Stroke-Hover Intent Recognition for Mid-air Curve Drawing Using Multi-Point Skeletal Trajectories

Umema H. Bohari
Schlumberger Corporation
14910 Airline Road
Rosharon, TX 77583, USA
Email: umemabohari@gmail.com

Ryan Alli
J. Mike Walker '66 Department of Mechanical Engineering
Texas A&M University
College Station, Texas 77843, USA
Email: realli15@yahoo.com

Alejandra Garcia
J. Mike Walker '66 Department of Mechanical Engineering
Texas A&M University
College Station, Texas 77843, USA
Email: aleja.r.garcia@gmail.com

Vinayak R. Krishnamurthy*
J. Mike Walker '66 Department of Mechanical Engineering
Texas A&M University
College Station, Texas 77843, USA
Email: vinayak@tamu.edu

Drawing curves is a fundamental task in mid-air interactive applications such as 3D sketching, geometric modeling, handwriting recognition, and authentication. Existing research in mid-air drawing is solely focused on determining what the user drew assuming that the intended curve is segmented from the continuous user-generated trajectory. In this work, our aim is to address the complementary problem: to determine when the user actually intended to draw without the use of any prescribed gestures or hand-held controllers (e.g. Wii remote, HTC Vive). In our previously published work, we demonstrated that in mid-air drawing tasks, not only is it possible to statistically learn drawing intent from hand motion, but it is also perceived to be more natural by users. Our idea was to simply classify each instance of hand trajectories as either a stroke or a hover. Our current work investigates new representations of the users' motion beyond a single point (such as a tracked palm) to richer multi-point trajectories obtained with other skeletal joints such as wrist and elbow. We trained several binary classifiers on 5 *such trajectory representations obtained from 3D drawing data from* 25 *users using a hand tracking device. We compare these representations and the corresponding classifiers for predicting user intent for mid-air drawing. Our extended approach resulted in improved prediction accuracy (mean:* 80.17%*, min:* 79.92%*, max:* 91.30%*) with respect to our earlier work (mean:* 76.75%*, min:* 74.23%*, max:* 84.01%*).*

1 Introduction

Spatial (mid-air) interactions are now commonly investigated in applications such as geometric modeling [1], 3D sketching [2, 3], mid-air authentication [4], and text input [5–7]. To this end, many techniques have been developed for mid-air symbol recognition [6, 8]. However, most existing research in mid-air curve input is focused on determining *what the user intends to express* (i.e. recognizing a symbol within

*Address all correspondence to this author.

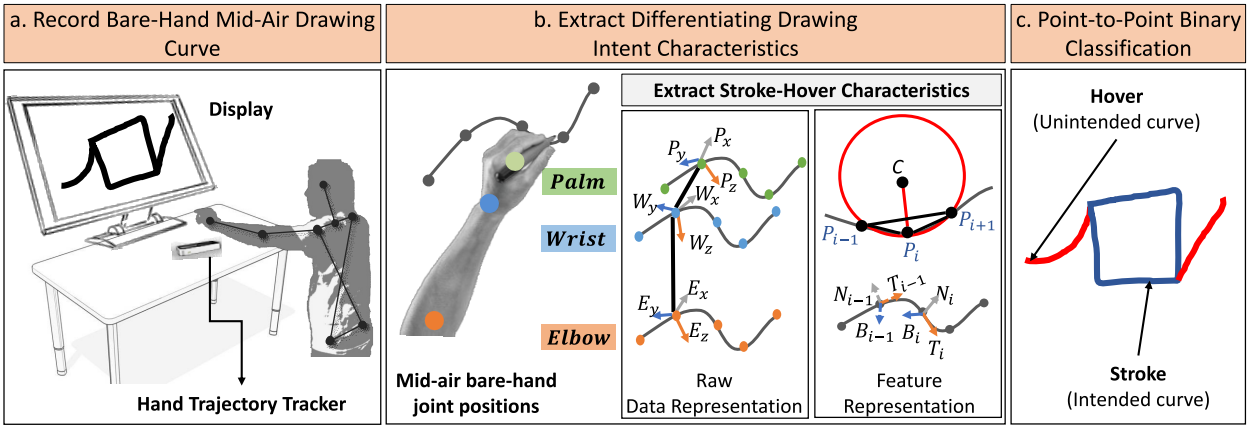

Fig. 1. Stroke-Hover Intent Recognition Workflow.

the user provided curve input). Our goal in this research is to investigate a complementary yet fundamental problem in mid-air curve input: ***determining when users actually intend to draw in mid-air***.

1.1 Problem & Motivation

While drawing on a tablet, a stroke from the user is a natural outcome of merely touching down and subsequently loosing contact from the touch-screen. As a consequence, any analysis of user input (so as to detect a gesture or perform geometric analysis for object and symbol recognition) need only consider the stroke that was captured through the touch device. In mid-air interactions, there are currently two methods to distinguish a stroke from a hover: (a) instrumented controllers wherein a user uses a physical control such as a button to express intent to register a stroke and (b) hand gestures such as pinching or pointing to prescribe the start and end of a stroke. Instrumented controllers are intrusive and constrained by the capabilities offered by the controller. On the other hand, while bare-hand interactions are non-intrusive, the use of prescribed hand postures for a task as simple as drawing is not particularly intuitive. In our prior work [9], we conducted a behavioral study to understand user preferences with using predefined hand postures while drawing in mid-air. We observed that sketching using predefined hand postures restricts the users in their movements, and in turn can hinder their capability to freely express their intent. Findings from this study strongly indicated the need for an approach that processes mid-air curves simply using posture-less, bare hand trajectories.

Specifically for sketching tasks (and similar tasks such as writing, stroke-based gestures), most works either segment user trajectory after the fact or somehow build the segmentation within a symbol identification system [8]. In any case, this necessitates a vocabulary of a finite set of symbols suggesting a top-down approach for pattern recognition (i.e. looking for known patterns). There is currently no means for detecting just the intention to make a *mark on the screen* and building a coherent shape or a symbol in a bottom-up manner.

In a future where spatial interactions will be as commonplace as touch interfaces are today, we argue that the ideal interaction scenario for mid-air curve input would be one where the user draws on the screen (or in 3D volume) in the same way we describe a shape or give directions to another human being — i.e. without holding a device or consciously focusing on using a specific hand pose prescribed by an interaction designer. In order to achieve natural mid-air communication, we propose that the recognition of signs and symbols should be separated from detecting the user's intent to (or not to) draw. By understanding the *stroke-hover* intent, it will be possible to generalize mid-air interaction workflows beyond sketching applications to a wider variety of tasks including gesturing, authentication, and object manipulation. The key advantage of this approach is that it will both enable the recognition of drawings and symbols not currently in some pre-defined vocabulary and will still be useful for classical problems such as symbol recognition. It will further allow for scaling mid-air interactions to larger spaces with many users with low computational complexity and high robustness.

1.2 Challenges

Detecting intent to draw in mid-air with no explicit mechanism (hand gesture or controller) is significantly challenging for three primary reasons. First, while a known set of shapes could be distinguished on the basis of their geometry, there is no obvious geometric or statistical distinction between a stroke and a hover in a given mid-air trajectory. Second, variation across users is driven by several parameters including the shapes that are drawn, the speed at which the user draws them, the skill of the user in drawing specific shapes etc. Finally, the dimensionality of the data (the trajectory of some tracked point on the hand) is generally low in contrast to many other classification problems making feature extraction difficult. Determining a a suitable data representation itself is, therefore, a non-trivial task.

1.3 Previous Work

This paper is the next step to our previous work [9], where we formulated mid-air drawing as a point classification problem in contrast to the commonly curve segmentation

problem. We found that *strokes* and *hovers* are characterized by differentiating properties such as curve completion times, speed profiles, and geometric properties. Based on these observations, we derived temporal geometric properties (speed profiles, curvature, change in Frenet frame, etc.) from recorded mid-air curves, and used these features to train a *stroke-hover* classification model to identify the user's drawing intent.

Mid-air drawing data from multiple users was recorded using the GeoMagic Touch device stylus, and binary classifiers identifying *stroke-hover* intent were trained. Bagged decision trees were found to predict the drawing intent with an average accuracy of 76.75% for a variety of shapes including alpha-numeric characters, 2D-3D primitives, and free form planar curves. However, testing the model on on-the-fly data recorded using the Leap Controller pointed towards certain drawbacks, primarily caused due to the physical and spatial limitations associated with the GeoMagic Touch device.

1.4 Current Approach

We posit that drawing in air is not merely an action of the tip of the finger, but an action performed by the whole hand including to the wrist and the elbow. Further, the random forest classifier indicated the speed of the recorded points to be the most discernible feature for *stroke-hover* classification. Based on these observations, we wish to employ a simpler representation of the hand trajectory while leveraging the information captured within the trajectories of multiple joints in the hand to develop a robust *stroke-hover* classifier. We propose a modified *stroke-hover* identification work-flow (Figure 1):

1. *Data Acquisition*: The mid-air drawing data (hand trajectories) is recorded by using the Leap Motion controller in conjunction with an untethered hand-held device to provide the ground truth for drawing intent [1].
2. *Skeletal and Feature Based Data Representation*: From the recorded 3D mid-air drawing data, distinguishing geometric and temporal properties of *strokes* and *hovers* are computed.
3. *Classifier Training & Testing*: Different combinations of these features are used to train binary classification models. During testing, the trained classifier is now used for intent classification for every new point recorded.

1.5 Contributions

We make three main contributions. First, we present a new data representation for recording user strokes in mid-air that utilizes not only the palm but also includes wrist and elbow joints. Unlike most prior works, this representation provides a richer space of skeletal data for analyzing mid-air *stroke-hover* intent and also increases the information input for our classifier. Second, we capture a data-set of a diverse range of signs, symbols, and shapes with both the stroke and hover information. This is unlike almost all other works where only the actual strokes are provided (since the focus is on symbol recognition). Such data could be instrumental in understanding user movement in many other applications. Finally, in order to comprehensively investigate our new three-point trajectory (palm-wrist-elbow), we derive five different data representations and conduct comparative evaluations of these representations to determine the best one to detect the user's *stroke-hover* intent (Section 5). Results from the model trained using this newly recorded data show an improvement in drawing intent prediction accuracy of our classifiers (Section 6). For our evaluation, we divided our data-set into training and testing sub-sets. While this evaluation itself is limited to the same type of testing data as the training data, we believe that our curve categories are sufficiently large and diverse to make a case for the generalizability of our trained models.

2 Related Works

Segmentation of sketched strokes into meaningful objects and components has been studied extensively in literature [10–12]. Many a time, curve inputs are also used as *gestures* by multi-touch interactions [13–17]. However, in mid-air interactions, the planarity of the recorded curve input is not guaranteed. Therefore, existing techniques for determining the meaningful parts of user input are not scalable.

To address this issue, many works have demonstrated techniques for recognition of motion-based gestures [8, 18–20], sketching [1–3], hand-writing [5–7], and development of natural user interfaces using augmented and virtual reality [21]. The key focus in these works is to be able to categorize the user input as a known symbol in an existing vocabulary (e.g. alphanumeric symbols). There are also approaches [22–25] that use hand-posture for detecting user intent for creating mid-air strokes. However, hand pose estimation and skeleton tracking are still areas under progress and are not yet robust enough for free-form tasks such as sketching in design applications. This further poses difficulties in the design and development of mid-air interactions with large displays [26–28] wherein multiple users would be able to collaborate in spatial environments.

Especially for design applications, special devices have been proposed. For instance, Grosman et al. [29, 30] demonstrated physical tape drawing for automotive curve design using hand trackers for hand skeleton detection. Similarly, Laundry et al. [31] describe 3D input techniques for large displays using infrared trackers for robust interactions. Yang et al. [32] describe an augmented reality based technique to train users on manual milling operations, where all user activities are recorded using Leap Motion controller and HTC Vive head mounted displays.

This paper is a part of our continuing research to enable mid-air interactions that do not need a prescribed set of gestures/postures or controllers to allow users to express 3D artifacts. In relation to the previous works, we make three observations that motivate our problem and approach. First, the effectiveness of 2D methods for segmentation and recognition of curve inputs on touch surfaces have not been particularly

[1] All raw data collected in our studies will be shared publicly if the manuscript is accepted.

successful in higher dimensional spaces [8]. Second, there is a need for methods that do not rely on clean and segmented data to compare user input against some pre-defined vocabulary of canonical shapes. To date, most curve recognition techniques use segmented data and cannot handle a continuous stream of 3D points [25, 33]. There are very few works that address this issue using either a sliding window based approach to perform activity recognition with streaming sensor data [34] or dynamic time warping for continuous dynamic identification of gestures [8]. Even in these works, each user input is first segmented and then classified with respect to the templates stored in a database. In contrast to these, our work processes mid-air curve input as sequential points which have individual set of feature vectors, and performs the classification on each point as and when new points are recorded. The final observation motivating this work is that using a spatial device or a hand posture is not natural to users especially for sketching tasks [1]. We take inspiration from Works such as Data Miming [35] and grasp-based virtual pottery [36] that investigate *gesture-free* approaches for enabling users to perform open-ended spatial tasks that are guided by how users interact with the physical world instead of prescribed actions.

3 Data Collection

The first step toward stroke-hover intent detection is the collection of data from users that (a) robustly captures the ground-truth (i.e. classified stroke and hover points) for a variety of mid-air input, (b) is simple enough for a new user to get accustomed to, (c) emulates, as closely as possible, a desktop-based curve drawing interaction, and (d) does not impose cognitive or physical constraints on users while performing drawing tasks. To achieve these goals, we conducted our data collection with an apparatus comprised of a Leap Motion controller, and a custom hand-held wireless device that acts as a 3D pen for the user to draw with. We chose not to use a standard device (HTC Vive or Wii remote) for both, the simplicity of software implementation as well as the form-factor of the hardware.

The Leap motion controller accurately [37] tracks three joints on the user's hand: palm, wrist, and elbow, as described in the following section. The flexible interaction volume of the Leap allows the user to move their hand while drawing freely without physical constraints. Furthermore, Leap motion offers a spatial tracking accuracy of 0.2 mm for static setups and 1.2 mm for dynamic setups [37]. Studies evaluating the spatial resolution and tracking accuracy of the Leap controller recommend this portable device in comparison with other commercially available optical sensors such as Microsoft Kinect and Optotrak [38, 39]. These attributes enable using the Leap motion controller for complex tasks involving finger and joint tracking such as stroke rehabilitation [40], sound synthesis [41], and air painting [42], amongst others. Before conducting our data collection, we first performed tests to ensure that holding the 3D pen does not affect the tracking of the palm, wrist, and elbow joints from the Leap Motion controller. Subsequently, we collected user data across six different shape categories.

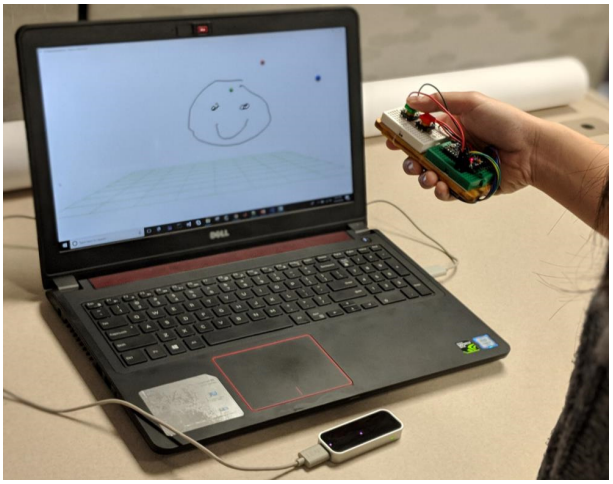

Fig. 2. Data collection setup using the Bluetooth-connected 3D pen and Leap Motion controller.

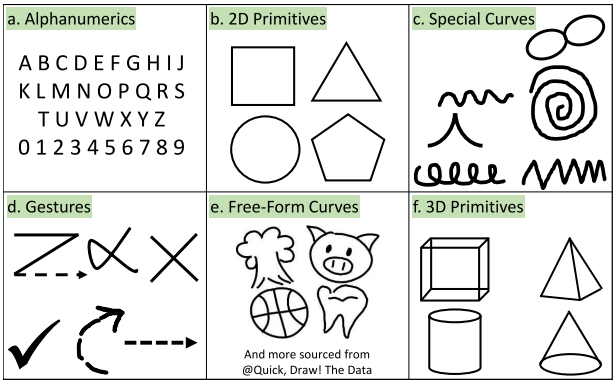

Fig. 3. Alphanumerics (a), 2D primitives (b), special curves (c), motion gestures (d), 2D free-form shapes (e), and 3D primitives (f) drawn by users during the data collection study.

3.1 Data Collection Setup

Our setup (Figure 2) comprises of the following components:

1. Arm Trajectory Tracking: Bare hand interactions in the mid-air involve a series of simultaneous movements of the user's elbow, wrist, and consecutively, palm joints. To incorporate the effect of these movements in the *stroke-hover* intent classifier, the Leap Motion controller is used to record time stamped coordinates of the user's elbow-wrist-palm inclusive three point trajectory. The Leap controller is mounted on the table, and the user draws 3D drawings within the interaction volume of the device.

2. Stroke-Hover Ground-truth Detection: To record the *stroke-hover* intent of a tracked point, a device resembling a 3D hand-held pen was developed. The device comprises of a button, that is pressed by the user every time a *stroke* is to be drawn. The pen communicates with the computer via a Bluetooth HC-06 serial port, and is programmed using an Arduino Pro Mini. At any given point of time, the pen sends a one byte long character string indicating 0-1 status of the button to the computer, suggestive of the button up or

down states, respectively. This information, combined with the 3-point trajectory recorded by the Leap Motion effectively constitute a single data point at any given time.

3. Software: The data recording interface is developed using C++ on the OpenGL platform, and implemented on Intel Core i7-6700HQ CPU running at 2.6GHz, on a Windows 10 Home operating system.

3.2 Participants

For recording the 3D drawing data, 25 engineering students (13 female) within the age range of 19-30 years were recruited. 3 participants had prior experience with 3D sketching (digital), while none of the participants had experience with any computer vision devices like the Leap motion controller, or 3D depth cameras. None of the participants had experience with using hand-held devices for drawing in mid-air.

3.3 Drawing tasks

We begin with the observation that while we aim to provide stroke-hover classification for any possible intended curve the user may wish to draw, drawing without context is meaningless. At the same time, training our classifiers on a narrow set of curves would also defeat the purpose since it will lead to a mere gesture recognition based approach. Therefore, our strategy was two-fold. First, we consider a wide variety of curve types that covers most (if not all) of the basic geometric features (in terms of continuity, curvature, and inflections). Additionally, we also add "*composite curves*" of objects such as trees etc where the individual strokes have no specific meaning in themselves but their composition is a meaningful shape. To this end, our choices of curve types are more based on both geometric as well as semantic features. Our second strategy is that we do **not** train our models with any prior information about the identity of the curves. For example, while the user may be drawing a circle, the model is trained completely ignorant of the fact that it was a circle. This enforces our algorithm to learn only to classify intentional vs. non-intentional points rather than recognizing the identity of the shape.

The data collection tasks were elaborately designed to incorporate multiple geometries, different planarity, and a broad variety of motion and gestural drawings. To ensure semantic variety, shapes belonging to the alphanumeric category, 2D/3D primitives, and planar curves were recorded. Observation from our previous work [9] indicates drawing speed to be one of the major factors affecting the *stroke-hover* classification of a given point. With this in mind, we ensured that the users were asked to draw shapes with minimal complexity (such as letters of alphabet, numbers, primitive geometries, etc.) so that they can solely focus on the task of drawing as naturally as possible in mid-air. 3D primitives were recorded to ensure the trained classifier is able to capture *stroke-hover* differences irrespective of the dimensionality of recorded curves. Each participant recorded data across 6 different shape categories (Figure 3) as follows:

1. **Alpha-numeric Characters** These curves include 26 letters of the English alphabet, and numbers.
2. **2D Primitives** These curves include 2D primitives such as the circle, triangle, square, and pentagon. To understand whether the size of a drawn shape affects the *stroke-hover* classification accuracy of the models, the participants were asked to draw the shapes in three sizes: small, medium, and large. All shapes were drawn on the front, top, and right planes to ensure that the effect of planarity on the *stroke-hover* intent is captured.
3. **Special Curves** To ensure the trained classier is robust towards varying geometric properties of the drawn curves, the participants were asked to draw curves with special geometric properties (degree 2, degree 3 polynomials) and features (inflexion points, repetitive waveforms, etc).
4. **Stroke Gestures** These curves involved the participants using the 3D pen to "draw" motion gestures in mid air, such as swipe, check mark, cross mark, etc. These set of curves are recorded to validate the utility of *stroke-hover* classification approach towards symbol/gesture recognition tasks. Part of these symbols are referred from the $1/P-recognizers family [14, 15].
5. **Planar Free-Form Doodles** This session allowed the participants to draw free form planar shapes on the front plane from Google's Quick, Draw! [43] database, such as a ball, cat, tree, etc. These set of curves are recorded to ensure that the *stroke-hover* classification model is trained using free-form data that generalizes the classifier beyond known shape primitives, symbols, and figures.
6. **3D Primitives** In comparison to the other shape categories, here, the participants were not restricted to a given plane, and were allowed to draw 3D primitives such as the cube, frustum, and cylinder, within the entire interaction volume of the Leap controller.

For each shape across the six categories, participants recorded data through three trials. Each shape was recorded once per session. No visual feedback about the mid-air curve input was provided to the participants, and instead, they were instructed to draw the curves as naturally as they could (i.e. as fast or as slow as they would if there were no interface). The hand trajectory tracking was monitored by the study proctor, and whenever the participants went out of the tracking zone, they were instructed by the proctor accordingly. After every session, the participants were allowed a break to ensure that hand fatigue does not bias the recorded data. It is assumed that the 2D data sketched by users is primarily planar, and can be drawn using single strokes or multiple strokes. The 3-dimensional primitives recorded are multi-planar and multi-stroke curves. For every curve drawn, the interface records a continuous sequence of 3D coordinates of the user's elbow $E_{x,y,z}$, wrist $W_{x,y,z}$, and the palm position $P_{x,y,z}$, and the classification of that point as being *stroke*(1), or *hover*(0). Along with the curve coordinates and classification status, the time stamp of all drawn points is recorded in milliseconds t.

4 Preliminary Data Analysis

In order to better design our representations and models, we conduct preliminary analysis on the recorded data. We identify differentiating visual and motion profile properties for the *stroke-hover* curves across six shape categories, and use these insights to better design our feature representations.

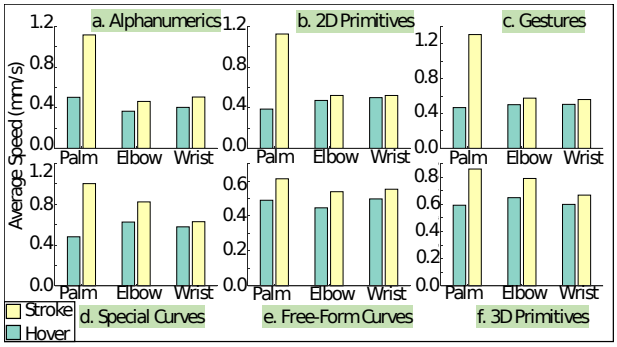

Fig. 4. Average stroke-hover speeds of the palm-wrist-elbow trajectory for all drawn shapes across the six shape categories. Speed is in mm/s.

4.1 Visual Profiles

Visual analysis of the spatial profiles for the palm, wrist, and elbow trajectory indicate the following aspects. When drawing any given shape mid-air, the palm point travels maximum distance, while the elbow trajectory is observed to be almost stationary. It is observed that participants typically used the elbow point as a pivot, and manipulated the wrist and palm to draw shapes – similar to how a person draws on a paper, or a tablet. Spatial variations in the elbow trajectory are observed however, when 3D primitives are drawn.

4.2 Speed Profiles & Direction

Overall *stroke-hover* speed profiles are different from as observed in the previous data recording study conducted using the GeoMagic Touch haptic device [9]. Here, on an average, strokes are traversed faster than hovers. This variation can be attributed to the additional spatial freedom of the palm-wrist joints in the new setup. Further, reduction in traversal speeds are observed along the transition points. Variations in speed profiles are observed across different shape categories (Figure 4). Alphanumeric characters, due to their familiarity, are traversed fastest (average speed = $0.9mm/s$), while free-form shapes are drawn the slowest (average speed = $0.624mm/s$). Also, as was observed in the behavioral study described in our previous work [9], *stroke* curves possess higher curvature (average curvature = 0.096) than *hover* curves, indicating that *hovers* are typically traversed in straight lines.

5 Stroke-Hover Classifier Training

To train a stroke-hover classifier using the recorded data, we first address two important questions: (a) how can we model information from the three trajectories representing the user's hand motion when drawing in mid-air to extract relevant *stroke-hover* features, and (b) using these representations, how can we train different classification models? We address these two aspects in the section below.

5.1 Geometric Feature Based Classification

Based on the mid-air drawing intent classification results discussed in our previous work [9], using the tracked palm, wrist, and elbow points, for every recorded point i, we extract geometric features given by $G_i = [s_i\ a_i\ j_i\ c_i\ S_r\ \omega_\alpha\ \omega_\beta]$.

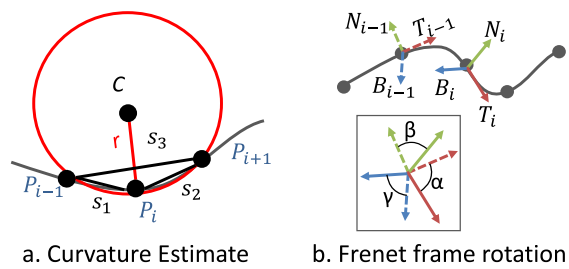

Fig. 5. Estimated curvature and discrete Frenet frames of the recorded curve.

Here, s_i, a_i, j_i are the speed, acceleration, and jerk relative to the previous point; S_r is the ratio between speed of successive points; c_i is the local curvature; and ω_α, ω_β, and ω_γ represent the change in planarity of the recorded mid-air curve input. These geometric features are used to construct the following models with the recorded data:

Palm-Point Geometric G_1: Using the time-stamped coordinate of the palm trajectory and the geometric features above, we train a classifier using the 8-dimensional geometric feature vector extracted from the recorded palm trajectory, $G_1 = \left[s_{pi}\ a_{pi}\ j_{pi}\ c_{pi}\ S_{pr}\ \omega_{p\alpha}\ \omega_{p\beta}\ \omega_{p\gamma}\right]$.
Palm-Wrist-Elbow-Point Geometric G_2: Along with considering the point-to-point geometric variations in the palm trajectory(G_p), we consider the wrist(G_w) and elbow (G_e) points too, and construct a 24-dimensional geometric feature vector given by $G_2 = \left[G_p\ G_w\ G_e\right]$.

5.2 Differential Coordinate-Based Classification

Geometric features described in the previous section are a derived representation of the nature of *stroke-hover* 3D data recorded. In this section, we explore models constructed from the differential coordinates of the hand trajectory. This is based on our prior experiments where the velocity turned out to be the most discriminating feature of the trajectories. For this, the palm-wrist-elbow data is recorded as a time stamped trajectory of sequential points. To extract differentiating *stroke-hover* characteristics from the recorded data, we use the raw time-series coordinates to train classifier models. That is, the observation at time step t_i is represented as the difference with respect to observation at instance t_{i-1}. This

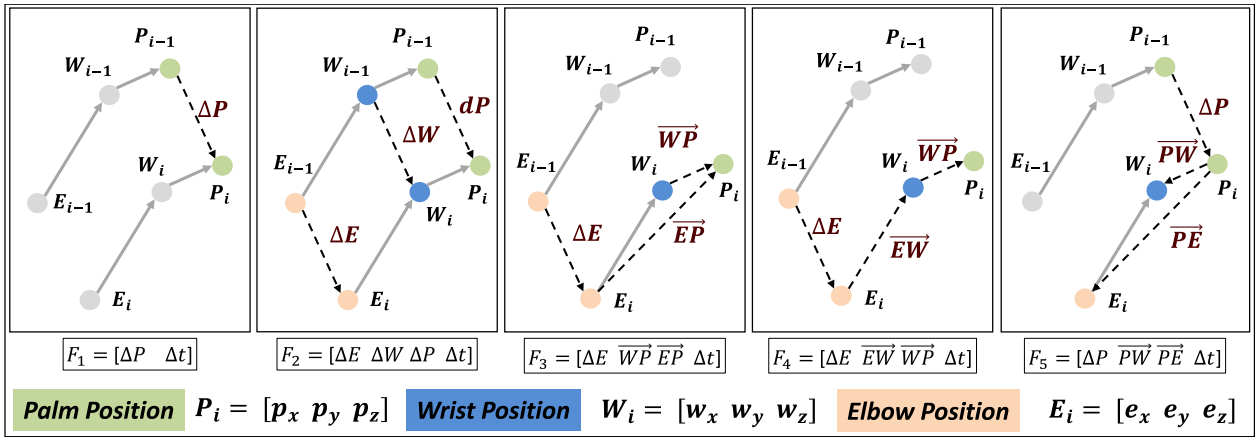

Fig. 6. One-point and three-point raw data based local differential representations extracted from palm-wrist-elbow trajectories of the user drawn shapes.

removes the trend and the resultant difference series represents the changes in observations, in this case, changes in the 3D trajectories of the elbow, wrist, and palm when the user draws a shape mid-air. Five different mid-air motion representations of the palm, wrist, and elbow trajectories are constructed (Figure 6) as follows:

One Point Representation: This is the simplest 4 dimensional motion representation (F_1) constructed using difference in the time stamped palm coordinates recorded for each shape.

Three Point Representations: Using the 3-point tracking data, we embody the multi-joint motion when drawing in mid-air through different representations of the bio-mechanical link system (Figure 6). While representation F_2 encodes the joint information through a simple difference between the palm, elbow, and wrist positions between any two given instances, F_3 and F_4 encode the motion using representations with elbow as the pivot or reference position for other joints in the bio-mechanical link. Finally, F_5 considers a representation with the palm as the reference point for embedding the consecutive wrist and elbow movements in 3D space when a user draws.

6 Classifier Training & Testing

6.1 Data Distribution

A total of 165,000 (70,813 *stroke*, 94,187 *hover*) time-stamped data points were recorded across 6 shape categories, with three trials each, from all participants. All data recorded is randomized to ensure that the training and testing data sets do not comprise of consecutively recorded shapes from similar categories by a single user. Of the total recorded data, 114,270 points (50,786 *stroke*, 63,481 *hover*) are used for training the classifiers, while the remaining 50,730 points (25,024 *stroke*, 25,076 *hover*) are used for testing and cross-validation (10-fold cross validation). Further, while training, the feature vectors are randomly sampled from the available set to eliminate over-fitting of the trained model due to consecutive data points.

6.2 Classifier Selection & Optimization

We use random forests to predict the *stroke-hover* intent from the recorded 3D data. In total, we trained seven random forest classifiers, two for the feature based (G_1 and G_2) representations, and five for the raw data based representations (F_1 to F_5). Our previous experiments showed that the classification accuracy of each recorded point depended more on the number of decision trees than the depth of the trees. Based on this, our goal was to first find the optimum number of decision trees that would result in the best classification accuracy for each data representation individually. Given any representation, we start with default values of $N_t = 10$, and the number of trees per forest are increased iteratively until $N_t = 150$. To counter the effects of *stroke-hover* imbalance in the data, a weighted cost matrix is used, with $w_{hover} = 1$ and $w_{stroke} = 1.15$. The results discussed in section 7 are obtained from these optimized classifiers, for each trajectory representation.

6.3 Evaluation Metrics

All classification models are evaluated on the basis of their prediction accuracy, using the remaining 30% split of data. Along with standard prediction accuracy (η) computed for every individual point recorded on the curve, we also compared the *precision* ($TP/(TP+FP)$) and *recall* ($TP/(TP+FN)$) where TP, TN, FP, and FN denote true positives, true negatives, false positives, and false negatives respectively.

6.4 Software Architecture

Our *stroke - hover* classification model can be described as comprising of following three main modules (Figure 7), namely, (1) the data recording visual interface, (2) the training module that was used to train the classifier based on pre-recorded user input, and (3) the testing module wherein we test the learnt models based on new user input.

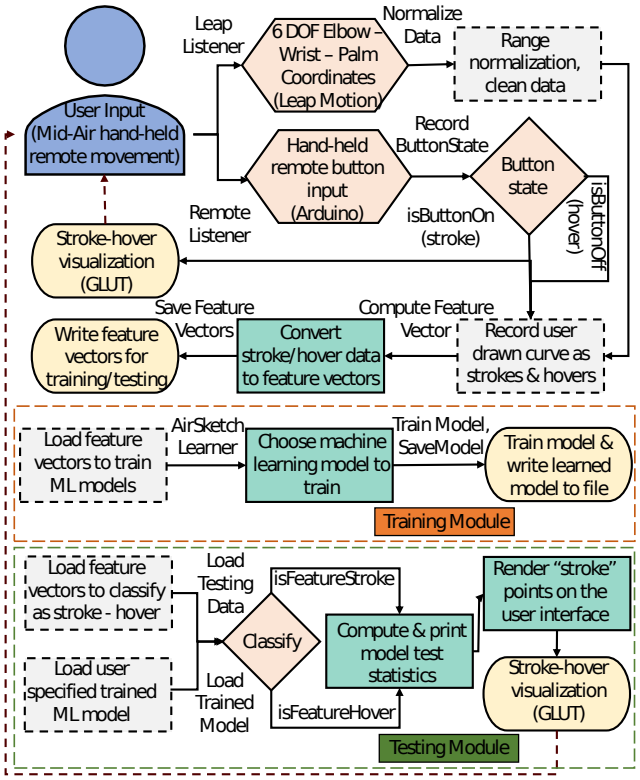

Fig. 7. Software architecture for stroke-hover classification data collection user interface, training, and testing modules.

6.4.1 Data Recording Visual Interface

This is the visual interface built on OpenGL where the user records mid-air curves using the Leap Motion controller and hand-held remote. Hand coordinates from the Leap Motion controller are normalized and converted into geometric and differential coordinate based feature vectors. This module allows the processed feature vectors to be saved into a file that can be re-used later for training/testing purposes.

During the data collection study, data from the hand-held remote was used to record users' *stroke - hover* intent. All data recorded as *stroke* was displayed on the screen using GLUT libraries.

6.4.2 Training Module

The training module - *Airsketch Learner*, comprises of functions to load feature vectors, and use machine learning models built using OpenCV to train the feature vectors. Model learning parameters (such as number of trees, depth per tree, etc. for random forests) are specified when calling the training functions. The learned model is then stored on a file to be used later for testing purposes.

6.4.3 Testing Module

This module allows to test user-recorded data using the trained learning models. Each feature vector is classified into *stroke* and *hover* points. The predicted user-intended curve is displayed as *stroke* points on the OpenGL interface, using GLUT functions.

7 Results & Observations

In this section, the prediction accuracy for binary classifiers trained using different curve data representations are discussed, and the best model is identified.

7.1 Geometric Feature Based Classification

The classifiers trained using features G_1 and G_2 predict *stroke-hover* with an accuracy of $\eta = 72.33\%$ and $\eta = 73.1\%$ respectively. Testing accuracy from the palm-point feature G_1 align closely with the single point model discussed in our previous models [9], and exhibit a precision-recall rate of 0.659 and 0.616. The three point geometric feature G_2 predicts with a slightly better accuracy, however, both models exhibit relatively high false negatives.

7.2 Skeleton-Based Classification

7.2.1 One Point Representation

With a 4-dimensional feature representation of the palm points(F_1), the classifier results in a training accuracy of $\eta = 70.56\%$ and an average test accuracy of $\eta_{F1} = 74.07\%$. The classification results show a significant improvement in comparison with the real-time bare-hand classification results obtained from the model trained on data collected using the haptics device [9]. The results however, indicate a high degree of false negatives, with a precision-recall distribution of 0.733 and 0.692 respectively. This presses the need for high dimensional representation of the hand motion, results of which are discussed next.

7.2.2 Three Point Representation

The four feature representations based on the palm-wrist-elbow trajectories are trained and tested using a similar data split, as described. Classifier trained using motion data representation F_2 predicts with a test accuracy $\eta_{F2} = 73.91\%$, whereas those with representations F_3 and F_4 have average accuracy equal to $\eta_{F3} = 74.23\%$ and $\eta_{F4} = 74.56\%$ respectively. Classifier trained using F_5 on the other hand predicts with an average accuracy of $\eta_{F5} = 79.87\%$ (Figures 8(a), 8(b), 9(b)). Further analysis of prediction accuracy across different shape categories indicates that best results are obtained for motion gestures (Figure 8(c)) and special curves (Figure 9(a)), with $\eta = 80.33\%$ and $\eta = 81.36\%$ respectively. With an out-of-bag error of 0.21 and differences in the palm position (ΔP_i) as the most dominating *stroke-hover* identifier, raw data representation $F_5 = \left[\Delta P \ \vec{PW} \ \vec{PE} \ \Delta t\right]$ is identified as the most accurate classifier for our *stroke-hover* classification problem.

7.3 Bare Hand Drawing Predictions

To test the best model with real-world data, we recorded bare hand drawings across the six shape categories using the Leap controller. Participants were instructed to draw shapes in mid-air within the interaction volume of Leap using their bare hand movements *without any visual feedback*. To observe how hand motion data representation affects the *stroke-hover* predictions, the recorded data was tested using one point and three point geometric feature based classifiers (G_1, G_2), and

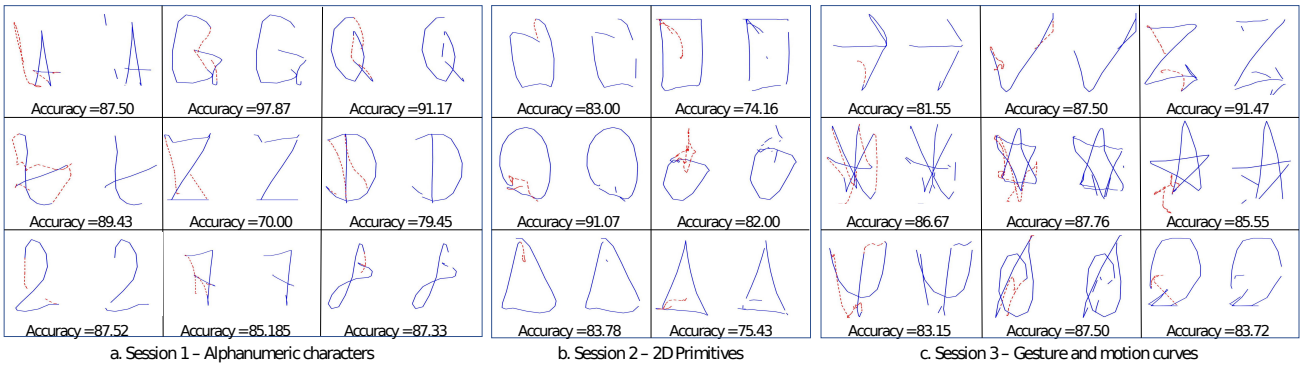

Fig. 8. Random Forest classifier predictions using raw data based representation F5 for (a) alphanumeric sketch data recorded during Session 1, (b) 2D primitives recorded during Session 2, and (c) gesture and motion curves recorded during Session 3.

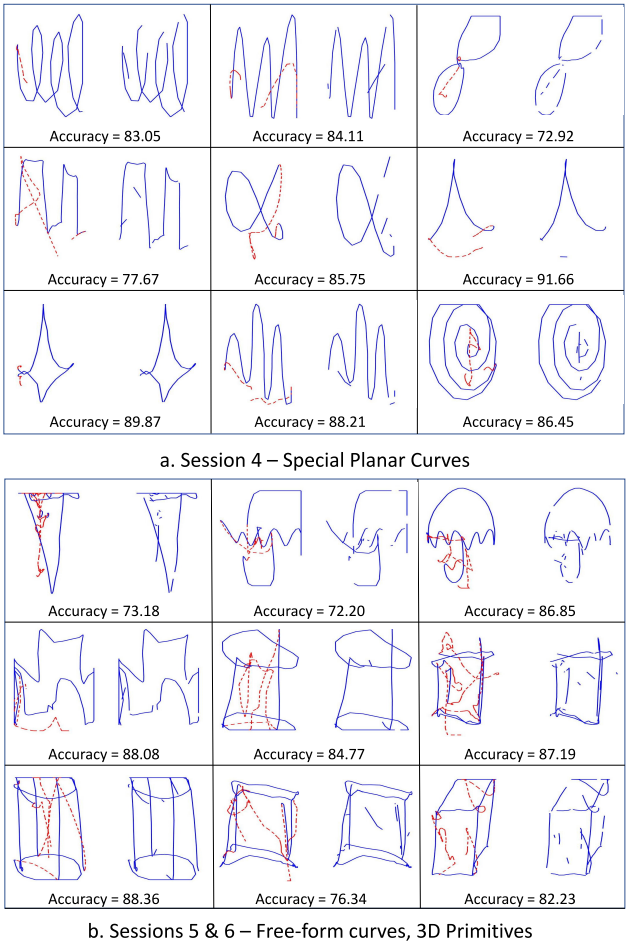

Fig. 9. Random Forest classifier predictions using raw data based representation F5 for (a) special planar curves recorded during Session 4 and (b) free-form planar curves and 3D primitives recorded during Sessions 5 and 6 respectively.

raw hand motion representations based classifiers (F_1, F_5). The three point motion representation classifier (F_5) exhibits the best prediction results with the minimum false negatives in comparison to other representations.

Prediction of the drawn curves using these four classifiers points towards two important results: a) detecting the intent for drawing improves by considering multi-point skeletal trajectories (G_1, F_1 and G_2, F_5), and b) training classifiers on differences captures *stroke-hover* characteristics better than derived geometric representations. It is important to note that despite displaying improved results in comparison to the bare hand predictions obtained in the previous work [9], the differences in the hand posture when drawing using the hand-held remote versus drawing bare-hand reflect in the false positives exhibited by the best model predictions (F_5).

7.4 Limitations

Despite eliminating spatial restrictions associated with the GeoMagic Touch device, the basic premise of recording data using the Leap controller involved using a hand-held device for recording the *stroke-hover* intent in mid-air. While the setup was designed to closely replicate how users typically draw in mid-air, the palm and wrist trajectories differ in the two scenarios, which is reflected in the bare-hand Leap data predictions (Figure 10). In future studies, we plan to eliminate dependencies on hand-held devices by using non-invasive intent recognition techniques such as finger-tap sensors. The biased false negatives of the model dictates the need for identifying critical geometric and temporal properties associated with *strokes*, which would result in more accurate predictions. Moreover, the data recorded in both studies involves asking the user to create *drawings* instead of *strokes*. In future, we plan to build a strokes-only data-set and use the current approach towards identifying *strokes* as an anomaly detection problem. Further, mid-air curve input speed is dependent on other factors like the type of curves, preciseness with which they are drawn, and applications for the 3D curve input. In future studies, we plan to record a comprehensive data-set with focused participant groups (arts, architecture, designers), and through drawing tasks designed in lieu of an end application. Our testing and training samples are sourced from the same data-set. Therefore, the generalizability of our methods needs to be evaluated further. Given that our curve categories are sufficiently large and diverse (ranging from primitives, curves with specific geometric properties, real-world sketches etc), we believe that our trained models can handle general curves. However, further studies are required to systematically test

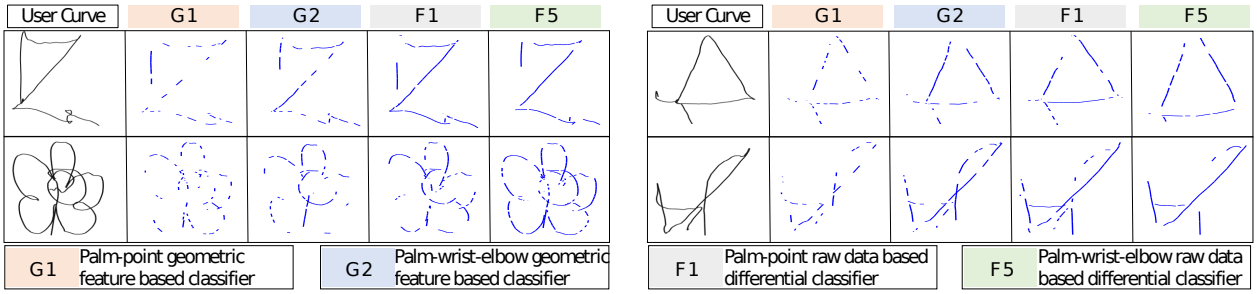

Fig. 10. Bare hand mid-air drawing data recorded using the Leap controller predicted using four different binary classifiers: 1-point geometric, 3-point geometric, 1-point differential, and 3-point differential.

	Different Data and Feature Representations						
	Geometric Features		Local Differential Features				
Feature	**G1**	**G2**	**F1**	**F2**	**F3**	**F4**	**F5**
Accuracy	72.23	73.1	74.07	73.91	74.23	74.56	79.87
Precision	0.659	0.679	0.733	0.71	0.693	0.717	0.8166
Recall	0.616	0.627	0.692	0.605	0.614	0.611	0.723

Fig. 11. Average accuracy, precision, and recall for test predictions using all combinations of geometric and differential feature vectors.

the generalizability of our methods.

8 Discussion

8.1 Hand Representation: From Palm to Elbow

The one point motion representation model (F_1) trained using the time stamped palm coordinate predicted curves with an average accuracy much lower than the models trained using three point motion representations (F_2 to F_5, Figure 11). This shift can also be clearly seen in the reduction in false positives and false negatives, as one moves from one point to three point models - both, for feature based as well as raw geometric motion representations (Figure 10). These variations suggest that apart from the palm trajectory, important *stroke-hover* information can be derived from the additional wrist and elbow joints. It further indicates that though the overall movement of the wrist and elbow joints is smaller when compared to the palm, inclusion of higher dimensional data for drawing intent classification task is an important aspect. In future studies, it would be worthwhile to study how upper body skeleton movements affects the classification models.

8.2 Trajectory Representation: Features vs Raw Motion Data

In line with previous work [9], using both one point and three point data, we extracted important geometric features from the recorded trajectories. While these geometric features based classifiers are able to identify hovers, they are characterized with a low precision-recall rate. However, using simple local differences of the palm-wrist-elbow trajectories exhibited high prediction accuracy as well as reasonable precision-recall. That is, training the classifier on the variations of spatial coordinates of the time stamped hand-trajectories proved to be a good indicator of identifying a user's *stroke-hover* intent. This observation points to an important question: is there a way to extract *hidden characteristics* from *strokes* and *hovers*? It would be interesting to model the drawing intent classification problem using some lower dimensional embedding of the raw data through data reconstruction algorithms such as autoencoders.

8.3 Device Form Factor: Pen vs. Remote

Training a classifier for drawing intent recognition task is a supervised learning problem — that is, training our models mandated the recording of ground-truth. In this work, we used the hand-held device with a Leap controller tracking the user's elbow-wrist-palm while they draw in mid-air. While this setup was designed to be as close as possible to the way people naturally would draw in mid-air, the inclusion of the hand-held device caused some variations in the wrist trajectories for both cases. The hand movements observed when using an instrumented controller, as compared to bare hand movements, are different. This explains the false positives and false negatives observed while predicting bare hand mid-air data recorded using the Leap controller. To ensure similarity between the trained classifier and its target application areas (say, mid-air drawing), it is necessary to use *stroke-hover* tracking mechanisms that are minimally invasive to the way users draw naturally in mid-air.

8.4 Robustness Towards False Negatives

Bare hand data tested using all models proposed in this paper exhibit a certain degree of false negatives. These are typically localized in areas of high curvature, or high speed transitions. Bare hand mid-air drawing involves 6-DOF movements of the palm-wrist-elbow joint-link structure, ranging from simple translation, rotation about the joints, tilting, etc. Thus, along with recording the 3D coordinates of skeletal joints, it is necessary to record aspects associated with such movements using sensors such as accelerometers and gyroscopes. A model trained on such comprehensive data is expected to help improve the *stroke-hover* prediction metrics.

9 Future Directions & Conclusions

In this paper, we presented a rigorous investigation of the motion trajectory representation of human movement in the

context of gesture-free mid-air curve input. The fundamental problem we addressed is that of *stroke-hover* classification. Our representation based on the differential coordinates for the palm-wrist-elbow configuration proved to provide the best accuracy in comparison to the previous results. Experiments with palm-point and upper-arm skeleton data representations further suggested that the use of additional skeletal information is key to understand fundamental actions such as drawing-intent in contrast to higher level actions such as pre-designed gestures.

We envision the integration of a stroke-hover detection sub-system in future spatial user interfaces (SUIs) for 3D modeling and design. To this end, we believe that our method can be combined with existing trajectory and gesture recognition engines to make spatial 3D modeling more robust. There are several challenges that still need to be addressed before this is achieved. Given that a large part of the current stroke-gesture recognition dwells on the issue of segmenting mid-air trajectory or determining meaningful portions of the trajectory, our work can serve as an intermediate step towards improving known frameworks for recognition of spatial symbols and gestures.

Acknowledgements

This work was supported by the startup funds provided by the Texas A&M Engineering Experiment Station (TEES) and the J. Mike Walker '66 Department of Mechanical Engineering at Texas A&M University.

References

[1] Chen, Y., Liu, J., and Tang, X., 2008. "Sketching in the air: a vision-based system for 3d object design". In Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on, IEEE, pp. 1–6.

[2] Arora, R., Kazi, R. H., Anderson, F., Grossman, T., Singh, K., and Fitzmaurice, G. W., 2017. "Experimental evaluation of sketching on surfaces in vr.". In CHI, pp. 5643–5654.

[3] Taele, P., 2014. "Intelligent sketching interfaces for richer mid-air drawing interactions". In CHI'14 Extended Abstracts on Human Factors in Computing Systems, ACM, pp. 339–342.

[4] Aslan, I., Uhl, A., Meschtscherjakov, A., and Tscheligi, M., 2014. "Mid-air authentication gestures: an exploration of authentication based on palm and finger motions". In Proceedings of the 16th International Conference on Multimodal Interaction, ACM, pp. 311–318.

[5] Schick, A., Morlock, D., Amma, C., Schultz, T., and Stiefelhagen, R., 2012. "Vision-based handwriting recognition for unrestricted text input in mid-air". In Proceedings of the 14th ACM international conference on Multimodal interaction, ACM, pp. 217–220.

[6] Vikram, S., Li, L., and Russell, S., 2013. "Writing and sketching in the air, recognizing and controlling on the fly". In CHI'13 Extended Abstracts on Human Factors in Computing Systems, ACM, pp. 1179–1184.

[7] Agarwal, C., Dogra, D. P., Saini, R., and Roy, P. P., 2015. "Segmentation and recognition of text written in 3d using leap motion interface". In Pattern Recognition (ACPR), 2015 3rd IAPR Asian Conference on, IEEE, pp. 539–543.

[8] Taranta II, E. M., Samiei, A., Maghoumi, M., Khaloo, P., Pittman, C. R., and LaViola Jr., J. J., 2017. "Jackknife: A reliable recognizer with few samples and many modalities". In Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems, CHI '17, ACM, pp. 5850–5861.

[9] Bohari, U., Chen, T.-J., et al., 2018. "To draw or not to draw: Recognizing stroke-hover intent in non-instrumented gesture-free mid-air sketching". In 23rd International Conference on Intelligent User Interfaces, ACM, pp. 177–188.

[10] Noris, G., Sỳkora, D., Shamir, A., Coros, S., Whited, B., Simmons, M., Hornung, A., Gross, M., and Sumner, R., 2012. "Smart scribbles for sketch segmentation". *Comput. Graph. Forum,* **31**(8), Dec., pp. 2516–2527.

[11] Paulson, B., and Hammond, T., 2008. "Paleosketch: Accurate primitive sketch recognition and beautification". In Proceedings of the 13th International Conference on Intelligent User Interfaces, IUI '08, ACM, pp. 1–10.

[12] Field, M., Gordon, S., Peterson, E., Robinson, R., Stahovich, T., and Alvarado, C., 2010. "The effect of task on classification accuracy: Using gesture recognition techniques in free-sketch recognition". *Computers & Graphics,* **34**(5), pp. 499–512.

[13] Bae, S.-H., Balakrishnan, R., and Singh, K., 2008. "Ilovesketch: As-natural-as-possible sketching system for creating 3d curve models". In Proceedings of the 21st Annual ACM Symposium on User Interface Software and Technology, UIST '08, ACM, pp. 151–160.

[14] Anthony, L., and Wobbrock, J. O., 2010. "A lightweight multistroke recognizer for user interface prototypes". In Proceedings of Graphics Interface 2010, GI '10, Canadian Information Processing Society, pp. 245–252.

[15] Anthony, L., and Wobbrock, J. O., 2012. "$ n-protractor: A fast and accurate multistroke recognizer". In Proceedings of Graphics Interface 2012, GI '12, Canadian Information Processing Society, pp. 117–120.

[16] Wobbrock, J. O., Wilson, A. D., and Li, Y., 2007. "Gestures without libraries, toolkits or training: A $1 recognizer for user interface prototypes". In Proceedings of the 20th Annual ACM Symposium on User Interface Software and Technology, UIST '07, ACM, pp. 159–168.

[17] Vatavu, R.-D., Anthony, L., and Wobbrock, J. O., 2012. "Gestures as point clouds: A $p recognizer for user interface prototypes". In Proceedings of the 14th ACM International Conference on Multimodal Interaction, ICMI '12, ACM, pp. 273–280.

[18] Suryanarayan, P., Subramanian, A., and Mandalapu, D., 2010. "Dynamic hand pose recognition using depth data". In Proceedings of the 2010 20th International Conference on Pattern Recognition, ICPR '10, IEEE Computer Society, pp. 3105–3108.

[19] Arandjelović, R., and Sezgin, T. M., 2011. "Sketch recognition by fusion of temporal and image-based features". *Pattern Recognition,* **44**(6), pp. 1225 – 1234.

[20] Willems, D., Niels, R., van Gerven, M., and Vuurpijl, L., 2009. "Iconic and multi-stroke gesture recognition". *Pattern Recognition,* **42**(12), pp. 3303 – 3312. New Frontiers in Handwriting Recognition.

[21] Regazzoni, D., Rizzi, C., and Vitali, A., 2018. "Virtual reality applications: guidelines to design natural user interface". In ASME 2018 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference, American Society of Mechanical Engineers Digital Collection.

[22] Melax, S., Keselman, L., and Orsten, S., 2013. "Dynamics based 3d skeletal hand tracking". In Proceedings of Graphics Interface 2013, GI '13, Canadian Information Processing Society, pp. 63–70.

[23] Sharp, T., Keskin, C., Robertson, D., Taylor, J., Shotton, J., Kim, D., Rhemann, C., Leichter, I., Vinnikov, A., Wei, Y., Freedman, D., Kohli, P., Krupka, E., Fitzgibbon, A., and Izadi, S., 2015. "Accurate, robust, and flexible real-time hand tracking". In Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems, CHI '15, ACM, pp. 3633–3642.

[24] Tagliasacchi, A., Schröder, M., Tkach, A., Bouaziz, S., Botsch, M., and Pauly, M., 2015. "Robust articulated-icp for real-time hand tracking". In Proceedings of the Eurographics Symposium on Geometry Processing, SGP '15, Eurographics Association, pp. 101–114.

[25] Ren, Z., Meng, J., and Yuan, J., 2011. "Depth camera based hand gesture recognition and its applications in human-computer-interaction". In 2011 8th International Conference on Information, Communications Signal Processing, IEEE, pp. 1–5.

[26] Ni, T., Schmidt, G. S., Staadt, O. G., Livingston, M. A., Ball, R., and May, R., 2006. "A survey of large high-resolution display technologies, techniques, and applications". In Proceedings of the IEEE Conference on Virtual Reality, VR '06, IEEE Computer Society, pp. 223–236.

[27] Czerwinski, M., Robertson, G., Meyers, B., Smith, G., Robbins, D., and Tan, D., 2006. "Large display research overview". In CHI '06 Extended Abstracts on Human Factors in Computing Systems, CHI EA '06, ACM, pp. 69–74.

[28] Lischke, L., Grüninger, J., Klouche, K., Schmidt, A., Slusallek, P., and Jacucci, G., 2015. "Interaction techniques for wall-sized screens". In Proceedings of the 2015 International Conference on Interactive Tabletops & Surfaces, ITS '15, ACM, pp. 501–504.

[29] Grossman, T., Balakrishnan, R., Kurtenbach, G., Fitzmaurice, G., Khan, A., and Buxton, B., 2002. "Creating principal 3d curves with digital tape drawing". In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, CHI '02, ACM, pp. 121–128.

[30] Grossman, T., Balakrishnan, R., Kurtenbach, G., Fitzmaurice, G., Khan, A., and Buxton, B., 2001. "Interaction techniques for 3d modeling on large displays". In Proceedings of the 2001 Symposium on Interactive 3D Graphics, I3D '01, ACM, pp. 17–23.

[31] Laundry, B., Masoodian, M., and Rogers, B., 2010. "Interaction with 3d models on large displays using 3d input techniques". In Proceedings of the 11th International Conference of the NZ Chapter of the ACM Special Interest Group on Human-Computer Interaction, CHINZ '10, ACM, pp. 49–56.

[32] Yang, C.-K., Chen, Y.-H., Chuang, T.-J., Shankhwar, K., and Smith, S., 2019. "An augmented reality-based training system with a natural user interface for manual milling operations". *Virtual Reality*, pp. 1–13.

[33] Dominio, F., Donadeo, M., and Zanuttigh, P., 2014. "Combining multiple depth-based descriptors for hand gesture recognition". *Pattern Recogn. Lett.,* **50**(C), Dec., pp. 101–111.

[34] Krishnan, N. C., and Cook, D. J., 2014. "Activity recognition on streaming sensor data". *Pervasive Mob. Comput.,* **10**, Feb., pp. 138–154.

[35] Holz, C., and Wilson, A., 2011. "Data miming: Inferring spatial object descriptions from human gesture". In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, CHI '11, ACM, pp. 811–820.

[36] Vinayak, and Ramani, K., 2015. "A gesture-free geometric approach for mid-air expression of design intent in 3d virtual pottery". *Computer-Aided Design,* **69**, pp. 11 – 24.

[37] Weichert, F., Bachmann, D., Rudak, B., and Fisseler, D., 2013. "Analysis of the accuracy and robustness of the leap motion controller". *Sensors,* **13**(5), pp. 6380–6393.

[38] Okazaki, S., Muraoka, Y., and Suzuki, R., 2017. "Validity and reliability of leap motion controller for assessing grasping and releasing finger movements". *J Ergon Technol,* **17**, pp. 32–42.

[39] Niechwiej-Szwedo, E., Gonzalez, D., Nouredanesh, M., and Tung, J., 2018. "Evaluation of the leap motion controller during the performance of visually-guided upper limb movements". *PloS one,* **13**(3).

[40] Khademi, M., Mousavi Hondori, H., McKenzie, A., Dodakian, L., Lopes, C. V., and Cramer, S. C., 2014. "Free-hand interaction with leap motion controller for stroke rehabilitation". In CHI '14 Extended Abstracts on Human Factors in Computing Systems, CHI EA '14, Association for Computing Machinery, p. 1663–1668.

[41] Hantrakul, L., and Kaczmarek, K., 2014. "Implementations of the leap motion device in sound synthesis and interactive live performance". In Proceedings of the 2014 International Workshop on Movement and Computing, pp. 142–145.

[42] Sutton, J., 2013. "Air painting with corel painter freestyle and the leap motion controller: a revolutionary new way to paint!". In *ACM SIGGRAPH 2013 Studio Talks*. pp. 1–1.

[43] Google. Quick, Draw! `https://quickdraw.withgoogle.com/`. [Online; accessed 01-October-2017].