identified through the dataset inspection exercise and with reference to the literature examining coupled finger movement. These are: i) digit depth of travel; ii) digit velocity; iii) inter-key interval; and iv) averaged digit velocity correlation. Each of these features are briefly described below.

4.2.1 Digit Depth. Coactivating digits were observed to typically trail behind the intended digit. There were many cases in the dataset where, upon a touch event, the coactivating digit would reach a shallower depth before both digits were retracted.

4.2.2 Digit Velocity. The intended activation of a digit produces a motion path with a given velocity. As the motion of coactivating digits is not intended, we hypothesize that these digits move with a lower velocity than the intended digit, particularly at the point of intersection with the virtual keyboard plane.

4.2.3 Inter-Key Interval. Due to inherent limitations in motor skills and the speed at which one can process information, there is an upper bound on realistic entry rates. Dhakal et al. [5] found the average fast typist had an interval between touch events of around 120 ms and a lower bound of around 60 ms. For a typical user, this suggests that touch events within 120 ms are likely to be unintended and ones within 60 ms are almost certainly unintended. The interval between touch events is therefore a likely indicator of coactivations.

4.2.4 Averaged Digit Velocity Correlation. Coupled motion of digits causing a coactivation was observed to exhibit similar motion paths between digits. This observation is consistent with Fish and Soechting [9], who found that while typing there is a reasonable degree of correlated digit movement when the digits were not engaged in a keypress task. Once a digit was selected for a task, its correlation to the other digits decreased substantially. It is reasonable to assume that this decoupling may not always occur in the virtual keyboard setting given the lack of physical resistance. This motivates consideration of correlation in the average digit velocity profile of two digits as an indicator of coactivations.

5 FEATURE EVALUATION

The four potential features identified in section 4.2 are now analyzed by plotting histograms of the intended and coactivated touch events. This distribution analysis was performed with half of the dataset (i.e. data from 12 of the 24 participants) to maintain the remainder as an unseen validation set. The intended and coactivated distributions are compared to determine the effectiveness of each as a discriminative feature. Note that in subsequent descriptions, the keyboard frame is defined as a fixed frame of reference where the Z-axis is normal to the keyboard plane such that z is zero at the keyboard surface and is increasingly positive at greater depth into the keyboard (further from the user).

5.1 Digit Depth

The depth of a touch event was computed based on the maximum depth of travel immediately following entry within the virtual keyboard plane. The depth of the touch event was taken as the deepest point of a continuous downward movement from the detection plane. The resulting distributions of intended and coactivated touch event depths are shown in Figure 5(a). Figure 5(a) shows that the mean of each distribution occurs at a different depth. The mean intended touch event depth and its standard deviation was 15 ± 7 mm whereas the average coactivation depth was much shallower, 5 ± 3 mm. This is a promising result indicating that digit depth may be an effective feature.

Due to the coupled motion of the digits, it is possible that deeper intended touch events would be more likely to pull a neighboring finger down into the detecting plane, producing a coactivation on the next touch event. To examine this, the depth of the previous digit was plotted and classed according to the current touch event. This

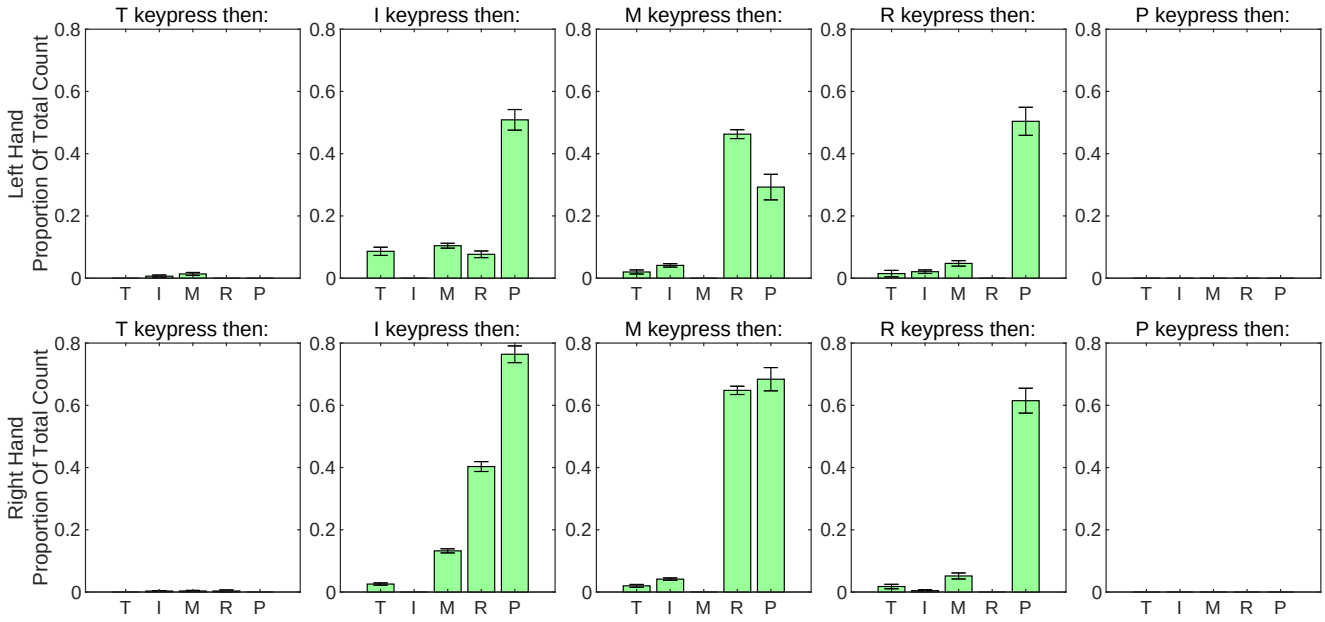
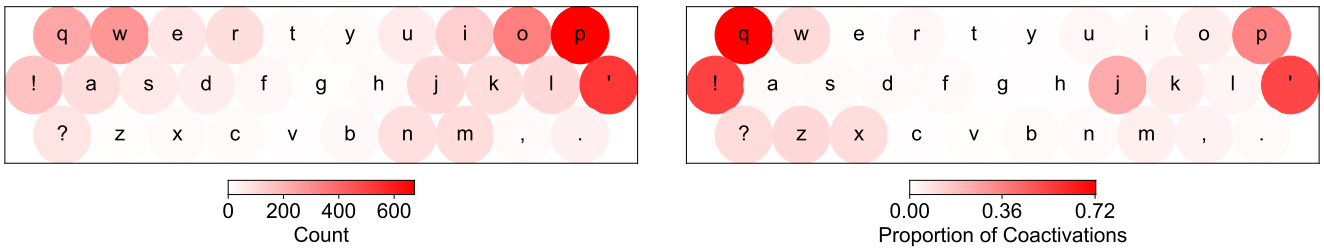


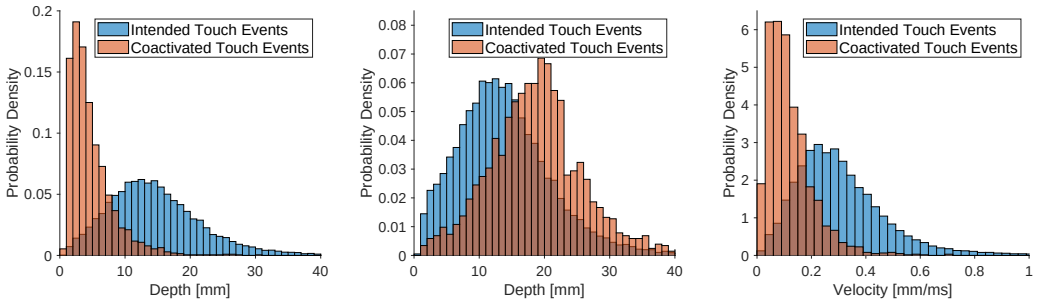
Figure 3: Proportion of all touch events for a given pair of digits that were coactivations. Digit pairings with a total count of less than 100 have been removed.



(a) Total coactivation counts.

(b) Proportion of touches that were coactivations.

Figure 4: Coactivation count distribution over the letters of the keyboard.

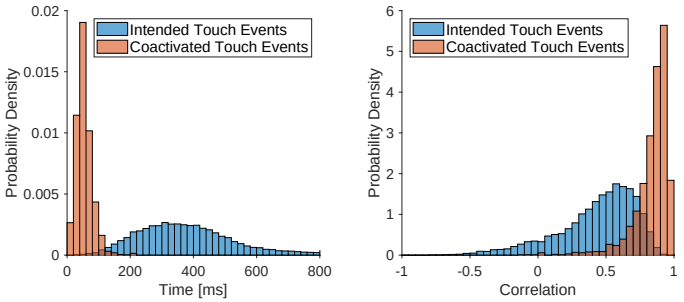


(a) Current digit depth feature.

(b) Previous digit depth feature.

(c) Digit velocity feature.

Figure 5: Depth and velocity feature results for the combined dataset of participant 1 to participant 12.



(a) Inter-key interval feature. (b) Digit correlation.

Figure 6: Inter-key and correlation feature results for the combined dataset of participant 1 to participant 12.

is shown in Figure 5(b). Figure 5(b) shows only a slight difference between the distribution means but has the unique property of only relying on data from the previous touch event.

5.2 Digit Velocity

The touch event velocity was calculated as the average velocity of the activating digit from approximately 30 ms before to approximately 30 ms after the touch event. This range was chosen using the following logic. Dhakal et al. [5] found the average interval between keypresses was 239 ms. In this time, four movements need to take place. The activating digit must move from its rest height to the detection plane, from there to its max depth, back to the detection plane, and then back to the rest height. If each of these actions take approximately the same amount of time, then the full downwards motion will occur from about 60 ms before a keypress to about 60 ms after it. Taking the center half of this motion to remove the portions where the digit was accelerating and decelerating, results in a 60 ms window over which to average. As before, this feature was calculated for half the dataset and plotted as histograms shown in Figure 5(c). This feature was evaluated in the keyboard frame of reference. This figure shows a clear separation between both the peaks and means of the distributions, which is a promising result. The average intended and coactivated touch event velocities were 0.31 ± 0.17 mm/ms and 0.12 ± 0.08 mm/ms respectively.

5.3 Inter-Key Interval

The interval assigned to each coactivated touch event was the time between that event and the coactivation's corresponding intended touch event. Intended touch events can cause multiple coactivations and so this interval may not necessarily be calculated between consecutive events. The interval assigned to intended events was the time between that event and the previous intended touch event. The resulting distributions are shown in Figure 6(a). Figure 6(a) shows the greatest separation of the distributions among the features examined. However, it does not identify which of the two touch events, used in its calculation, is the coactivation. This feature must therefore be combined with others to distinguish the coactivation.

5.4 Average Digit Velocity Correlation

The average velocity correlation assigned to each touch event was calculated between pairs selected in the same way as for the inter-key interval feature. The correlation was calculated as the average dot product of the digits' velocity vectors, normalized by the magnitude of the vectors. The average was taken from 50 ms before the first touch event to 50 ms after the second touch event. The times were chosen using a similar logic to the velocity feature. The motion to select a key begins approximately 60 ms before the touch event and ends approximately 60 ms after it. The first and last 10 ms of the motion were removed to reduce the part of the motion where there is little to no movement. Using a window that extends from before the first touch event to after the second ensured the motion of the two events was captured. The equation used is given below, where $\mathbf{X}^j_{T_i}$ is the three-dimensional position of digit j at time T_i:

$$Corr = \frac{\mathbf{V}_{D_1} \cdot \mathbf{V}_{D_2}}{||\mathbf{V}_{D_1}|| \, ||\mathbf{V}_{D_2}||} \qquad \text{where} \quad \mathbf{V}_{D_j} = \frac{\mathbf{X}^j_{T_2} - \mathbf{X}^j_{T_1}}{T_2 - T_1}. \quad (1)$$

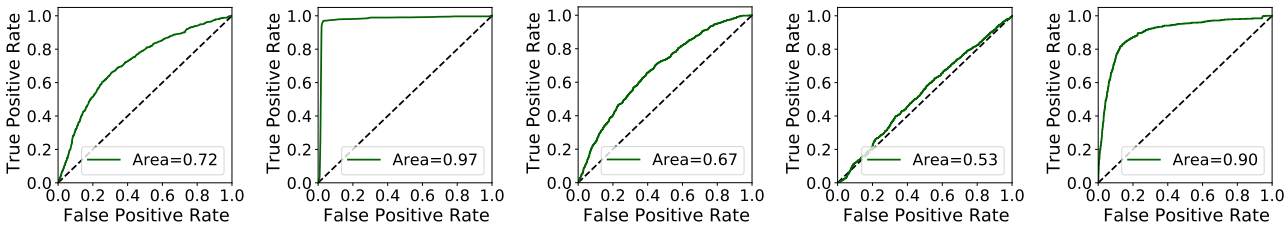
This feature was also evaluated in the keyboard frame. The resulting distribution is shown in Figure 6(b). The mean and standard deviation of the intended and coactivated distributions were 0.43 ± 0.29 and 0.84 ± 0.13 respectively. This shows that on average the velocity of a coactivating digit was highly correlated with its corresponding intended digit. It also shows, however, that there is a non-insignificant degree of correlation between all digits.

6 DETECTING COACTIVATIONS

In this section, we describe the design of a coactivation detection technique implemented using three alternative models. We begin with an overview of the chosen features before describing the model selection and training procedure in section 6.2. Finally, we report on the cross-validated accuracy evaluation of the developed models in section 6.3.

6.1 Feature Selection

From among the features tested in section 5, the current and previous digit depth and velocity, inter-key interval, and average correlation features were identified as worthy elements of the



(a) Previous depth feature. (b) Interval feature. (c) Current velocity feature. (d) Prev. velocity feature. (e) Vel. correlation feature.

Figure 7: ROC curves for potential feature vector elements.

feature vector. These features combine data from multiple time steps, extract useful information and reduce the size of the feature vector.

One problem with some of these features are their dependence on information occurring after the current touch point to be classified. Although using all the features listed above is likely to produce the best performance, for a model to run easily online, it should run in real time with no such dependencies. The depth of the current touch event relied on the current digit's z position several time steps after the event had occurred and therefore it was removed.

The average correlation, shown in subsection 5.4, also relied on data from time steps recorded after the touch event. The extra 50 ms of data included after the second touch event was used to achieve a more accurate average. The correlation calculation was altered to remove these extra time steps. The extra time steps included before the first touch event were also removed to reduce the complexity of implementation. Removing the current depth feature and modifying the calculation of the correlation feature, leaves a total of five input features with no post-event information dependencies: the current and previous digit velocity, the depth of the previous touch event, the time interval, and the average velocity correlation. These alterations limit the features, and therefore the feature vector, to only data prior to the current touch event, allowing the model to be more easily incorporated into an online system. Further evaluation of these five features in subsection 6.2.1 reveals that the previous digit velocity performs poorly and is subsequently removed from the final feature vector, leaving four features implemented into the final models.

6.2 Model Selection

Three alternative models were selected for evaluation. These were a naïve Bayes model, a support vector machine and a neural network. Each of these models was trained and tested using cross-validation. We partitioned the dataset into training and test sets for K-fold cross-validation. Effort was taken to assign participants into representative groupings. There was high variation between the users in the dataset in terms of coactivation counts and typing speed. Stratified folds were therefore assembled based on two criteria. First, participants were grouped to minimize the difference in total coactivation counts per group. An exhaustive search found several hundred groupings where the maximum difference between totals was one. Second, final group assignments were made based on the objective of minimizing the difference in average words per minute across groups.

The design of each coactivation detection model is now summarized in the following subsections. Note that we describe the design of the neural network first since we leverage its ability to isolate the influence of individual features to further refine the feature vector used by the other models.

6.2.1 Neural Network. Although the task of detecting coactivations is inherently a time dependent problem, it was decided to address the problem with a time independent solution. This choice was motivated by the fact that time dependent solutions are often more complex and more difficult to train, and thus demand more data, which is expensive and difficult to collect for this problem. In preliminary results, the time independent features investigated also proved effective in identifying coactivations and therefore we pursued a time independent solution.

The architecture chosen for the neural network comprises the following layers: 32 node dense with batch normalization, ReLU activation and 20% dropout, followed by another 32 node dense with batch normalization, ReLU activation and 20% dropout, then a 16 node dense with batch normalization, ReLU activation and 20% dropout, and finally a 1 node dense with a sigmoid activation function. The network therefore has just less than 2,000 trainable parameters. The model was trained on three of the groups, with the fourth used for testing and validation. To train, all coactivations and an equal number of intended touch events were selected from each participant in the training group. This ensured the model was trained on an equal number of positive and negative cases. As a result, approximately 7,500 data points were used in the training of the model's 2,000 parameters—3.75 data points per parameter. We used 32 nodes per layer to maximize the complexity of the relationships that can be learned while maintaining a sufficient number of training points per parameter [23]. The specific choice of 32 nodes stems from the convention of assigning node counts at powers of base two.

Calibration of the output is also an important consideration given the objective of integrating the model with a statistical decoder. A binary cross-entropy loss function was selected given its suitability to this type of problem. The cross-entropy loss function is a probabilistic loss and is minimized when the model's predictions approximate the ground truth conditional distribution [13].

To test the effectiveness of each feature, the feature vector was reduced to a single feature at a time for testing. The model was trained and tested using cross-validation. A confusion matrix was created for each fold of the validation and used to calculate the mean and standard deviation of the accuracy, precision, recall and

Table 1: Table of metrics showing the performance of various features (± 1 standard deviation).

Feature	Accuracy	Precision	Recall	AUC
Previous Digit Depth	0.58 ± 0.07	0.20 ± 0.07	0.64 ± 0.11	0.64 ± 0.05
Current Digit Velocity	0.61 ± 0.03	0.22 ± 0.03	0.72 ± 0.01	0.70 ± 0.02
Previous Digit Velocity	0.54 ± 0.16	0.14 ± 0.04	0.47 ± 0.21	0.50 ± 0.03
Interval	0.96 ± 0.02	0.80 ± 0.11	0.96 ± 0.01	0.98 ± 0.01
Correlation	0.85 ± 0.02	0.48 ± 0.07	0.76 ± 0.05	0.88 ± 0.01

Table 2: Metrics showing the performance of the three models tested (± 1 standard deviation).

Model	Accuracy	Precision	Recall	F1 Score	AUC
Neural Net	0.978 ± 0.011	0.882 ± 0.072	0.968 ± 0.009	0.921 ± 0.044	0.990 ± 0.006
SVM	0.973 ± 0.011	0.854 ± 0.065	0.971 ± 0.010	0.907 ± 0.042	0.985 ± 0.006
Naïve Bayes	0.976 ± 0.005	0.922 ± 0.028	0.902 ± 0.021	0.912 ± 0.023	0.984 ± 0.008

F1 score. These evaluation metrics are given for each feature in Table 1. The ROC curves for each feature from the first fold of the cross-validation are shown in Figure 7.

An examination of the evaluation metrics presented in Table 1 and the ROC curves reveals that the interval feature achieves the best performance while the previous key velocity feature performs the worst. The remaining features demonstrate good performance. As a result, it was decided to remove the previous key velocity feature from the feature vector. This was done as its ROC curve showed it contributed very little to the classification prediction and reducing the feature vector would increase the data points per parameter, potentially improving the fit of the other features. The final feature vector is therefore composed of the following features: previous depth, current velocity in the keyboard frame, interval, and average velocity correlation in the keyboard frame.

6.2.2 Naïve Bayes. The naïve Bayes model was chosen as it is a simple, commonly used classifier. The probability of a coactivation given the values of the four selected features was found using Bayes theorem in combination with the empirical distributions. The conditional distributions of the features were found by first normalizing the histograms presented in section 5.

6.2.3 Support Vector Machine. The final model selected was a calibrated support vector machine (SVM). SVMs are well known and well understood classifiers, which serve as reliable baselines for evaluating classification performance. A linear kernel was chosen for simplicity and the SVM was calibrated using Platt scaling.

6.3 Detection Accuracy

To evaluate coactivation detection accuracy, the three models were trained on the chosen features and tested using the same cross-validation splits to ensure consistency. This was done using cross-validation in the same way the individual features were tested using the neural network described above. A confusion matrix was created for each fold of the validation and used to calculate the mean and standard deviation of the accuracy, precision, recall and F1 score. This was repeated for each model and the resulting evaluation metrics are given in Table 2.

All models show a high level of accuracy in detecting coactivations and none of the models have significantly higher accuracy than any of the others. The naïve Bayes model's precision is higher than the SVM, but its recall is lower. This suggests that the SVM and naïve Bayes trade off between false positives and false negatives. The neural network's precision and recall sits between these two. The neural network achieved the highest F1 score and was therefore chosen as the model for subsequent integration and evaluation with the decoder.

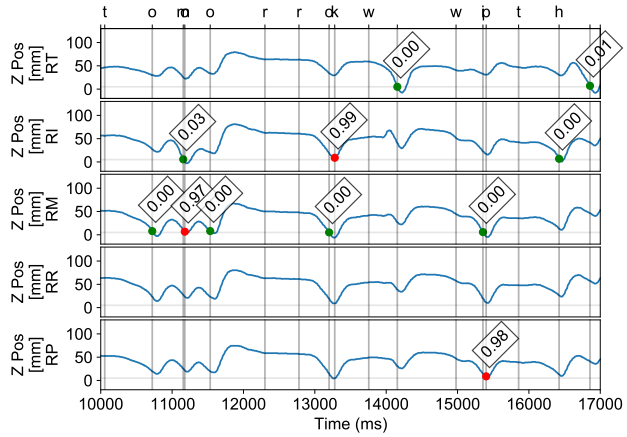
7 USING COACTIVATION DETECTION IN STATISTICAL DECODING

To validate the utility of the developed coactivation detection functionality, we integrated it as a module within an existing probabilistic text input decoder. As an illustration of the behavior of the module, independent of the decoder, Figure 8(a) demonstrates how discrete input events are scored based on a continuous stream of finger tracking data. Figure 8(a) plots the z offset of the right hand digits only during the entry of "tomorrow with". The vertical lines indicate when a letter (shown at the top of the plot) was activated. The markers indicate the ground truth of whether a touch event was intended (green), or a coactivation (red). A vertical line with no corresponding marker indicates the letter was activated by the left hand. Beside each marker is the system's coactivation prediction for the touch event. Figure 8(a) shows the original text typed, "tomoorrokw wipth" and shows the system correctly identifying the coactivations "o", "k" and "p". Removing these letters would return the correct phrase "tomorrow with". This figure also highlights that the system can correctly identify coactivations by different digits, as the features are digit independent.

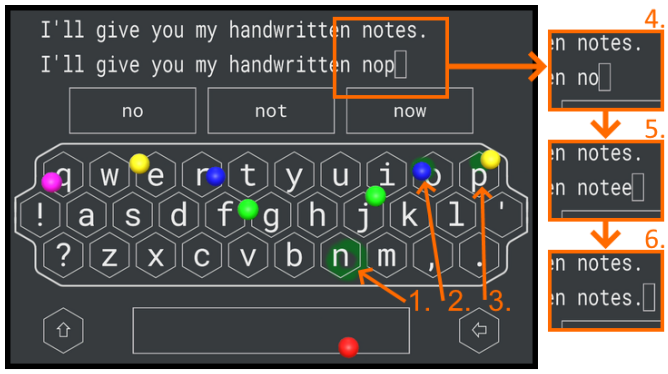
The promising behavior observed from the isolated detection module and illustrated in Figure 8(a) motivated its complete integration with an existing statistical decoder. The decoder integration operates as follows: at each touch event the probability that the event is a coactivation is calculated. If the probability is greater than a certain threshold the character is removed from the input field. The system can also be configured to remove this observation from the set passed to the decoder. If the character is not removed, the

Table 3: Resulting error rates under different decoder configurations (± 1 standard deviation).

Error Metric	Full Dataset	Coactivation Subset
CER without auto-correct	10.7 ± 0.5%	13.8 ± 1.0%
CER with auto-correct but without coactivation detection	8.7 ± 0.5%	11.0 ± 1.1%
CER with auto-correct and coactivation detection	7.8 ± 0.7%	9.0 ± 1.1%
Relaxed Error Rate	7.7 ± 0.2%	10.7 ± 0.5%



(a) The z (mm) offset of the right hand digits plotted against time, showing model coactivation predictions (0: intended, 1: coactivated) and ground truths (green: intended, red: coactivated).



(b) Illustrated operation of the decoder's coactivation detection and auto-correction components. 1: Intended 'n'. 2: Intended 'o'. 3: Coactivated 'p'. 4: 'p' removed by coactivation module. 5: 'tee' entered by user. 6: Auto-correction of 'notee' to desired 'notes'.

Figure 8: Sample evaluations of the decoder with the integrated coactivation detection.

weighted coactivation probability is combined with the likelihood that the character is an insertion error. This is done for each observation until a space or other word delimiter is activated to force a decode. We are interested in the effect that coactivation detection can have on reducing the character error rate (CER) when typing. The CER is calculated based on how many insertion, deletion or substitution corrections are needed to convert the final text into the stimulus text the user was trying to replicate. It is normalized by the length of the stimulus text. Using the same cross-validation partitioning described in subsection 6.2, we determine appropriate hyperparameter values from the training sets and evaluate the resulting CER with the corresponding test sets. Approximate hyperparameters were first found using Bayesian optimization, followed by grid search to fine tune the parameters.

7.1 Eliminating Errors in Typing

The dataset upon which the models are trained and evaluated with cross-validation involved typing using a simulated decoder with no facility for error correction. Therefore subsequent error values should be interpreted as uncorrected error rates. The mean CER and standard deviation of the final hyperparameter setting of the decoder across the four folds of the cross-validation were calculated with and without the coactivation module enabled. These values are summarized in Table 3. Also included in Table 3 is the CER without auto-correct and the relaxed error rate, a metric introduced by Dudley et al. [6] based on a distance tolerance attached to each target key that can only be calculated if the system has complete knowledge of what the user is trying to type. The CER without auto-correct reflects direct interpretation of observations based on the closest key to the touch point and therefore serves as a pessimistic baseline. By contrast, the relaxed error rate reflects the performance under the oracle decoder, which allows for a high degree of touch inaccuracy, and therefore serves as an optimistic baseline. The Full Dataset column lists the results for all typed phrases (i.e. 1,020 phrases per fold). The Coactivation Subset column reports the subset of phrases where coactivations are present (n = [531, 547, 519, 659] across the four folds).

The results show that with the coactivation module enabled in the statistical decoder a further reduction in character error rate by approximately 10% relative and 0.9% absolute can be achieved. This improvement is more significant when the dataset is restricted to just samples containing coactivations, achieving character error rate reductions of approximately 18% relative and 2.0% absolute. Figure 8(b) illustrates the keyboard behavior with the integrated coactivation module. Figure 8(b) shows the keyboard state just as the stream of characters, 'nop', have been entered using the right index (1.), middle (2.) and ring (3.) fingers. The user is trying to type 'notes' but has caused a coactivation with their ring finger and inserted the 'p' into the input field. The coactivation module subsequently identifies and removes the coactivated 'p' character (4.). The remaining characters, 'tee', are entered (5.), before the

literal input, 'notee', is corrected to the desired word, 'notes', by the decoder (6.) upon typing the period.

8 DISCUSSION

This investigation has shown that identification and removal of coactivations from a text input stream can be completed with simple lightweight features. Four features have been identified that can aid the detection of coactivations. Combined, these features can achieve a detection accuracy exceeding 97%. The module integrated into the input decoder also demonstrates how coactivation detection can effectively reduce typing error rates. The simplest and most effective feature is the interval between touch events. This feature suggests that a naïve strategy for mitigating coactivations could rely on simply enforcing a minimum time (50–100 ms) before the next letter can be selected.

Despite the immediate demonstrated efficacy of the coactivation detection strategy presented, limitations and opportunities for future work remain. First, the model is limited by its simplicity and relatively small number of features. This was a design choice based on the size of the dataset and the challenge in collecting a large representative dataset for this application context. As a result, most of the raw digits' motion is unavailable to the model or is only available in a condensed form. While it would be theoretically possible to overcome the data sparseness issue with representative synthetic coactivation data, this would necessitate a generative model capturing both physical and neuromuscular behaviors encountered while typing. This would, however, be a challenging research project in its own right. Second, as described in section 4, we exclude pre-intended-coactivations from consideration due to the pursuit of streamlined online detection. Although pre-intended-coactivations are a very small proportion of all coactivations, there is potential value in robustly detecting all spurious input events. Related to this point, the choice of features was also constrained to those available up until the current time step (see section 6.1). More advanced strategies that continually reassess past potential coactivations as new data streams in are feasible but would require more complex accounting and event handling functionality. As a result, they would therefore likely demand a tighter coupling between the coactivation detection module and the decoding framework. A specific design objective in this work was to avoid a tight coupling of components, as this increases the complexity of the system design. However, a more flexible strategy for handling coactivation detection that can propagate backwards to recent events is a promising avenue for future work.

In general, it would likely be fruitful to extend the model to include prior probabilities of digits causing coactivations and characters becoming coactivated. The potential contribution of this prior information is clearly illustrated in Figure 4 and could be readily incorporated. Finally, the current pandemic has prevented the execution of a user study to evaluate the potential benefit of coactivation detection and mitigation during 'live' text entry.

9 CONCLUSIONS

In this paper, we presented a detailed characterization of coactivations encountered in ten finger mid-air typing in VR. The most informative features of coactivations were found to be: inter-key interval, key depth, key velocity, and digit velocity correlation. We demonstrated that coactivations can indeed be effectively detected. The final models successfully identified 97 ± 1% of coactivations while only incorrectly labeling 2 ± 1% of intended touch points as coactivations. As 4% of characters typed were coactivations, this equates to a 2 ± 1% reduction in errors. Finally, we demonstrated how coactivation detection can be integrated into a statistical decoding framework to achieve a reduction in uncorrected CER of approximately 10% relative and 0.9% absolute.

ACKNOWLEDGMENTS

This work was supported by Facebook Reality Labs. J. J. Dudley and P. O. Kristensson were also supported in part by EPSRC (grants EP/R004471/1 and EP/S027432/1). Supporting data for this publication is available at https://doi.org/10.17863/CAM.63030.

REFERENCES

[1] Nancy A. Baker, Rakié Cham, Erin Hale Cidboy, James Cook, and Mark S. Redfern. 2007. Kinematics of the fingers and hands during computer keyboard use. *Clinical Biomechanics* 22, 1 (2007), 34 – 43. https://doi.org/10.1016/j.clinbiomech.2006.08.008

[2] Nancy A. Baker, Rakié Cham, Erin Hale, James Cook, and Mark S. Redfern. 2007. Digit kinematics during typing with standard and ergonomic keyboard configurations. *International Journal of Industrial Ergonomics* 37, 4 (2007), 345 – 355. https://doi.org/10.1016/j.ergon.2006.12.004

[3] Garrett Benoit, G. Michael Poor, and Alvin Jude. 2017. Bimanual Word Gesture Keyboards for Mid-air Gestures. In *Proceedings of the 2017 CHI Conference Extended Abstracts on Human Factors in Computing Systems (CHI EA '17)*. Association for Computing Machinery, Denver, Colorado, USA, 1500–1507. https://doi.org/10.1145/3027063.3053137

[4] Doug A. Bowman, Christopher J. Rhoton, and Marcio S. Pinho. 2002. Text Input Techniques for Immersive Virtual Environments: An Empirical Comparison. *Proceedings of the Human Factors and Ergonomics Society Annual Meeting* 46, 26 (2002), 2154–2158. https://doi.org/10.1177/154193120204602611

[5] Vivek Dhakal, Anna Maria Feit, Per Ola Kristensson, and Antti Oulasvirta. 2018. Observations on Typing from 136 Million Keystrokes. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems* (Montreal QC, Canada) *(CHI '18)*. Association for Computing Machinery, New York, NY, USA, 1–12. https://doi.org/10.1145/3173574.3174220

[6] John J. 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)*. 289–300. https://doi.org/10.1109/ISMAR.2019.00027

[7] John J. Dudley, Keith Vertanen, and Per Ola Kristensson. 2018. Fast and Precise Touch-Based Text Entry for Head-Mounted Augmented Reality with Variable Occlusion. *ACM Trans. Comput.-Hum. Interact.* 25, 6, Article 30 (Dec. 2018), 40 pages. https://doi.org/10.1145/3232163

[8] Anna Maria Feit, Daryl Weir, and Antti Oulasvirta. 2016. How We Type: Movement Strategies and Performance in Everyday Typing. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems (CHI '16)*. ACM, New York, NY, USA, 4262–4273. https://doi.org/10.1145/2858036.2858233

[9] Jack T. Fish and John F. Soechting. 1992. Synergistic finger movements in a skilled motor task. *Experimental Brain Research* 91 (1992), 327–334.

[10] Jens Grubert, Lukas Witzani, Eyal Ofek, Michel Pahud, Matthias Kranz, and Per Ola Kristensson. 2018. Effects of Hand Representations for Typing in Virtual Reality. IEEE, 151–158.

[11] Jens Grubert, Lukas Witzani, Eyal Ofek, Michel Pahud, Matthias Kranz, and Per Ola Kristensson. 2018. Effects of Hand Representations for Typing in Virtual Reality. In *2018 IEEE Conference on Virtual Reality and 3D User Interfaces (VR)*. 151–158. https://doi.org/10.1109/VR.2018.8446250

[12] Jens Grubert, Lukas Witzani, Eyal Ofek, Michel Pahud, Matthias Kranz, and Per Ola Kristensson. 2018. Text Entry in Immersive Head-Mounted Display-Based Virtual Reality Using Standard Keyboards. IEEE, 159–166.

[13] Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q. Weinberger. 2017. On Calibration of Modern Neural Networks. arXiv:1706.04599 [cs.LG]

[14] Aakar Gupta, Cheng Ji, Hui-Shyong Yeo, Aaron Quigley, and Daniel Vogel. 2019. RotoSwype: Word-Gesture Typing using a Ring. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems (CHI '19)*. Association for Computing Machinery, Glasgow, Scotland Uk, 1–12. https://doi.org/10.1145/3290605.3300244

[15] Charlotte Häger-Ross and Marc H. Schieber. 2000. Quantifying the Independence of Human Finger Movements: Comparisons of Digits, Hands, and Movement Frequencies. *Journal of Neuroscience* 20, 22 (2000), 8542–8550. https://doi.org/10.1523/JNEUROSCI.20-22-08542.2000

[16] InformedHealth.org [Internet]. Cologne, Germany: Institute for Quality and Efficiency in Health Care (IQWiG). 2018. *How do hands work?* https://www.ncbi.nlm.nih.gov/books/NBK279362/

[17] Pascal Knierim, Valentin Schwind, Anna Maria Feit, Florian Nieuwenhuizen, and Niels Henze. 2018. Physical Keyboards in Virtual Reality: Analysis of Typing Performance and Effects of Avatar Hands. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems (CHI '18)*. ACM, New York, NY, USA, 345:1–345:9. https://doi.org/10.1145/3173574.3173919

[18] Anders Markussen, Mikkel R. Jakobsen, and Kasper Hornbaek. 2013. Selection-Based Mid-Air Text Entry on Large Displays. In *Human-Computer Interaction – INTERACT 2013: 14th IFIP TC 13 International Conference, Cape Town, South Africa, September 2-6, 2013, Proceedings, Part I*, Paula Kotzé, Gary Marsden, Gitte Lindgaard, Janet Wesson, and Marco Winckler (Eds.). Springer Berlin Heidelberg, Berlin, Heidelberg, 401–418. https://doi.org/10.1007/978-3-642-40483-2_28

[19] Anders Markussen, Mikkel Rønne Jakobsen, and Kasper Hornbaek. 2014. Vulture: A Mid-air Word-gesture Keyboard. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '14)*. ACM, New York, NY, USA, 1073–1082. https://doi.org/10.1145/2556288.2556964

[20] Keith L. Moore, Arthur F. Dalley, and A.M.R. Agur. 2013. *Clinically Oriented Anatomy*. Wolters Kluwer Health/Lippincott Williams & Wilkins.

[21] Tao Ni, Doug Bowman, and Chris North. 2011. AirStroke: Bringing Unistroke Text Entry to Freehand Gesture Interfaces. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '11)*. ACM, New York, NY, USA, 2473–2476. https://doi.org/10.1145/1978942.1979303

[22] Taihei Ogitani, Yoshitaka Arahori, Yusuke Shinyama, and Katsuhiko Gondow. 2018. Space Saving Text Input Method for Head Mounted Display with Virtual 12-key Keyboard. In *2018 IEEE 32nd International Conference on Advanced Information Networking and Applications (AINA)*. 342–349. https://doi.org/10.1109/AINA.2018.00059

[23] Kevin L. Priddy and Paul E. Keller. 2005. *Artificial Neural Networks: An Introduction*. SPIE Press.

[24] Marc H. Schieber and Marco Santello. 2004. Hand function: peripheral and central constraints on performance. *Journal of Applied Physiology* 96, 6 (2004), 2293–2300. https://doi.org/10.1152/japplphysiol.01063.2003 PMID: 15133016.

[25] Garth Shoemaker, Leah Findlater, Jessica Q. Dawson, and Kellogg S. Booth. 2009. Mid-air text input techniques for very large wall displays. In *Proceedings of Graphics Interface 2009 (GI '09)*. Canadian Information Processing Society, Kelowna, British Columbia, Canada, 231–238.

[26] John F. Soechting and Martha Flanders. 1997. Flexibility and Repeatability of Finger Movements During Typing: Analysis of Multiple Degrees of Freedom. *Journal of Computational Neuroscience* 4, 1 (1997), 29–46.

[27] Srinath Sridhar, Anna Maria Feit, Christian Theobalt, and Antti Oulasvirta. 2015. Investigating the Dexterity of Multi-Finger Input for Mid-Air Text Entry. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems (CHI '15)*. ACM, New York, NY, USA, 3643–3652. https://doi.org/10.1145/2702123.2702136

[28] Craig L. Taylor and Robert J. Schwarz. 1955. The anatomy and mechanics of the human hand. *Artificial limbs* 2, 2 (1955), 22–35.

[29] Naoki Yanagihara, Buntarou Shizuki, and Shin Takahashi. 2019. Text Entry Method for Immersive Virtual Environments Using Curved Keyboard. In *25th ACM Symposium on Virtual Reality Software and Technology (VRST '19)*. Association for Computing Machinery, Parramatta, NSW, Australia, 1–2. https://doi.org/10.1145/3359996.3365026

[30] Xin Yi, Chun Yu, Mingrui Zhang, Sida Gao, Ke Sun, and Yuanchun Shi. 2015. ATK: Enabling Ten-Finger Freehand Typing in Air Based on 3D Hand Tracking Data. In *Proceedings of the 28th Annual ACM Symposium on User Interface Software & Technology (UIST '15)*. ACM, New York, NY, USA, 539–548. https://doi.org/10.1145/2807442.2807504

[31] Difeng Yu, Kaixuan Fan, Heng Zhang, Diego Monteiro, Wenge Xu, and Hai-Ning Liang. 2018. PizzaText: Text Entry for Virtual Reality Systems Using Dual Thumbsticks. *IEEE Transactions on Visualization and Computer Graphics* 24, 11 (Nov. 2018), 2927–2935. https://doi.org/10.1109/TVCG.2018.2868581