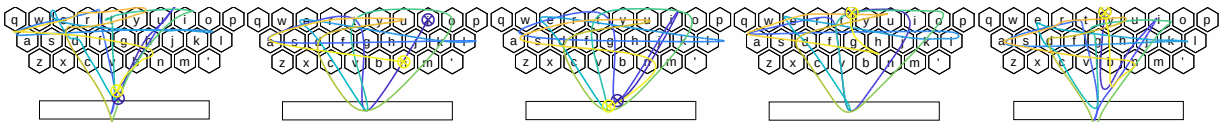
Simulating Realistic Human Motion Trajectories of Mid-Air Gesture Typing

Junxiao Shen* University of Cambridge
John Dudley† University of Cambridge
Per Ola Kristensson‡ University of Cambridge



(a) Real (b) Jerk-Minimization (c) RNN (d) GAN-Transfer (e) GAN-Imitation

Figure 1: Example traces for the phrase, 'I talked to Duran'. Trace start is shown by the blue 'X' and then the trace transitions through green and finishes at the yellow 'X'. a) Real trajectories. b) Trajectories synthesized from the Jerk-Minimization model. c) Trajectories synthesized from the Recurrent Neural Network (RNN)-based generative model. d) Trajectories synthesized from the GAN-based generative model in the *Transfer* setting, where the style is transferred from the original trajectories to trajectories that simply connect the key centers for the corresponding phrase. e) Trajectories synthesized from the GAN-based generative model in the *Imitation* setting, where the style is transferred within the original dataset, such that different variants of original trajectories are produced.

Abstract

The eventual success of many AR and VR intelligent interactive systems relies on the ability to collect user motion data at large scale. Realistic simulation of human motion trajectories is a potential solution to this problem. Simulated user motion data can facilitate prototyping and speed up the design process. There are also potential benefits in augmenting training data for deep learning-based AR/VR applications to improve performance. However, the generation of realistic motion data is nontrivial. In this paper, we examine the specific challenge of simulating index finger movement data to inform mid-air gesture keyboard design. The mid-air gesture keyboard is deployed on an optical see-through display that allows the user to enter text by articulating word gesture patterns with their physical index finger in the vicinity of a visualized keyboard layout. We propose and compare four different approaches to simulating this type of motion data, including a Jerk-Minimization model, a Recurrent Neural Network (RNN)-based generative model, and a Generative Adversarial Network (GAN)-based model with two modes: style transfer and data alteration. We also introduce a procedure for validating the quality of the generated trajectories in terms of realism and diversity. The GAN-based model shows significant potential for generating synthetic motion trajectories to facilitate design and deep learning for advanced gesture keyboards deployed in AR and VR.

Index Terms: Human-centered computing—Human computer interaction (HCI)—Interaction techniques—Text input;

1 Introduction

Human motion analysis and simulation is a popular research topic in the Augmented Reality (AR) and Virtual Reality (VR) community due to its varied potential applications, including: i) real-time avatar animation and control [26, 36]; ii) smooth human-robot interaction [3]; iii) sports training and analysis [16]; and iv) data augmentation in human skeleton-based deep neural network model training [40]. In the design and development of these different applications, multiple experiments, or extensive data collection activities, are usually required for either the validation of prototypes or the training of deep learning-based systems.

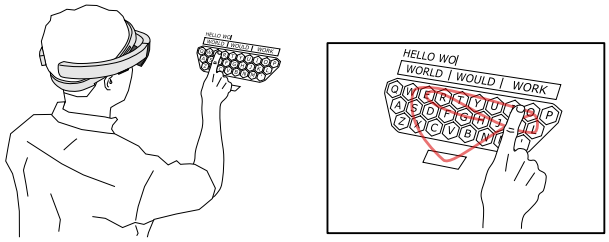


Figure 2: Illustration of a user gesturing on a mid-air gesture keyboard in AR, reproduced from Dudley et al. [12] (left), and illustration of a user's gesture trajectory (right).

We start the investigation of human motion simulation by examining low dimensional data. In this paper, we specifically explore the trajectory of an index fingertip articulating word patterns on a mid-air gesture keyboard [23, 25, 44]. It is typically referred to as "gesture typing" on Android and "QuickPath" on iOS. A user entering text on a mid-air gesture keyboard is illustrated in Figure 2. The gesture keyboard was also recently introduced in the development release for the Microsoft HoloLens 2 [34].

Two important scenarios for gesture keyboard design invoke a need for synthetic data. First, mid-air keyboard layout design optimization faces a challenging optimization problem in terms of determining different keyboard parameters, such as keyboard size, key size, the distance between keys, etc. These parameters can vary greatly in mid-air. Despite the complicated optimization task, the design process can lead to an optimized design balancing accuracy, speed, and comfort. Mainstream parameter-optimization approaches include random search, grid search, and Bayesian optimization. The first two approaches require a large amount of data and Bayesian optimization requires a tightly coupled responsive evaluation mechanism [11, 19, 20]. Considering the potentially enormous parameter space to search for an optimized design strategy, extensive user experiments are required. Computational simulation of human motion data is one potential way to efficiently reduce the number of prototypes required and thereby speeding up the design process while reducing costs. It also has the potential to fully automate this design process and revolutionize experimental-based decision-making [37].

Second, developing a neural network-based gesture keyboard recognition model for a mid-air optical see-through AR gesture keyboard is a significant undertaking. Recognition of two-dimensional

*e-mail: js2283@cam.ac.uk
†e-mail: jjd50@cam.ac.uk
‡e-mail: pok21@cam.ac.uk

Generation Model	Accuracy	Latent Space Distance	Condition on Arbitrary Keyboard Layout	Condition on Arbitrary Phrases	Optimization Time
Jerk-Minimization	Y	N	Y	Y	NA*
RNN	N	N	Y	Y	6 hours
GAN-Transfer	Y	Y	Y	Y	20 minutes
GAN-Imitation	Y	Y	N	N	20 minutes

Table 1: The characteristics of the four proposed trajectory generative models assessed in terms of different aspects and suitability for different tasks. For example, being able to adapt to a new keyboard layout and new phrases is desirable for a synthesizer when automating keyboard design. Showing a close distance in latent space suggests the potential to help data augmentation to train a machine learning based keyboard recognition model. NA* indicates the requirement for human intervention to optimize the hyperparameters. Y indicates *Yes* and N indicates *No*; for the Latent Space Distance, Y indicates *Close* and N indicates *Far*.

trajectories articulated on touchscreens (e.g. [1, 23, 25]) is less challenging compared to trajectories articulated in mid-air due to the lack of haptic feedback and inaccuracy of tracking sensors. Deep learning-based mid-air gesture keyboard recognition models are a potential approach to overcome this challenge [1]. However, a key requirement for training a deep neural network gesture keyboard decoder model is access to very large volumes of data based on representative user traces. Deep learning-based models require a large amount of training data to generalize—otherwise there is a high risk of over-fitting reducing performance. Unfortunately, an abundance of such data is unavailable for such an AR headset setting due to the limited adoption of such technology to date and the lack of concerted public data collection activities. Collecting and annotating such data requires extensive user data collection activities, which are time-consuming, expensive and challenging. Moreover, any data collected usually correspond to a single operating point (set of keyboard parameters) and thus may not generalize well beyond this setting. This significant bottleneck introduces an obstacle to the long-term goal of building a robust gesture keyboard decoder model and automating the design process for the envisaged mid-air gesture keyboard. However, if the probability distribution of human motion for this task can be accurately approximated, data can be efficiently sampled from this approximated distribution and be used as a surrogate for real samples. The data sampled from the approximated probability distribution can then be used to augment limited training data. Shen et al. [40] used a GAN-based synthesis model to sample synthetic skeleton hand gesture data from an approximated distribution of real hand gestures, and demonstrated the advantage and improvement brought by the synthetic data in data augmentation in training a deep neural network recognition model to classify the hand gestures.

However, generating realistic finger trajectories is challenging and should satisfy: i) the nonlinear system dynamics; ii) the physical limits of human hands, such as their maximum speed; and iii) the position constraints that impose that the trajectory should pass near the waypoints of the ideal gesture on the keyboard. To tackle the above challenges, we propose and explore four data synthesis approaches that can generate four types of synthetic trajectories with different properties:

1. **Jerk-Minimization Model**: This is a deterministic model and is optimized by minimizing the high order derivatives along the trajectories. It requires human intervention to fine-tune the parameters of the model in order to generate realistic trajectories [33].

2. **Recurrent Neural Network (RNN)-based Model**: This is a generative model based on a Recurrent Neural Network. We used a mixed dataset of real data and the data generated from the Jerk-Minimization model to stabilize the convergence of the model, considering the model itself is hard to train [14]. We apply an important technical innovation (detailed in Section 3.2) on the method used by Graves [14] to facilitate training despite using only one tenth of the amount of data.

3. **Generative Adversarial Network (GAN)-based Model**: We used a GAN structure similar to CycleGAN for an unpaired trajectory-to-trajectory style transfer [45]. The GAN-based model has two settings: GAN-Imitation and GAN-Transfer. GAN-Transfer transfers the styles of the real trajectories to each other, producing trajectories with more variation. GAN-Transfer transfers the style of real trajectories to synthetic trajectories that are produced by simply connecting the key centers together. The GAN-Imitation model is adapted from Shen et al. [40] and modified to synthesize fingertip gesture trajectories as opposed to hand gestures. The GAN-Transfer model is a further novel modification which enables the synthesis of new gesture trajectories for which there is no raw user data.

The gesture traces generated by these methods are validated using five micro metrics that characterize the finger motion data, a dimensionality reduction algorithm for visualization of the latent space, and finally, a simulated gesture keyboard recognizer for assessing accuracy and noise levels. We summarize the properties of the four different synthesizers in Table 1.

1. **Micro Metrics**: We use geometric micro metrics, including curvature, aspect, vicinity curliness, linearity, and slope, to characterize the low-level features of the trajectories [17]. In addition, we calculate Kullback–Leibler (KL) divergence between the micro metrics per trajectory.

2. **Accuracy**: The trajectories should be recognized accurately while maintaining a certain degree of error to mimic human behavior.

3. **Latent Space Distance**: The distance of the data distributions in latent space after dimensionality reduction. The distance reveals the similarity between latent attributes of both the temporal and spatial information, with a smaller distance indicating a better data augmentation strategy for deep learning.

4. **Condition on Keyboard Layout**: The ability to generate new trajectories conditioning on arbitrary keyboard layouts and designs.

5. **Condition on New Phrases**: The ability to generate new trajectories from any arbitrary phrase.

6. **Optimization Time**: The time to find an optimized strategy to generate realistic data.

To inform and guide the development of a generative model, we built a simulated mid-air gesture keyboard deployed on a Microsoft

HoloLens 1 optical see-through AR headset to collect real user traces. This collected dataset serves two purposes. First, it helps characterize user behavior and typical variation exhibited in mid-air gesture typing in this setting. Second, it establishes a baseline for validating the synthesized traces.

The potential of the four approaches to generate synthetic trajectories is thoroughly illustrated by the evaluation of the micro metrics, the latent space visualization, and accuracy. We note that each approach also has its strengths and weaknesses. Combining both the consideration of the characteristics of the generated trajectories and the properties of the generative model, we can find a model that suits its purpose. For example, if one needs a data augmentation model, a smaller distance in latent space is preferred, while consistency, in terms of micro metrics and accuracy, is more important for keyboard design.

In summary, our main contribution is the demonstration of four data efficient models for synthesizing trajectories with different properties suited for different design purposes. We also provide the code and dataset to support the replication and extension of this research.[1]

2 Related Work

Various strategies have been used to simulate human motion including kinematic and biological models [29], control models based on energy minimization [39, 43], and data-driven models leveraging neural networks [1, 4, 13]. Data-driven models are attractive in instances where the underlying motor control or biological behaviors are difficult to articulate. The trajectories produced while gesture typing fall into this category, although Quinn and Zhai [39] have demonstrated the potential of Jerk-Minimization-based approximations. In this paper, we focus primarily on using neural networks to synthesize motion trajectories and devote the majority of this literature review to that topic. Alternative approaches are briefly reviewed at the end of this section.

Deep neural network models trained on real human motion trajectories can approximate the probability distribution of the training data and sample new data from the approximated distribution. Alsharif et al. [1] proposed a hybrid approach combining an RNN and a conventional Finite State Transducer to address some of the challenges in 2D gesture typing on a touchscreen, such as elision, co-articulation, and high variability. One of their trajectory datasets used to train the network is generated by connecting the characters using a minimum jerk model, which is similar to the Jerk-Minimization model we propose here. Akash et al. [32] proposed a Generative Adversarial Network (GAN)-based model to generate continuous 2D trajectories that conform to user idiosyncrasies. However, the number of real trajectories required to train the proposed GAN-based model remains high (2.2 million) [32]. The key difference between this GAN-based model and our proposed novel GAN-Transfer setting is that: i) we do not require a large amount of data to train the GAN compared to the work from Akash et al. [32]; and ii) Akash et al. generate 2D trajectories for a word, whereas we generate 2D trajectories for phrases which are considerably longer than the word-based trajectories. The longer the sequence to be generated, the more challenging the task is since the neural network should process much longer temporal information. This necessitates the use of teacher forcing in the Imaginative GAN [40] to handle long temporal relationships. Handwriting synthesis, which is the generation of handwriting for a given text, is another popular research area for human motion simulation. This is similar to our research problem in that the challenge is conditioning the predictions on the text while the two sequences have very different lengths. Further, another similar challenge is that the alignment between the two sequences is unknown. One of the models which are able to tackle this difference in sequence length and the unknown alignment issue is the RNN transducer [13]. Graves et al. [14] proposed an alternative model which contains a 'soft window' that is convolved with the text string and used as an extra input to the generative network.

The other direction of human motion simulation is a biological model-based approach which incorporates muscle models and other biological considerations. The diverse behaviors of human locomotion can be generated with a complex neuromuscular gait model [41]. Ignor et al. [35] developed a lower-limb musculoskeletal model for animating human activities driven by the lower body. Reinforcement learning, which uses an agent to optimize a reward function, has had success in learning muscle activation and joint torque patterns [18, 27]. Anand et al. [2] used a deep reinforcement learning-based approach to learn the individual specific walking behavior at various walking speeds by directly learning the muscle activation pattern.

Kinematic theory related to rapid human movements [38] was leveraged by Leiva et al. [28] to produce synthesized stroke gestures that hold the same statistical characteristics as human-generated gestures. This was achieved by introducing local and global perturbations to the model parameters. Perlin noise has also been used as a perturbation model to produce realistic synthetic data for augmenting the training of a gesture recognizer [42]. Classification accuracy and statistical measures representing geometric, kinematic and articulation aspects of traces have been used to measure the similarity between the synthesized data and human-generated data [29]. Leiva et al. [30] identified a set of representative and useful features describing stroke gestures which can be used in different scenarios, such as when evaluating the quality of synthesized stroke gestures.

3 Synthesizing Gesture Traces

Two types of synthetic data can be generated: 1) synthetic data conditioned on other inputs; 2) synthetic data with its style altered. Conditional synthetic data is a more desirable form of synthetic data since it is generated conditioned on inputs, such as audio, sentences, etc. The second type of synthetic data (style alteration) results in variants of the trajectories. The trajectories will be of different styles due to hand size, posture, movement speed, etc. but still possess the original general shape.

In this paper, we will generate trajectories from two conditions. First, the trajectories can be synthesized from arbitrary phrases. Second, the trajectory generation model can be synthesized from arbitrary new keyboard layouts. For this challenging task, the approach entails transferring the styles of the real trajectories to trajectories that are produced by a deterministic generation algorithm.

3.1 Jerk-Minimization Trajectory Generation

Originating from trajectory planning applications involving Unmanned Aerial Vehicles (UAVs), and in particular quadrotors, we used a Jerk-Minimization model to generate realistic index tip trajectories [7]. This model is similar to the data augmentation approach employed in training a gesture typing recognition model used by Alsharif et al. [1]. Alsharif et al. [1] successfully demonstrated an improvement in deep learning-based recognition model performance using this approach. The trajectory is generated where high order derivatives along the trajectory are minimized while satisfying waypoints (equality) and axis-parallel box constraints (inequality). The high order derivatives refer to acceleration and jerk which is the time rate of change of acceleration itself. By minimizing high-order derivatives, we can effectively generate human-like trajectories provided by different constraints. We have taken several constraints and costs into consideration including: i) the time to complete a phrase on the gesture keyboard; ii) the distance between the waypoints (key centers) and the trajectory; and iii) the energy consumption, which is the length of the trajectory. The objective and constraints are

[1] www.github.com/CambridgeIIS/Gesture-Keyboard-Traj-Gen

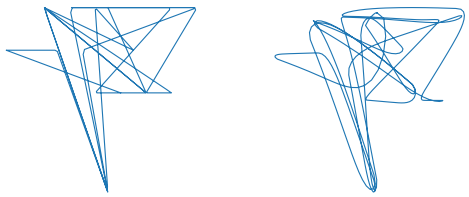
formulated in quadratic programming. We use piecewise polynomials to parameterize the trajectory. It defines the primitive of the curve as a polynomial spline, and the optimization target is polynomial coefficients. We implemented the model from Spedicato et al. [33] and included our aforementioned specific constraints and costs. There are several hyperparameters, including the polynomial order, the number of segments of polynomial, and the maximum degree of continuity when joining two segments, that need to be tuned for the purpose of making the generated trajectory as similar to the real trajectory as possible. The similarity measure is calculated from the Kullback–Leibler (KL) divergence of the micro metrics describing the trajectories. We used Bayesian optimization to facilitate the tuning of the hyperparameters [5]. A Gaussian Process is used as the surrogate model and a multi-objective acquisition ensemble algorithm (MACE) as the acquisition function. The surrogate model approximates the function of the jerk-minimization model from historical interactions whilst providing uncertainty estimates needed to guide exploration. The acquisition function acts as a proxy to the true sequential risk, measuring the utility of gathering new input points by trading off exploration and exploitation [6]. After the optimization, we manually check the samples to fine-tune the hyperparameters.

3.2 RNN-Based Trajectory Generation

Handwriting synthesis is closely related to our research topic since it is also based on generating trajectories conditioned on text. Graves et al. [14] proposed a model consisting of three LSTM layers and a predictive network for handwriting synthesis. LSTM units overcome the inability of traditional RNNs to handle long-term dependencies by including read and write gates that learn how to use and store information in an artificial memory cell [15, 22]. The LSTM unit is composed of a cell, an input gate, an output gate, and a forget gate and we use a standard structure [15]. The input to the model, $\vec{x}_t$ where t is the time step, is the offset between each trajectory point. The output describes a Mixture Gaussian model, which is the predictive network that is used to predict the next input $\vec{x}_t$, which is also offset to the next trajectory point. For the model to generate trajectories conditioned on text, it has a 'soft window' which is convolved with a set of character vectors and fed in as an extra input to the prediction network. Therefore, the model can generate trajectories based on the text string convolved with the 'soft window'.

Synthesizing fingertip trajectories of gestured words and phrases is more challenging than synthesizing handwriting. For example, gesture trajectories do not have a clear boundary between consecutive characters unlike in handwriting where each character has a distinctive trajectory and is spatially separated. As a consequence, only a gesture trajectory for a complete word is meaningful. Complicating matters further is the fact that distinct words can have very spatially similar gesture trajectories, such as 'vampire' and 'value'.

These various ambiguities and lack of common patterns in the trajectory data frustrate efforts to train a deep neural network without an enormous amount of data. We therefore introduce a novel modification to the typical training scheme. We encode the phrases by one-hot encoding on a two-character horizon (bigram) to capture key-to-key transitions. Each input vector $\vec{c}_u$ at index u has a length of $n \cdot 729$ (27×27 exhaustive bigram combinations of 26 letters and one space key), where n is the number of characters in the corresponding phrase. We use an attention mechanism that solves the alignment problem by allowing the network to leverage all information held by the original sequence. It then generates the proper sequence according to the currently processed character given the previous character and recent trajectory. To further mitigate the paucity of data, we combine the synthetic trajectories from the Jerk-Minimization model with the user trajectories to form mixed training data.



(a) Before *Transfer*. (b) After *Transfer*.

Figure 3: Example traces in the *Transfer* setting for the phrase: 'bad for the environment'. a) Simple trajectory connecting the key centers. b) Trajectory after style transfer from real trajectories.

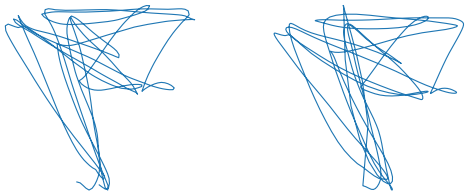
3.3 GAN-Based Trajectory Generation

A Generative Adversarial Network (GAN) trains a pair of networks competing against each other: a *generator* attempts to generate as real samples as possible while a *discriminator* attempts to distinguish between the generated and real samples. We adopted the model from Shen et al. [40], which proposed the 'Imaginative GAN' which is a data-efficient model that can transfer styles between two datasets in two domains, such as CycleGAN but in trajectory domains instead of image domains [45]. Shen et al. used the Imaginative GAN to augment the training data for hand gesture classification model, which achieves great performance in the classification accuracy and the time to find an optimized augmentation strategy is greatly reduced. The Imaginative GAN introduced teacher forcing in the generator, such that ground truth data is used instead of noise to be the input to the generator. This efficiently reduces the amount of data required to train a stable GAN and handles the long temporal information of the data.

The goal of the Imaginative GAN is to transfer the latent attributes between two domains. The latent attributes include behavioral attributes such as posture, movement speed, and physical attributes, such as hand size. While the Imaginative GAN generates synthetic skeleton-based hand gesture data, the GAN-based synthesizer we propose can generate synthetic trajectories. Based on the Imaginative GAN, we developed two modes of operation for GAN-based synthesizers: GAN-Imitation (GAN-I) and GAN-Transfer (GAN-T).

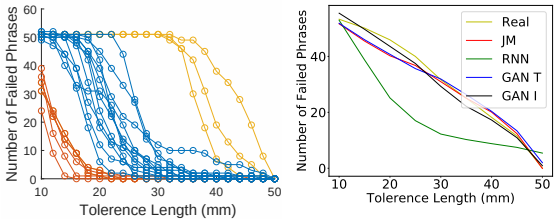
1. **GAN-Imitation**: The two inputs X and Y are the complete set of real trajectories. Therefore, while performing stochastic gradient descent on the model in mini-batches, styles can be transferred within the dataset, resulting in more trajectories with diverse attributes. The GAN-Imitation model is adapted from the method introduced in Shen et al. [40] for synthesizing hand gestures.

2. **GAN-Transfer**: The two inputs X and Y to the model are the straight line segments connecting the key centers together and the original trajectories, respectively. Therefore, the styles of the original trajectories, such as the curliness, linearity, slope, and curvature are transferred to the line segments, producing realistic trajectories that adapt to new keyboard layouts and new phrases. This approach is similar to the work from Akash et al. [32] but requires substantially fewer real traces. Our GAN is easy to train and does not require large amounts of data: a few hundred traces is sufficient. This GAN-Transfer setting is a novel extension of the Imaginative GAN [40], enabling the transfer of learned human behavioral features onto the purely synthetic raw trajectories.



(a) Before *Imitation*. (b) After *Imitation*.

Figure 4: Example traces in the *Imitation* setting for the phrase: 'bad for the environment'. a) Real trajectory. b) Trajectory after being altered by GAN-Imitation.



(a) (b)

Figure 5: a) Number of failed phrases as a function of tolerance length for each participant. b) The trade-off between accuracy and robustness for the four synthesization approaches compared to the real trajectories, computed as the average of each type of trajectory.

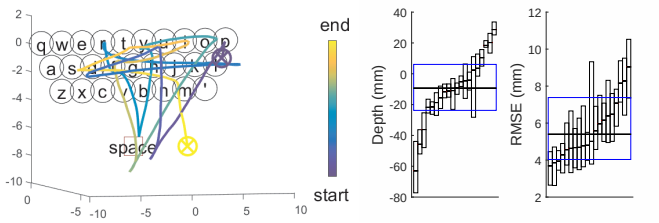
4 Experiments

4.1 User Trace Data Collection

To ensure we collected representative traces, we used a Wizard of Oz approach [8]. We built a simulated mid-air AR gesture keyboard model that simulates a 'real' recognition model and enables the collection of representative user traces. The simulated function evaluates whether the user's fingertip passes within the tolerance region of the intended character keys in sequence for a given stimulus phrase. Such an approach has been used before to study the feasibility of novel text entry methods that are difficult to implement (e.g. [10, 24]). The Wizard of Oz approach requires that the system has prior knowledge of the stimulus text but this is well-suited to a typical text entry transcription task.

Motivated by the difficulty in clearly delimiting individual word gestures, we introduced the requirement that users must explicitly articulate the space between each word by tracing over the spacebar. From the perspective of gesture synthesis, this means that the spacebar just represents an additional key to be visited and influences the form of gestures in the same way as any other key. This implementation contrasts with a typical gesture keyboard implementation on a capacitive touchscreen where the 'touch down / lift off' serves as a clear indication of the start and end of a word gesture.

We used a Microsoft HoloLens 1 to display a virtual keyboard and we used an OptiTrack motion capture system to collect accurate finger traces. We collected gesture traces from 20 participants. The average age of the participants was 24 with a standard deviation of 4 years. Among the participants, there were 13 male participants and 7 female participants. 3 participants had experience with Augmented Reality Head Mounted Display (HMD) while the rest did not. Participants were given 20 minutes to familiarize themselves with the AR HMD by gesturing test phrases. The stimulus phrases were taken from the Enron [21], and MacKenzie phrase sets [31] and each participant typed 51 randomly selected phrases. A total of 1020



(a) (b)

Figure 6: a) Example of the three dimensional raw trace data for the phrase: 'I talked to Duran'. b) Median and interquartile range for depth (left) and planar error RMSE (right) for each participant.

unique phrases were entered with an average of 5.3 words per phrase. The average length of the trajectory for each phrase was 607 samples with a standard deviation of 244 samples when logged at a sample rate of 100 Hz. Figure 6a illustrates one instance of a collected trace. For data collection, the tolerance length of the simulated recognizer was set to 50 mm on a keyboard with an apparent size of 180 $\times$ 76 mm. This tolerance length was determined in pilot testing as a suitable value to encourage users to draw natural traces instead of forcing them to sequentially travel to the center of each target key.

We transformed the coordinates of the keyboard displayed on the HoloLens into a common frame with the OptiTrack's tracking data, allowing the tolerance-based simulated gesture recognizer to run 'online'. No input feedback was provided to participants until the trace of the set stimulus phrase was completed.

Figure 6b captures a view of users' perception of the virtual keyboard. It plots the root mean square error (RMSE) between the trajectory points and a plane fitted to those same points (right) and the depth of that plane to the actual keyboard (left). This plot highlights the fact that most people tend to gesture consistently within a plane, albeit not necessarily within the virtual keyboard plane. This finding is also reported by Khan et al. [9]. This suggests it is possible to reduce the dimensionality by projecting the raw data points onto a plane. This simpler representation allows a deep neural network to be more efficiently trained.

4.2 Preprocessing

This section describes the preprocessing steps applied to the raw data before training the generative models, as well as additional processing of generated data before the evaluation of the micro metrics. The main steps of the preprocessing include denoising, interpolating, resampling and normalizing.

1. **Denoising**: We use a Savitzky–Golay filter to remove noise. This is achieved by fitting successive subsets of adjacent data points with a low degree polynomial using linear least squares. The window length is set to 7 and the order of the polynomial used to fit the data is set to 3.

2. **Interpolation**: Spline interpolation of zeroth order is used to interpolate any missing points of the raw data.

3. **Resampling**: This step is only for processing the synthetic trajectories, but not for preprocessing the training data of the RNN- and GAN-based models as temporal information will be lost after resampling, which is crucial information when training the generative models. The generated data is resampled to ensure adjacent points are equidistant. Figure 7 shows the trajectory before and after equidistant resampling.

4. **Normalizing**: The data is divided by the Frobenius norm along the temporal dimension. The goal of normalization is to ensure

(a) Before resampling. (b) After resampling.

Figure 7: Trajectories before and after equidistant resampling.

the trajectories lie within a common bounding box. This promotes speed and stability in training the deep neural networks.

4.3 Evaluation of Geometric Micro Metrics

To assess the ability of the synthesized trajectories to approximate real user traces, we compare their low-level features by examining a set of geometric micro metrics. These micro metrics are adapted from Jaeger et al. [17], who originally used them to characterize handwriting trajectories. This assessment serves in part to determine whether the generative models can replicate human-like features in the trace.

The micro metrics are computed in a *vicinity*, as illustrated in Figure 9a. A vicinity $V(t)$ is a bounding box which contains the preceding and succeeding points of $(x(t), y(t))$.

1. **Curvature**: The curvature of a point $(x(t), y(t))$ is measured by β in Figure 9b. It can be calculated as follows:

$$\vec{u} = (x(t-3), y(t-3)) - (x(t), y(t))$$

$$\vec{v} = ((x(t+3), y(t+3)) - (x(t), y(t)))$$

$$\beta = \arccos \frac{\vec{u} \cdot \vec{v}}{\|\vec{u}\| \times \|\vec{v}\|}$$

2. **Aspect**: The aspect $A(t)$ of a point $(x(t), y(t))$ characterizes the height-to-width ratio of $V(t)$, as shown in Figure 9a. It is calculated as:

$$A(t) = \frac{\Delta y(t) - \Delta x(t)}{\Delta y(t) + \Delta x(t)}$$

where $\Delta y(t)$ and $\Delta x(t)$ are respectively the height and width of the vicinity.

3. **Curliness**: Curliness $C(t)$ characterizes the deviation from a straight line in the vicinity. It is calculated as:

$$C(t) = \frac{l(t)}{max(\Delta x(t), \Delta y(t))}$$

where $l(t)$ is the sum of lengths of all line segments in vicinity $V(t)$. $\Delta x(t)$ and $\Delta y(t)$ are the width and height of the vicinity $V(t)$.

4. **Linearity**: Linearity $L(t)$ is measured as the average square distance, with each distance defined to be d_i, between the N points in the vicinity $V(t)$ and the straight line joining the first and last point in this vicinity. It is defined as follows:

$$L(t) = \frac{1}{N} * \sum_i d_i^2$$

5. **Slope**: The slope describes the angle α in degrees. It measures the slope of the straight line joining the first and last point in the vicinity $V(t)$ and is calculated as:

$$\alpha = \arctan \frac{\Delta y(t)}{\Delta x(t)}$$

4.4 Implementation Details

We used a computer that has 3 $\times$ Nvidia GeForce RTX 3090 GPU, 1 $\times$ Ryzen 9 3970x CPU.

4.4.1 Jerk-Minimization Model

The polynomial order is 4. The maximally imposed continuity between segments is 1. The weights for the first, second, and third-order derivatives are 100, 4, 0, respectively. We used KL divergence for the five proposed micro metrics for the loss metric, and the number of maximum iteration is set to 50.

4.4.2 RNN-based Model

We adopted the same training scheme from Graves et al. [14]. We trained the generative model using a dataset of 40,000 traces consisting of 10% real user trajectories and 90% traces generated from the Jerk-Minimization model.

4.4.3 GAN-based Model

We use the same training scheme from Zhu et al. [45]. The training batch size is 64. λ_1 and λ_2 in the objective function for the generators are set to 10 and 5, respectively. The number of hidden units for the GRU cell in the generator is 512.

4.4.4 Procedure

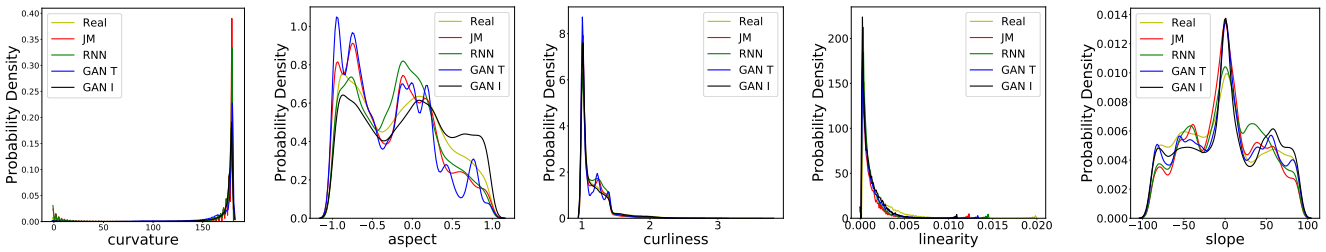
We generated synthesized traces corresponding to the 1,020 phrases used to collect the 1,020 real gesture trajectories. The vicinity size is set to 7 points. The micro metrics are computed averaged across all phrases. The KL divergence is computed for each phrase. In Section 5, we present only a subset of phrases to improve visual clarity. We used 0.99 as the number of components to keep for the PCA decomposition, perplexity equal to 25, and the maximum number of iterations for the optimization equal to 2000 for t-SNE.

5 Results

In this section, we compare the geometric micro metrics computed for each trajectory generation method against real user traces. Later in Section 5.2, we visualize the different generation methods in terms of the latent features they replicate. Finally, in Section 5.3, we assess the degree to which the different synthesized traces replicate the performance characteristics of real user traces in terms of simulated recognition accuracy.

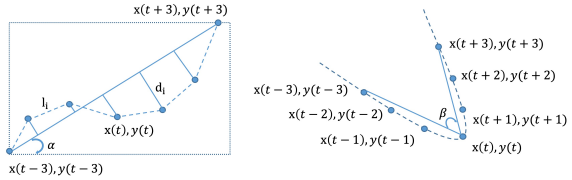
5.1 Geometric Micro Metrics

The micro metrics characterize the trajectories. The different types of trajectories all have a high probability of curvature around 180 degrees, while Jerk-Minimization trajectories exhibit an even higher probability density. Trajectories produced by RNN have a higher probability density for curvature at zero degrees. For all the micro metrics, it is the GAN-Transfer trajectories that show the closest KL distance to the real trajectories. Therefore, if design automation is the purpose, GAN-Transfer is most desired as it not only has the smallest KL divergence on those micro metrics, but it can also condition on arbitrary keyboard layouts and phrases. The Jerk-Minimization and the RNN-based models are good choices to start with as well since they can all generate trajectories according to arbitrary keyboard layout and phrases.



(a) (b) (c) (d) (e)

Figure 8: The histogram of the five micro metrics for the five modes of trajectories. i) **Real**: Real trajectories. ii) **JM**: Trajectories generated from Jerk-Minimization model. iii) **RNN**: Trajectories generated from the RNN-based generative model. iv) **GAN-T**: Trajectories generated from the GAN-based generative model in the *Transfer* mode. v) **GAN-I**: Trajectories generated from the GAN-based generative model in the *Imitation* mode.



(a) Vicinity. (b) Curvature.

Figure 9: Illustration of vicinity $V(t)$.

5.2 Latent Space Distance

Visualization through dimensionality reduction can make high-dimensional data easier to interpret. We first use a two-layer Bi-LSTM model with 400 neurons each layer to extract the latent feature vectors from the data. Then we perform dimensionality reduction using first Principal Component Analysis (PCA) for efficiency, which is then followed by t-Distributed Stochastic Neighbor Embedding (t-SNE) for accuracy. Figure 11 shows that GAN-based trajectories lie closest to the original trajectories in latent space. This suggests that this model may perform well in training deep neural networks since deep neural networks first extract latent features from the data and then perform recognition by outputting a probability distribution of outputs based on those latent features. It is noted that the similarity/distance measure from micro metrics and latent space can be different for the trajectories. This is because latent space visualization accounts for more latent attributes that cannot be characterized by micro metrics, such as temporal correlation. Temporal correlation is an important factor in training deep-learning based models as it contains temporal information. For example, temporal information can help to strengthen the confidence of the identification of turning points, gesture error and ambiguous spatial similarity.

5.3 Accuracy

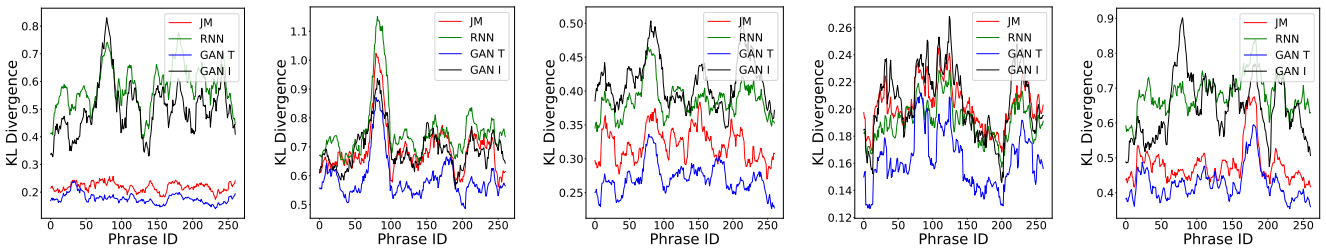
The synthesized trajectories should have some human-like errors, as illustrated in Figure 5a. Figure 5a captures the degree of variation exhibited in the traces. This accuracy metric is produced by passing the trajectories to the aforementioned simulated recognizer that uses the tolerance region as the key recognition condition. This recognizer is neutral and does not bias towards any type of data. Accuracy is measured as the smallest tolerance length that allows the trace to pass the simulated recognition model. Therefore, the lower the number of phrases to fail the simulated recognition model, the higher the accuracy. Participants shown in red in Figure 5a exhibit high accuracy since their traces mostly pass a tolerance length of 20 mm. Participants shown in yellow in Figure 5a have difficulty in gesturing accurate traces. Such participants often failed to pass the simulated recognition model and were required to repeat the trace. The participants indicated in yellow gestured a variety of inconsistent traces that lead to many of their traces failing the simulated recognition model, even at 40 mm. However, participants indicated in red demonstrate a very consistent approach to gesturing with little variation. Participants indicated in blue, which also represent the majority of the participants, demonstrate a good balance between variation and accuracy. They exhibit variability in gesturing within 30 mm and accuracy between 30 mm and 50 mm. Figure 5b shows that trajectories from the RNN-based approach only learn from the behavior of the majority while it ignores the dynamics exhibited by the minority. In contrast, the trajectories from both the Jerk-Minimization and GAN-based approaches achieve a good balance between accuracy and error.

6 Discussion

The results presented in Section 5 highlight the various capabilities and limitations of the trajectory generation methods. However, as previously stated, an important higher-level consideration is whether a generative model can produce trajectories for unobserved phrases and layouts. Table 1 summarizes the capabilities of the models in this regard. Apart from the GAN-Imitation model, it is an intrinsic ability for all the models to be able to generate synthetic data from unobserved phrases and keyboard layouts. The Jerk-Minimization model produces trajectories based on the absolute positions of the via-points, which can be generated from an arbitrary input phrase and keyboard layout. The novel training scheme for the RNN-based generation model, which incorporates synthetic traces from the Jerk-Minimization model, also allows the model to learn a mapping from text and keyboard layouts to trajectories. Finally, the GAN-Transfer generation model transfers realistic and representative styles and specific human behaviors from captured real data to synthetic trajectories produced by simply connecting key centers.

The analysis in this paper is constrained to a set of phrases of roughly the same length. For the Jerk-Minimization model, the quality of generated trajectories is independent of the number of words given that it is a deterministic movement model. Both the RNN-based and GAN-based models are composed of LSTM cells which can account for long temporal dependencies. Nevertheless, the quality of these models is dependent on the total number of trajectory points and may be negatively impacted by significant increases in phrase length. It is possible to mitigate the potential negative effects of increasing phrase length by reducing the assumed number of trajectory points associated with each character.

The micro metrics introduced in Section 4.3 are chosen to characterize how closely the synthesized trajectories reflect human-like behaviors. There are several qualities observable in the synthetic



(a) Curvature. (b) Aspect. (c) Curliness. (d) Linearity. (e) Slope.

Figure 10: Kullback-Leiblear (KL) divergence of different micro metrics per phrase. The Jerk-Minimization model generates trajectories that have similar curvature properties to the original trajectories.

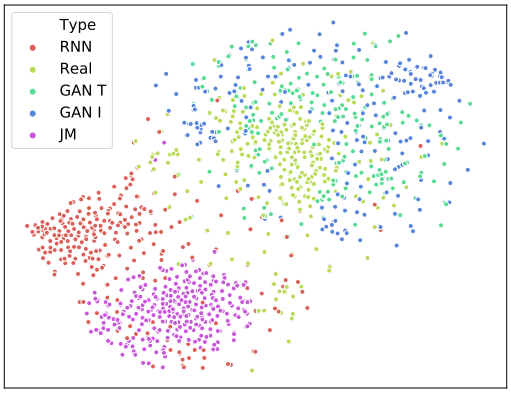


Figure 11: The latent space visualization of the four types of synthetic trajectories compared to the real trajectory through dimensionality reduction. Each colored cluster represents a group of featured trajectories. We note that GAN-Transfer and Imitation trajectories lie closest to the original trajectories. This suggests that the latent attributes of the two types of trajectories exhibit a closer distance to the latent attributes of real trajectories.

data indicating good alignment with the behaviors and features present in real human trajectories. First, the models tend to follow a straight line between via-points (as highlighted by the linearity metric), which is consistent with observed patterns of motor control in terms of energy minimization. Second, the models produce qualities intuitively expected given the keyboard layout: the aspect metric, where -1 indicates width, is much greater than height, which correctly suggests that the keyboard has a larger width than height while the slope metric correctly suggests that the keyboard is symmetric. Third, the synthetic traces contain sharp turns at some of the via-points but otherwise remain smooth over their trajectories (based on the curvature and curliness metrics), which is in alignment with what is typically observed in user-generated traces. Finally, Figure 9 shows that all models exhibit noise levels close to those observed in human-generated traces. These broad consistencies all suggest that our models exhibit human-like behaviors.

7 Limitations and Future Work

We have demonstrated the potential of using state-of-the-art generative models to synthesize realistic lower dimensional human motion trajectories. There are many higher dimensional human motion trajectories involving spatial information that are also suitable for this treatment. Variants of high dimensional trajectories can be produced from the Imaginative GAN since it enables the data to transfer styles internally. However, generating conditional trajectories conditioned on other inputs, such as performing continuous hand gestures or human motion based on a continuous input, is challenging. A promising avenue of future work is to simulate high-dimensional human motion data and generate data that is conditioned on other inputs. We are interested in exploring this challenge using reinforcement learning. Another avenue of future work is to use the Jerk-Minimization model, RNN-based model and the GAN-Transfer synthesizer to automate aspects of mid-air gesture keyboard design by exploiting Bayesian optimization and comparing the performance of the three approaches. A deep-learning based recognition model can also be trained with synthetic data, preferably from the GAN-based synthesizers, to prevent over-fitting. We hypothesize this will achieve better performance than using the limited real dataset.

8 Conclusions

In this paper we have proposed and explored four approaches for generating synthetic trajectories that are suitable for different scenarios. The different approaches produce trajectories that exhibit distinct characteristics and unique properties, which can be exploited depending on the specific design problems at hand. For example, for design optimization, Jerk-Minimization may be the preferred option since the model easily adapts to new layouts and can generate traces for unseen phrases. For training a deep learning-based recognition model the GAN-Imitation and GAN-Transfer models are preferable since they exhibit a closer latent space distribution to real data. Overall, GAN-Transfer demonstrates well-rounded properties across all evaluation metrics, including accuracy, latent space distance, conditioning on arbitrary keyboard layout and phrases, and optimization time. We anticipate that these different approaches to simulating realistic human motion trajectories will provide a useful tool in advancing the state-of-the-art of mid-air gesture typing in augmented and virtual reality.

Acknowledgments

John Dudley and Per Ola Kristensson were supported by EPSRC (grant EP/S027432/1).

References

[1] O. Alsharif, T. Ouyang, F. Beaufays, S. Zhai, T. Breuel, and J. Schalkwyk. Long short term memory neural network for keyboard gesture decoding. In *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 2076–2080, 2015.

[2] A. S. Anand, G. Zhao, H. Roth, and A. Seyfarth. A deep reinforcement learning based approach towards generating human walking behavior with a neuromuscular model. In *2019 IEEE-RAS 19th International Conference on Humanoid Robots (Humanoids)*, pp. 537–543. IEEE, 2019.

[3] A. Antakli, E. Hermann, I. Zinnikus, H. Du, and K. Fischer. Intelligent distributed human motion simulation in human-robot collaboration

environments. In *Proceedings of the 18th International Conference on Intelligent Virtual Agents*, IVA '18, p. 319–324. Association for Computing Machinery, New York, NY, USA, 2018. doi: 10.1145/3267851.3267867

[4] D. Bullock and S. Grossberg. Neural dynamics of planned arm movements: emergent invariants and speed-accuracy properties during trajectory formation. *Psychological review*, 95(1):49, 1988.

[5] A. I. Cowen-Rivers, W. Lyu, R. Tutunov, Z. Wang, A. Grosnit, R. R. Griffiths, H. Jianye, J. Wang, and H. B. Ammar. An empirical study of assumptions in bayesian optimisation. *arXiv preprint arXiv:2012.03826*, 2020.

[6] A. I. Cowen-Rivers, W. Lyu, Z. Wang, R. Tutunov, H. Jianye, J. Wang, and H. B. Ammar. Hebo: Heteroscedastic evolutionary bayesian optimisation. *arXiv preprint arXiv:2012.03826*, 2020. winning submission to the NeurIPS 2020 Black Box Optimisation Challenge.

[7] N. Dadkhah and B. Mettler. Survey of motion planning literature in the presence of uncertainty: Considerations for uav guidance. *Journal of Intelligent & Robotic Systems*, 65(1):233–246, 2012.

[8] N. Dahlbäck, A. Jönsson, and L. Ahrenberg. Wizard of oz studies: why and how. In *Proceedings of the 1st international conference on Intelligent user interfaces*, pp. 193–200, 1993.

[9] N. K. Dim, C. Silpasuwanchai, S. Sarcar, and X. Ren. Designing mid-air tv gestures for blind people using user- and choice-based elicitation approaches. In *Proceedings of the 2016 ACM Conference on Designing Interactive Systems*, DIS '16, p. 204–214. Association for Computing Machinery, New York, NY, USA, 2016. doi: 10.1145/2901790.2901834

[10] J. Dudley, H. Benko, D. Wigdor, and P. O. Kristensson. Performance envelopes of virtual keyboard text input strategies in virtual reality. In *2019 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*, pp. 289–300. IEEE, 2019.

[11] J. J. Dudley, J. T. Jacques, and P. O. Kristensson. *Crowdsourcing Interface Feature Design with Bayesian Optimization*, p. 1–12. Association for Computing Machinery, New York, NY, USA, 2019.

[12] J. J. Dudley, K. Vertanen, and P. O. Kristensson. Fast and precise touch-based text entry for head-mounted augmented reality with variable occlusion. *ACM Trans. Comput.-Hum. Interact.*, 25(6), Dec. 2018.

[13] A. Graves. Sequence transduction with recurrent neural networks. *arXiv preprint arXiv:1211.3711*, 2012.

[14] A. Graves. Generating sequences with recurrent neural networks, 2013.

[15] S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[16] R. M. Hu, Z. D. He, and F. Bai. The research of 3d human motion simulation and video analysis system implemented in sports training. In *Progress in Applied Sciences, Engineering and Technology*, vol. 926 of *Advanced Materials Research*, pp. 2743–2746. Trans Tech Publications Ltd, 7 2014. doi: 10.4028/www.scientific.net/AMR.926-930.2743

[17] S. Jaeger, S. Manke, J. Reichert, and A. Waibel. Online handwriting recognition: the npen++ recognizer. *International Journal on Document Analysis and Recognition*, 3(3):169–180, 2001. doi: 10.1007/PL00013559

[18] Y. Jiang, T. Van Wouwe, F. De Groote, and C. K. Liu. Synthesis of biologically realistic human motion using joint torque actuation. *ACM Transactions On Graphics (TOG)*, 38(4):1–12, 2019.

[19] F. Kadner, Y. Keller, and C. Rothkopf. Adaptifont: Increasing individuals' reading speed with a generative font model and bayesian optimization. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*, CHI '21. Association for Computing Machinery, New York, NY, USA, 2021. doi: 10.1145/3411764.3445140

[20] M. M. Khajah, B. D. Roads, R. V. Lindsey, Y.-E. Liu, and M. C. Mozer. Designing engaging games using bayesian optimization. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*, CHI '16, p. 5571–5582. Association for Computing Machinery, New York, NY, USA, 2016. doi: 10.1145/2858036.2858253

[21] B. Klimt and Y. Yang. The enron corpus: A new dataset for email classification research. In J.-F. Boulicaut, F. Esposito, F. Giannotti, and D. Pedreschi, eds., *Machine Learning: ECML 2004*, pp. 217–226. Springer Berlin Heidelberg, Berlin, Heidelberg, 2004.

[22] J. F. Kolen and S. C. Kremer. *A field guide to dynamical recurrent networks*. John Wiley & Sons, 2001.

[23] P. O. Kristensson. *Discrete and Continuous Shape Writing for Text Entry and Control*. PhD thesis, Linköping University, MDA - Human Computer Interfaces, 2007.

[24] P. O. Kristensson and K. Vertanen. The potential of dwell-free eye-typing for fast assistive gaze communication. In *Proceedings of the symposium on eye tracking research and applications*, pp. 241–244, 2012.

[25] P. O. Kristensson and S. Zhai. Shark^2:a large vocabulary shorthand writing system for pen-based computers. *Proceedings of the 17th Annual ACM Symposium on User Interface Software and Technology*, 01 2004.

[26] J. Lee, J. Chai, P. S. A. Reitsma, J. K. Hodgins, and N. S. Pollard. Interactive control of avatars animated with human motion data. *ACM Trans. Graph.*, 21(3):491–500, July 2002. doi: 10.1145/566654.566607

[27] Y. Lee, M. S. Park, T. Kwon, and J. Lee. Locomotion control for many-muscle humanoids. *ACM Trans. Graph.*, 33(6), Nov. 2014. doi: 10.1145/2661229.2661233

[28] L. A. Leiva, D. Martín-Albo, and R. Plamondon. Gestures à go go: Authoring synthetic human-like stroke gestures using the kinematic theory of rapid movements. *ACM Trans. Intell. Syst. Technol.*, 7(2), Nov. 2015. doi: 10.1145/2799648

[29] L. A. Leiva, D. Martín-Albo, and R. Plamondon. The kinematic theory produces human-like stroke gestures. *Interacting with Computers*, 29(4):552–565, 2017. doi: 10.1093/iwc/iww039

[30] L. A. Leiva, R.-D. Vatavu, D. Martín-Albo, and R. Plamondon. Omnis prædictio: Estimating the full spectrum of human performance with stroke gestures. *International Journal of Human-Computer Studies*, 142:102466, 2020. doi: 10.1016/j.ijhcs.2020.102466

[31] I. S. MacKenzie and R. W. Soukoreff. Phrase sets for evaluating text entry techniques. In *CHI '03 Extended Abstracts on Human Factors in Computing Systems*, CHI EA '03, p. 754–755. Association for Computing Machinery, New York, NY, USA, 2003.

[32] A. Mehra, J. R. Bellegarda, O. Bapat, P. Lal, and X. Wang. Leveraging gans to improve continuous path keyboard input models. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 8174–8178. IEEE, 2020.

[33] D. Mellinger and V. Kumar. Minimum snap trajectory generation and control for quadrotors. In *2011 IEEE international conference on robotics and automation*, pp. 2520–2525. IEEE, 2011.

[34] Microsoft. Hololens 2 release notes. `https://docs.microsoft.com/en-us/hololens/hololens-release-notes#swipe-to-type`.

[35] I. Mordatch, J. M. Wang, E. Todorov, and V. Koltun. Animating human lower limbs using contact-invariant optimization. *ACM Transactions on Graphics (TOG)*, 32(6):1–8, 2013.

[36] S. Narang, A. Best, and D. Manocha. Simulating movement interactions between avatars agents in virtual worlds using human motion constraints. In *2018 IEEE Conference on Virtual Reality and 3D User Interfaces (VR)*, pp. 9–16, 2018. doi: 10.1109/VR.2018.8446152

[37] J. F. Nunes, P. M. Moreira, and J. M. R. Tavares. Human motion analysis and simulation tools: a survey. In *Handbook of Research on Computational Simulation and Modeling in Engineering*, pp. 359–388. IGI Global, 2016.

[38] R. Plamondon. A kinematic theory of rapid human movements. *Biological Cybernetics*, 73:95–95, 06 1995. doi: 10.1007/BF00199060

[39] P. Quinn and S. Zhai. Modeling gesture-typing movements. *Human–Computer Interaction*, 33(3):234–280, 2018. doi: 10.1080/07370024.2016.1215922

[40] J. Shen, J. Dudley, and P. O. Kristensson. The imaginative generative adversarial network: Automatic data augmentation for dynamic skeleton-based hand gesture and human action recognition. *arXiv*, 2021.

[41] S. Song and H. Geyer. A neural circuitry that emphasizes spinal feedback generates diverse behaviours of human locomotion. *The Journal of Physiology*, 593(16):3493–3511, 2015. doi: 10.1113/JP270228

[42] E. M. Taranta, M. Maghoumi, C. R. Pittman, and J. J. LaViola Jr. A rapid prototyping approach to synthetic data generation for improved 2d gesture recognition. In *Proceedings of the 29th Annual Symposium on User Interface Software and Technology*, pp. 873–885, 2016.

[43] E. Todorov and M. I. Jordan. Smoothness maximization along a pre-

defined path accurately predicts the speed profiles of complex arm movements. *Journal of Neurophysiology*, 80(2):696–714, 1998. PMID: 9705462. doi: 10.1152/jn.1998.80.2.696

[44] S. Zhai and P.-O. Kristensson. Shorthand writing on stylus keyboard. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '03, p. 97–104. Association for Computing Machinery, New York, NY, USA, 2003.

[45] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Proceedings of the IEEE international conference on computer vision*, pp. 2223–2232, 2017.