To Draw or Not to Draw: Recognizing Stroke-Hover Intent in Non-instrumented Gesture-free Mid-Air Sketching

Umema Bohari
Texas A & M University
umemabohari_123@tamu.edu

Ting-Ju Chen
Texas A & M University
carol0712@tamu.edu

Vinayak
Texas A & M University
vinayak@tamu.edu

ABSTRACT
Drawing curves in mid-air with fingers is a fundamental task with applications to 3D sketching, geometric modeling, hand-writing recognition, and authentication. Mid-air curve input is most commonly accomplished through explicit user input; akin to click-and-drag, the user may use a hand posture (e.g. pinch) or a button-press on an instrumented controller to express the intention to start and stop sketching. In this paper, we present a novel approach to recognize the user's intention to draw or not to draw in a mid-air sketching task without the use of postures or controllers. For every new point recorded in the user's finger trajectory, the idea is to simply classify this point as either hover or stroke. Our work is motivated by a behavioral study that demonstrates the need for such an approach due to the lack of robustness and intuitiveness while using hand postures and instrumented devices. We captured sketch data from users using a haptics device and trained multiple binary classifiers using feature vectors based on the local geometric and motion profile of the trajectory. We present a systematic comparison of these classifiers and discuss the advantages of our approach to spatial curve input applications.

ACM Classification Keywords
H.5.2. User Interfaces: Interaction styles (e.g., commands, menus, forms, direct manipulation); I.3.6. Methodology and Techniques: Interaction techniques; I.5.1. Models: Statistical

Author Keywords
Mid-air interactions; Random forest; Intent recognition; Curve Modeling

MOTIVATION
The ability to draw curves in air using hand and arm movements is fundamental to spatial (mid-air) interactions with applications in 3D sketching [5, 30], geometric modeling [8], hand-writing [26, 34, 1], and spatial authentication [6]. Recently, there has been significant interest in techniques for recognizing symbols within curves drawn in the air [34, 32]. Most, if not all, of these works focus on determining the semantic content in a curve input (*what the user wanted to express through the curve input*). Our goal in this paper is to investigate a complementary yet fundamental problem in mid-air sketching task: determining when users actually intend to draw in mid-air.

IUI'18, March 7–11, 2018, Tokyo, Japan

DOI: https://doi.org/10.1145/3172944.3172985

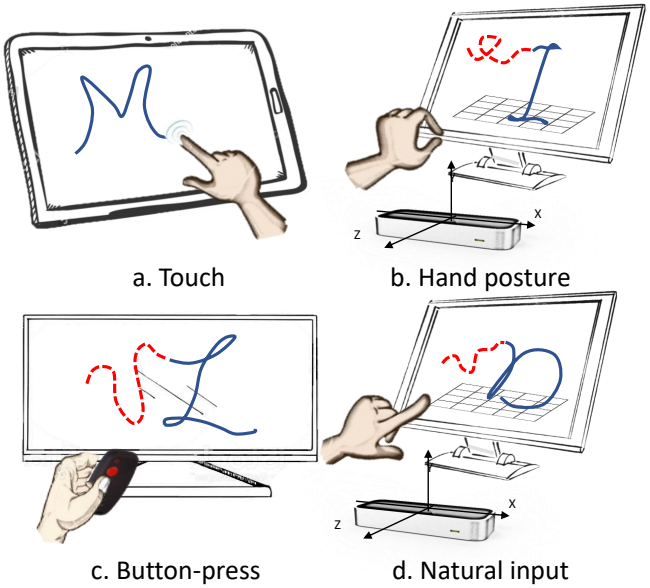


Figure 1. Different mid-air input modalities.

When sketching on a desktop or a tablet, the distinction between a stroke (*what the user actually intends to draw*) and a hover (*all other movements that are not intentional to the drawing task*) is trivially accomplished through explicit events such as mouse button down or touch down (Figure 1(a)). In mid-air interaction, such explicit *interrupts* can either be provided with some specific hand posture (Figure 1(b)) or with a hand-held device with buttons (Figure 1(c)). Most existing approaches embody this requirement in their hardware setup and use events that (such as pen up, pen down, hand posture, etc.) segment the hover points from stroke points for performing further analysis for the stroke segments of the curve. With this in view, the following questions motivate our work:

1. How do users draw in mid-air to express general shapes? What is the effect of spatial input devices or hand postures on the intuitiveness of mid-air sketch input?

2. For processing and recognition of symbols and alphanumeric patterns, a library of pre-defined canonical shapes

is needed. How do we approach recognition of sketches that represent known objects but may not be composed of individually recognizable strokes (such as design sketches)?

3. Does the movement of the hand while mid-air sketching posses inherent geometric and temporal structure that can be used to identify when the user intends to draw in mid-air?

Motivated by these questions, our intention was to study how users move during mid-air sketching, develop qualitative insights regarding such movements, and consequently implement a computational approach to determine when the user intends to draw in mid-air without the use of an explicit mechanism (such as an instrumented controller or a specified hand-posture). In doing so, we make two contributions. First, we present an observational study wherein we characterize hand trajectories generated by users in mid-air sketching tasks with three sketching interfaces. This study helped us gain a better understanding of the quantitative aspects of spatial user input in terms of relative speeds of stroke and hover, intuitiveness of postures and instrumented controllers for sketching, and the types of shapes such as characters, shape primitives, and general multi-stroke shapes. Secondly, we present a novel approach to recognize the user's intention to draw or not to draw in a mid-air sketching task without the use of postures or controllers. Given a sequence of points in the user's finger trajectory, we trained a binary classifier to learn the relationship between the motion profile and geometric features of each point in the trajectory with its true classification (hover or stroke). The resulting model allows for the classification of points on the user's spatial trajectory.

RELATED WORKS

Digital sketching is an extensively studied area of research. In contrast to 3D geometric models, sketches are rough, ambiguous, and vague by nature. As a result there has been significant prior work [23, 24, 11] on segmenting sketches into meaningful objects or components. Curve inputs, in general, have also been used as *gestures* by several approaches in multi-touch devices [7, 2, 3, 37, 33]. However, these techniques are not always scalable to spatial inputs due to the additional uncertainty added by the third dimension: the intended planarity of strokes is not guaranteed.

With the increasing availability of interactive devices such as Wii Remote, Microsoft's Kinect, and Leap Motion, there has been rising interest in techniques for mid-air curve input for gesture recognition [32, 29, 4, 36], hand-writing [26, 34, 1], and sketching [5, 8, 30]. These works are focused on the identification of the curve drawn by the user as some known symbol from an existing list (e.g. alphanumeric symbols) and subsequent analysis of that known symbol. There are also hand-posture based approaches [21, 27, 31, 25] for spatial curve input for detecting user intent for creating mid-air strokes. Despite the significant advances, hand pose estimation and skeleton tracking still lack the required robustness for simple tasks such as sketching. Further, posture recognition is not scalable for interactions with large displays [22, 9, 20], making it difficult to use multiple body skeleton tracking controllers to identify the user intent. Special devices have also been proposed for curve creation, especially for design applications. Grosman et al. [14, 13] propose a physical tape drawing metaphor for automotive curve design, and this approach involves using robust hand trackers for hand skeleton detection. 3D input techniques for large displays as described in [19] make use of infrared trackers and Wiimotes for improved robustness of the interactions.

Our work aims at enabling future interfaces that will be able to eliminate the need for a prescribed set of gestures/postures or controllers to allow users to express 3D artifacts. There are three observations in relation to the previous works that motivate our problem and approach. First, as Taranta et al. [32] note, the particular effectiveness of such recognizers for segmentation and recognition of curve inputs on touch surfaces have not been particularly successful in higher dimensional spaces. Second, in order to expand the scope of intelligent user interfaces beyond symbol recognition to free-form design interfaces, there is a need for methods that do not rely on comparing user input with canonical shapes in a repository. Most techniques are developed and evaluated using segmented data, rather than considering a continuous time-stamped data sequence of 3D points [25, 10]. To address this issue, Krishnan et al. [18] develop and evaluate a sliding window based approach to perform activity recognition with streaming sensor data. The approach adopted in JackKnife [32] proposes implementation of dynamic time warping for continuous dynamic identification of gestures. While JackKnife [32] treats each gesture as a time series representation of direction vectors and then classifies with regard to all templates stored in a database, 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. Finally, unlike sketching on a tablet with a finger or stylus, using a spatial device or a hand posture is not necessarily natural [8]. Works such as Data Miming [15] and grasp-based virtual pottery [35] have demonstrated that for continuous and free-form tasks such as design, users' movements are guided by their interactions with the physical world rather than actions prescribed by the interface designer. We draw from these latter approaches and present a method that encapsulates human movement patterns during mid-air sketching within an intelligent framework for sketch intent recognition.

MID-AIR SKETCHING: OBSERVATIONAL STUDY

To better motivate the need for our approach, we conducted an observational study where our intention was to observe how users specifically react to known spatial input conditions (such as constraining the sketch on a canvas or using a specific gesture) in comparison to a completely rule-free scenario (how one might describe an object through spatial movement without a computer interface at all). For this, we implemented three interfaces using the Leap Motion Controller to record the trajectory of the user's hand skeleton while drawing a given curve. Based on the recorded hand position, the following rules were applied for detecting whether the recorded point is stroke or hover:

I1 *Proximal Plane:* We consider a trajectory point as a stroke point if the palm is in proximity to a pre-defined sketching plane within a threshold (Figure 2 (a)). The interface recognizes the user's intention to sketch when the palm is within a certain threshold of the front facing plane on the screen. Even though this method is agnostic to any specific pose of the hand, we asked users to assume a pointing posture while sketching to maintain a natural way of providing input.

I2 *Pinch Posture:* We consider a trajectory point as a stroke point if the hand assumes the pinching posture (Figure 2 (b)). Our choice of the pinch posture is motivated from the way one holds a pen while sketching on a piece of paper. We recognize a pinch based on the Euclidean distance between the thumb and index finger is within a pre-defined threshold.

I3 *Unrestricted:* In the third interface, we simply asked users to describe a curve in the air without any restrictions on their hand movement or posture. Here, there was no explicit distinction between hover and stroke for the recorded data (i.e. all points were stroke points).

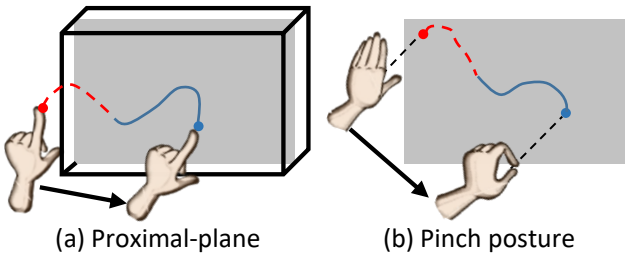


Figure 2. Illustration of the plane-proximity and pinch gesture mechanisms used in Leap Motion interfaces I1 and I2 for the observational study.

Participants

We recruited 10 engineering students (5 female) within the age range of 19-30 years. Except one participant, none of these participants had prior experiences with motion tracking devices such as Wii and Kinect or mid-air sketching.

Procedure

The total time taken during the experiment varied between 30 and 35 minutes and the three interfaces were randomized across the participants. After describing the setup, the purpose of the study, we described the features of the first sketching interface and practically demonstrated its usage. For each participant and task **T3**, we recorded a video of the task, the completion time, and the time-stamped 3D coordinates of the trajectories generated by the users for each sketched shape. Each participant performed the following tasks:

T1 *Unrestricted Sketch*: Participants were asked to draw primitive shapes a square in mid-air without restrictions on their hand posture or movement. The participants' responses were video recorded for observational exploration.

T2 *Posture-preference*: Participants were then asked to repeat the primitive sketching, but with one or more hand postures of their choice from a list of pointing, two-finger pinch, open palm, and pen-holding posture. In addition to video recording, we also recorded the reasons for these choices.

P *Practice*: To familiarize themselves with the interaction of their hand postures with the corresponding three interfaces, the participants were given a brief demonstration of the software and its functions, and were allowed to practice for 5 minutes on each interface. They were allowed to ask questions and were provided guidance when required.

T3 *Sketching with **I1** & **I2***: Participants were asked to sketch on the planar-proximity and pinch based interfaces in a randomized manner. For each of these interfaces, every participant sketched at least two primitives (Figure 3). Although the duration of time for each interface was set to five minutes, we allowed the participants to sketch more shapes with their aspirations. The canvas was cleared after completion of each primitive.

Q *Questionnaire*: Finally, each participant answered a series of questions regarding their perception of each of the interfaces in terms of ease of use, intuitiveness, and robustness. We also collected open-ended comments regarding the tasks.

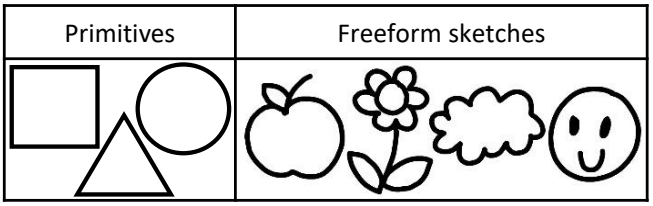


Figure 3. Primitives and free-form multi-stroke sketches drawn by users during the observational study.

Findings

With each participant drawing 6 sketches, a total of 106670 points (55919 *stroke*, 60751 *hover*) were recorded.

Posture Comparison: As expected, nine out of ten participants used index finger to draw single-stroke primitives in the unrestricted sketch interface (**I3**). However, in the posture-preference task (**T2**), three out of these nine participants who used the pointing posture in **I3**, changed their preference to the pinching posture. One user stated: *"It is more like using pen and paper"*. Another user stated: *"Pinching can help me deal with more complex, detailed drawings"*. This indicates that (a) there is a natural *default* hand posture and body movement that manifests commonly across users during mid-air curve input and (b) the way users move while sketching in air is dependent on the nature of the artifact itself that they are trying to draw.

Interface Comparison: We further observed that the order of interfaces affected ease of adopting different hand postures in mid-air sketching. A participant who completed sketching tasks with **I1** and **I2** before **I3**, stated: *"It is difficult to turn it on and off, people normally do not change their hand posture despite it is a stroke or a hover"*. Nine participants out of ten also preferred **I3** over the other two even though I3 did not have any visual cue that distinguished hover from stroke. They

stated: *"The normal interface was easiest"* and *"I do not need to worry about gestures and everything was detected with the normal interface"*.

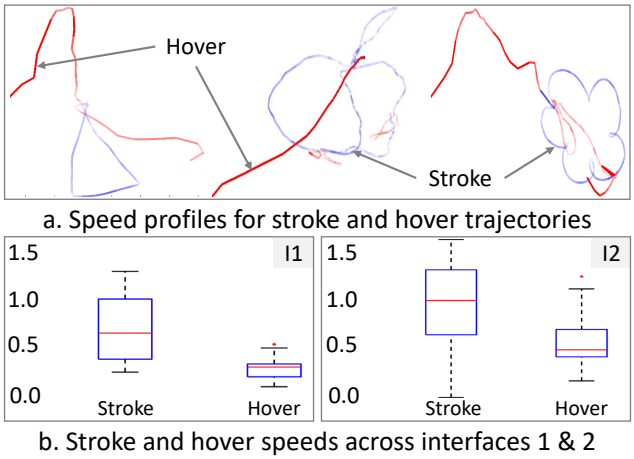


Figure 4. The speed profile (top) shows a near-constant speed for hover that is greater than stroke speeds. The avaerage hover and stroke speeds were greater for the pinch-posture interface (I2) in comparison to proximal plane (I1)

Motion Characteristics: As expected, participants were generally slower (Figure 4(a)) while creating strokes (**I1:** 0.35 m/s, **I2:** 0.56 m/s on average) as opposed to hover (**I1:** 0.66 m/s, **I2:** 0.9 m/s on average). For the proximal plane interface (**I1**), this was observed to a larger extent in comparison to pinch interface (**I2**) (Figure 4(b)). Further, the uncertainty of users' trajectory increased as they reached closer to the instance of transitioning from hover to stroke. This was observed in terms of large straight hover trajectories followed by short zigzag ones while transitioning from hover to stroke.

Shape Type: There were significant differences in how participants approached different types of primitives. They spent time in refining details for multi-stroke shapes. The distribution of hover and stroke regions trajectories for drawing single-stroke primitives were common across users when compared with general multi-stroke shapes. This strongly indicated that for arbitrary sketches, there is a need for a general computational strategy for segmenting and recognizing meaningful parts of the users' trajectory.

STROKE-HOVER RECOGNITION METHODOLOGY

The lack of robust finger tracking was a major concern for our pinch-posture interface (**I2**). As a result, stroke points were intermittently lost due to incorrect prediction of the pinch posture. One of the participants, who had experience in motion tracking devices such as Wii and VR, stated: *"The software interface with hardware probably had some minor issues with detection of fingers...It couldn't follow the movement of my hand and veered off course many times"*. In contrast to our behavioral study, we used the GeoMagic Touch device for data acquisition. Here, users specified their intent to sketch through the use of buttons on the device stylus (similar to a mouse click-and-drag). Our choice of hardware was directly a result of the lack of robustness observed in our behavioral studies with the Leap Motion wherein capturing user intent was prohibitively difficult.

Participants

We recruited 21 engineering students (11 female) within the age range of 19-30 years. None of these participants had prior experience with using the GeoMagic Touch device.

Sketching tasks and assumptions

Each participant was asked to draw 47 different sketches comprising of symbols, 2D and 3D primitives, and free-form sketches in mid-air (Figure 5). The participants were instructed to sketch the curves as naturally as they could (i.e. as fast or as slow as they would if there were no interface). We assume that the 2D data sketched by users is primarily planar, and can be drawn using single strokes or multi-strokes; while the 3-dimensional primitives recorded are multi-planar and multi-stroke sketches.

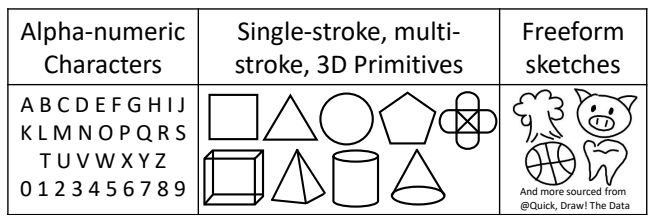


Figure 5. Symbols (left), 2D and 3D primitives (center), and free-form 2D sketches (right) drawn by users during 3D sketching data collection using the GeoMagic Touch device.

Recorded Data

We developed a simple interface that allows the participants to sketch in 3D using the GeoMagic Touch stylus (Figure 6). For every curve drawn, the interface records a continuous sequence of 3D coordinates of the stylus tip trajectory $P_{x,y,z}$, and the classification of that point as being *stroke*(1), or *hover*(0). While writing on a piece of paper, or sketching on tablet surfaces, velocity of the traversal trajectories between two successive strokes is much faster than the stroke trajectories themselves. As noted by Johnson et al.[16], time data is generally used for segmenting strokes in different sketch recognition algorithms. Speed profiles of the data recorded in the observational studies (Figure 4), also point towards a similar distinguishing factor between *stroke* and *hover* curves. Therefore, along with the curve coordinates and classification status, we record the time stamp in milliseconds t of all drawn points.

Features

Based on the motion profiles and local geometric properties of the recorded 3D temporal data, to train and test our classifiers, we construct a feature list comprising of the following:

1. **Motion profile** For every given point i on the trajectory, we calculate its speed (s_i), acceleration (a_i), and jerk (j_i) relative to the $(i-1)^{th}$ point.

$$s_i = \|P_i - P_{i-1}\|, a_i = \frac{s_i - s_{i-1}}{t_i - t_{i-1}}, j_i = \frac{a_i - a_{i-1}}{t_i - t_{i-1}}$$

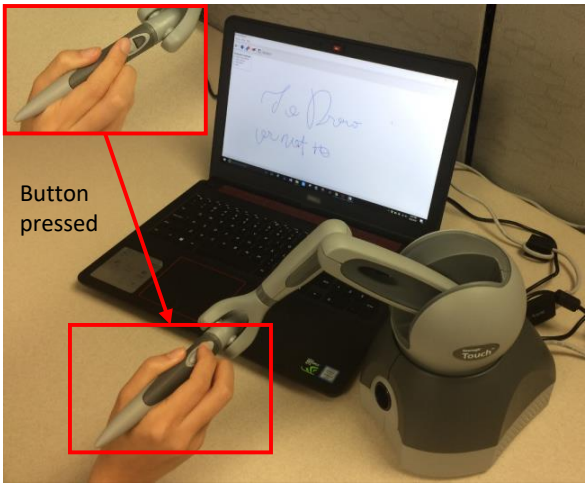
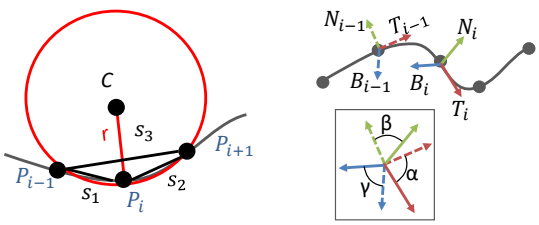


Figure 6. GeoMagic Touch 3D sketching data recording setup. Inset: Stylus button that needs to be pressed by the user to draw an intended *stroke* curve.



a. Curvature Estimate b. Frenet frame rotation

Figure 7. Estimated curvature and discrete Frenet frames of the recorded curve.

Based on the motion characteristics of the 3D data recorded in the observational study (figure 4), we note that at the *stroke-hover* transition, there is an abrupt shift in the local curve speed. To capture this shift, we calculate the relative speed ratio (S_r) at a given point, as:

$$S_r = \frac{s_i}{s_{i-1}}$$

2. **Curvature** Recorded curve trajectories from the Leap Motion observational study, and GeoMagic Touch data collection study are suggestive of the fact that *hover* trajectories have a higher degree of flatness, whereas *stroke* trajectories have a higher curvature. This measure of flatness of the curve at the $(i)^{th}$ point is captured by estimating the Menger curvature (c_i) at that point (Figure 7 (a)). For a given triangle formed by P_i, P_{i-1}, P_{i+1}, the curvature is given by:

$$c_i = \frac{1}{r} = \frac{4A}{s_1 s_2 s_3}$$

where A is the area of $\triangle\, P_{i-1}P_iP_{i+1}$, and s_1, s_2, and s_3 are lengths of the triangle's sides.

3. **Rate of change of Frenet frame** We associate discrete Frenet frames for every edge forming the curve (Figure 7 (b)). The three orthogonal components of each Frenet frame (tangent t_i, normal n_i, binormal b_i) are computed as:

$$t_i = \frac{P_i - P_{i-1}}{\|P_i - P_{i-1}\|}, b_i = t_{i-1} \times t_i, n_i = b_i \times t_i$$

As the user draws the *stroke* and *hover* curves, the change in planarity is estimated by calculating the rate of change of angles between the consecutive Frenet frame components. These angular velocities (ω_α, ω_β, ω_γ) form the last three components of our training and testing feature vector:

$$\omega_\alpha = \frac{\alpha_i - \alpha_{i-1}}{t_i - t_{i-1}}, \omega_\beta = \frac{\beta_i - \beta_{i-1}}{t_i - t_{i-1}}, \omega_\gamma = \frac{\gamma_i - \gamma_{i-1}}{t_i - t_{i-1}}$$

Training and Testing

Of the total data recorded from 21 participants, data from 2 participants had to be discarded, due to misrepresentation of visual cues that hinted towards it being a plane drawing task instead of mid-air sketching task. Of the remaining data, we used 50000 points (22923 *stroke*, 27078 *hover*) for training the model, while the remaining 50000 points (24292 *stroke*, 25708 *hover*) were used to test and cross-validate the data. While training, the feature vectors were randomly sampled from the available pool to eliminate any kind of over-fitting of the trained model due to adjacent data points. For testing, sequences of points belonging to a given shape were sampled. We used MATLAB's implementations of different binary classifiers. With mid-air sketches essentially being a series of alternating *stroke* and *hover* curves, we believe that the status of a given point on the curve is dependent on that of its neighbors. To test this, we train a k-Nearest Neighbor binary classifier. We ensure there is an equal mix of *stroke* and *hover* features in the training list to eliminate model bias towards the over-represented class. Since Support Vector Machines (SVM) find utility in multiple gesture recognition tasks, we also train a binary SVM classifier. Training and testing data is uniformly scaled for accurate implementation of SVM. Finally, we explore random forest classifiers, and discuss different tuning parameters for accurate predictions.

RESULTS

In this section, we discuss the prediction accuracies of the 3 classifiers, and identify the optimum model for *stroke-hover* classification. Next, we perform a parametric optimization for the random forest model, and discuss its prediction results, and limitations.

Classifier Comparison

To compare the classification accuracies (η) across the 3 models, we sample and predict symbols, 2D and 3D primitives, and free-form sketches from the testing feature list.

After performing parametric optimization, with 27 closest neighbors, and *correlation* distance metric, the k-NN model predicts with an average accuracy of $\eta = 57.11\%$ at a 0.04 seconds per point prediction rate. For cross-validated two-class SVM models using both radial basis function, and Gaussian kernels, the average prediction accuracy is $\eta = 51.44\%$ at a 0.02 seconds per point prediction rate. It is observed that while the k-NN model shows some trends of segmenting

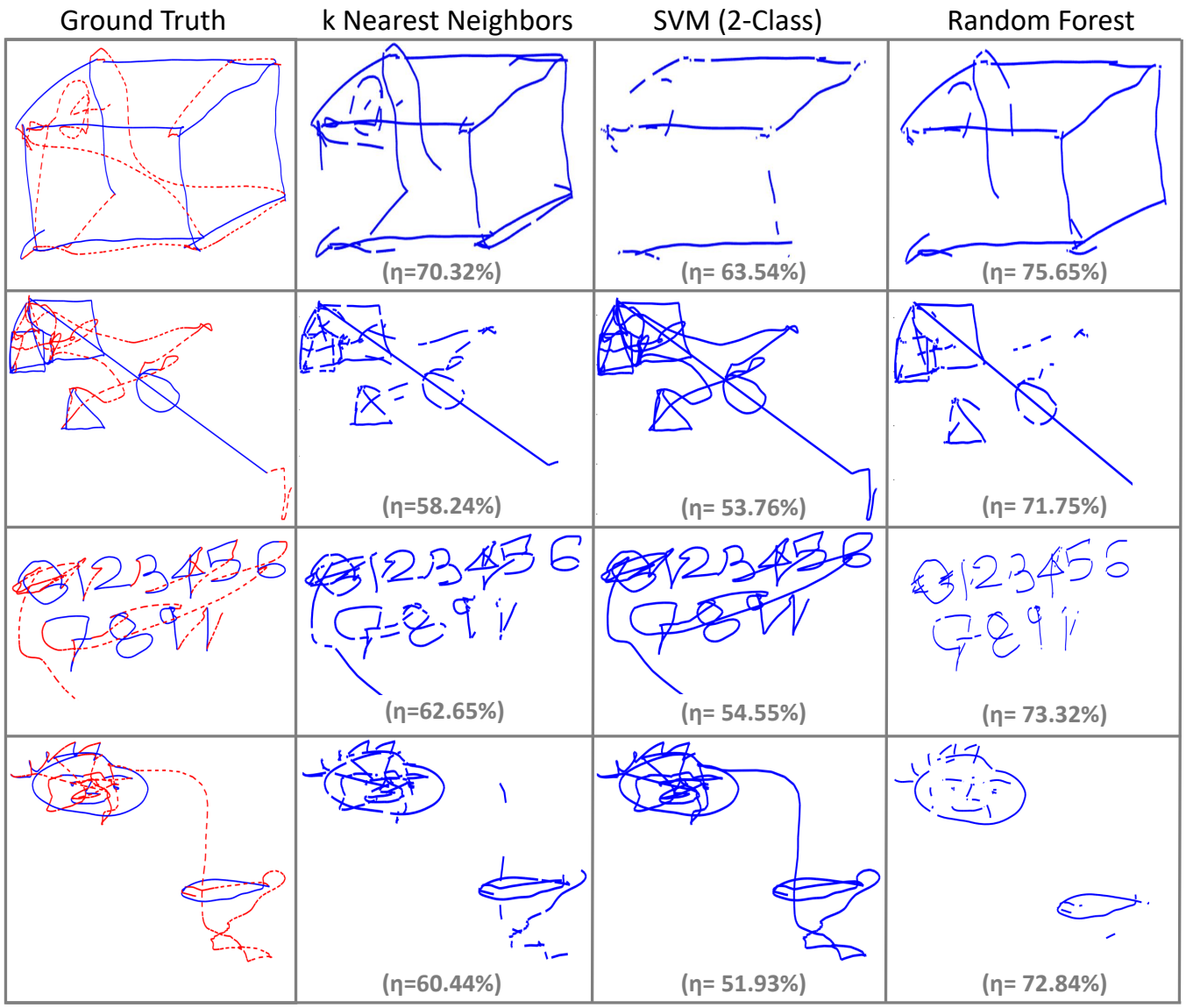


Figure 8. Sampled prediction results with accuracies (η) for *k*-NN, SVM, and Random Forest binary classifiers. *k*-NN and SVM models show high false positives.

the stroke-hover data (Figure 8 (1a),(2a),(3a),(4a)), the SVM model classifies every point as being *stroke* (high false positive) (Figure 8 (2b), (3b),(4b)). The random forest model, on the other hand, exhibits good demarcation between *stroke* and *hover* curves, and classifies them with an average accuracy of $\eta = 71.63\%$ at 0.03 seconds per point prediction rate.

Random Forest

We explore the random forest classifier to further improve its accuracy, by training different models using the following feature list:

$$F_i = \begin{bmatrix} s_i & a_i & j_i & c_i & S_r & \omega_\alpha & \omega_\beta & \omega_\gamma \end{bmatrix} \quad (1)$$

Two main tuning parameters – maximum node splits per decision tree N_s, and number of trees per forest, N_t, are used to control the prediction accuracy. The depth of every decision tree is controlled by N_s. We begin with the default values of $N_s = 7$ and $N_t = 10$, and perform a grid search to identify optimum parameters. Based on this analysis, the model with $N_s = 779$ and $N_t = 40$ performs best with a prediction accuracy of $\eta = 82\%$ for 50000 points (Figure 9).

As in the case of SVM, we explore the effect of scaling 3D data coordinates as well as processed features, on the model prediction accuracy. It is observed that the model with no scaling performs with the best accuracy. Speed (s_i), Speed ratio (S_r), local curvature (c_i), and ω_α are found to be the dominating features, and we retrain the final random forest model using only these parameters.

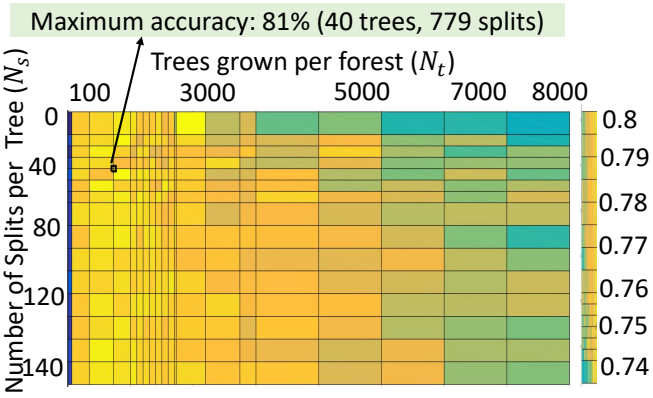


Figure 9. Variation in random forest model accuracy (η) with changes in number of individual tree splits (N_s) and number of trees per forest (N_t).

Instrumented Controller Predictions:
The optimum random forest model is tested using symbols, 2D and 3D primitives, and free-form shapes from the testing feature list. Higher prediction accuracy is observed in case of primitives and free-form shapes $\eta = 85.75\%$ (Figure 11, 10), as compared to symbols $\eta = 78.75\%$ (Figure 10 (a)). This can be explained in a couple of ways. First, during the data collection study, it was observed that due to familiarity with symbols, participants sketched them at a much faster rate. On the other hand, the participants were found to be cautious while sketching multi-stroke free-form sketches and 3D primitives. This had a direct effect on our hypothesis that *hover* trajectories should essentially be traversed faster as compared to *stroke*, in the sense that, this was followed for the multi-stroke shapes, but was not reflected for the recorded symbols data. Second, from a close analysis of the data collection video, it was observed that symbol sketches were primarily restricted to a single plane, while multi-stroke primitives and free-form sketches spanned multiple planes. While the multi-planarity was effectively captured for the shapes using the three Frenet frame vectors, the same cannot be guaranteed for symbols.

Interesting results are observed for free-form 2D sketches and 3D primitives (Figure 11). A majority of these sketches are multi-stroke, involving constant shifts between *hover* and *stroke* curves. The 3D primitives also involve frequent movements between different planes. Despite this fact, the classifier is able to segment such curves with a high accuracy without consulting an existing database to identify the nature of the drawn sketches. In other words, the model is able to classify a given point based on its local properties, rather than classifying it based on the nature of the entire sketch as a whole.

Bare Hand Predictions:
We measured the performance of the model by predicting 3D sketching data recorded in the Unrestricted Interface **I3** using the Leap Motion controller (Figure 12). The predicted results have two characteristics: discontinuity in predictions (Figure 12 (a)), and high false positive rates (hover misclassification) (Figure 12 (b)). These characteristics can be attributed to the different ways in which the training and testing data sets are recorded on the GeoMagoc Touch and Leap motion interfaces, respectively. On one hand, while the GeoMagic Touch based interface recorded data with accurate sketch-hover demarcation, it ultimately is a tethered device, with limited work volume availability. On the other hand, with the Leap interface, users had comparatively more freedom with respect to workspace availability, and were able to draw curves at a much faster rate. However, it is worth noting that the model was able to classify sufficient points without geometrically pre-processing the recorded data using techniques like corner detection.

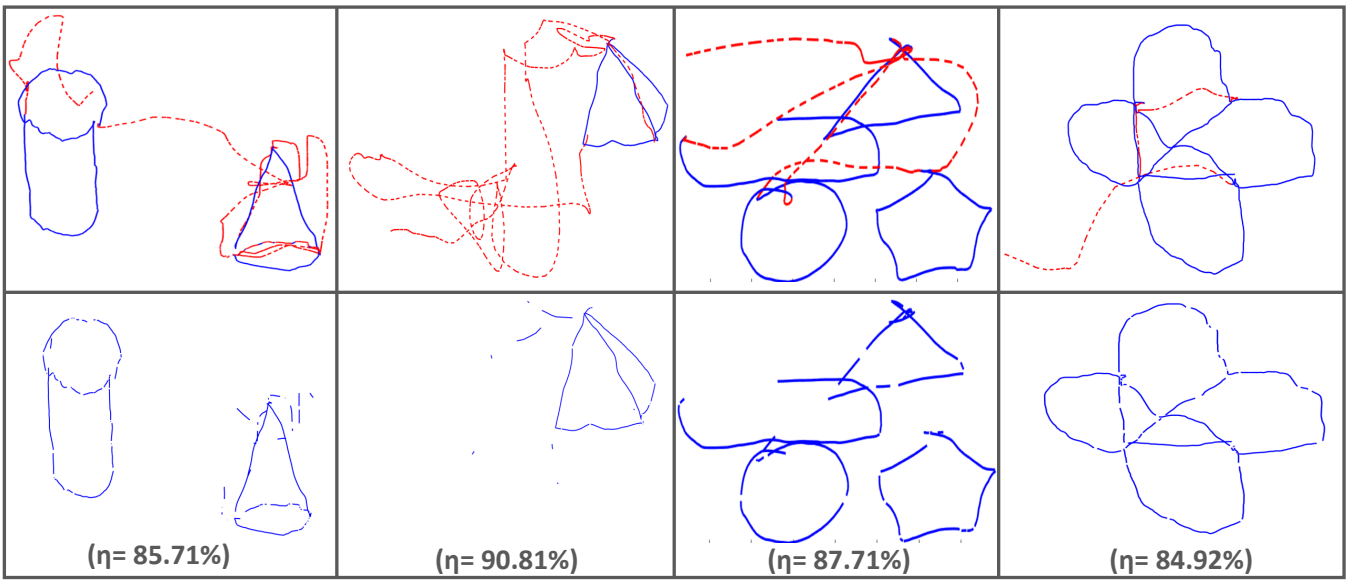
Limitations
The primary reason for using the haptics device for 3D sketching data collection was to ensure accurately recording the *stroke-hover* data of the drawn points. However, the device itself posed certain limitations due to its physical manipulation capacities, offering roughly an interaction space of 16 in x 12 in x 10 in. Also, since it was primarily tethered to the computer, the user did not have much freedom in terms of moving away from the computer and sketching. It was also observed that participants experienced high resistance from the stylus pivot while traversing wavy curves like the tree crown (Figure 11). This resulted in the user dwelling for much longer time while drawing *stroke* curves. Restricted motion about the stylus pivot also resulted in wavy hover trajectories, resulting in high curvature *hover* curves. In future, we plan to eliminate these limitations by conducting a similar study using a non-tethered instrumented controller. Further, mid-air curve input speed is dependent on some other factors like the type of curves, preciseness with which they are drawn, and the application for the 3D curve. In future studies, we plan to record and develop a model on variety of mid-air 3D curves.

DISCUSSION

Design Implications
At its core, our approach offers a different perspective for mid-air sketching, that is, considering intentional strokes as *anomalies* within a continuous hover activity. The ability to detect this anomaly changes the way we design mid-air interfaces. The validation of this model opens up new vistas in the domain of mid-air interactions in context of design ideation, and early stage concept design. With an on-the-fly implementation of our approach, designers will be able to naturally construct 3D concepts through consecutive curve descriptions without the need for specific gestures or instrumented controllers. The classified *stroke* data can then be recognized using standard sketch recognition techniques. In collaborative design ideation setups, coupling this approach with semantic data may lead to the development of intelligent mixed-initiative interfaces.

Application to 3D Modeling
Our approach can be considered complementary to existing segmentation and recognition applications. It is interesting to consider the implications of this model in context of 3-dimensional sketching applications like Tilt Brush [12] and Gravity Sketch [28], that use augmented reality and instrumented controllers for enabling mid-air sketching. Participant feedback from the behavioral study is indicative of the fact

a. Ground truth (top row) and predictions (bottom row) for alpha-numeric characters



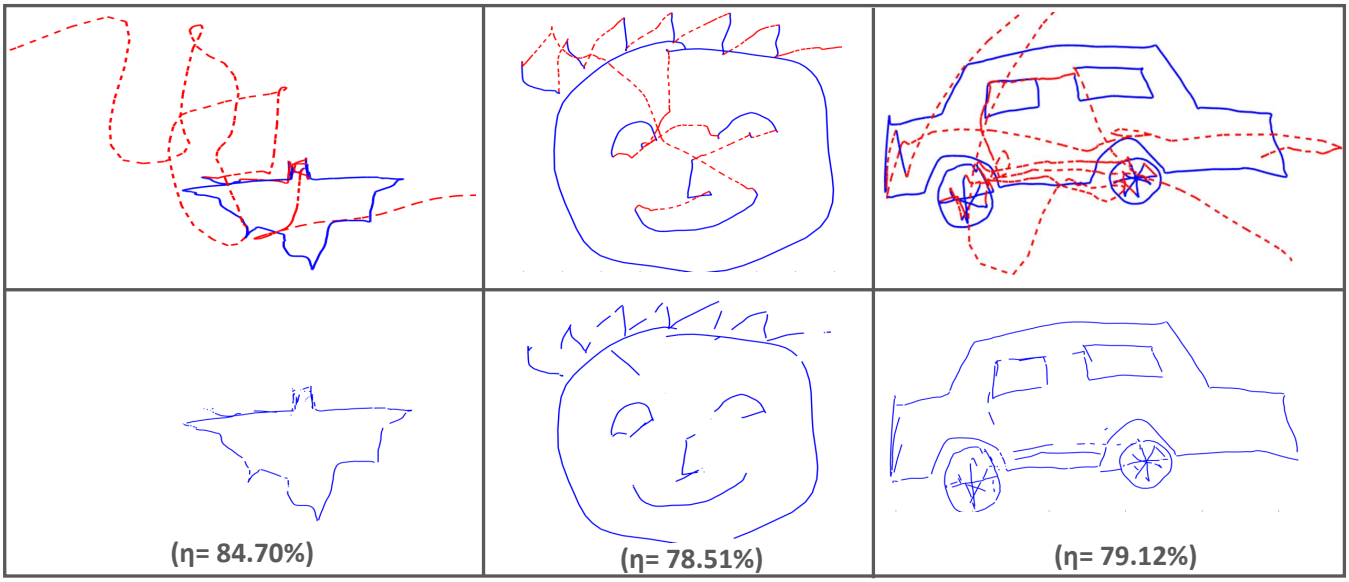
b. Ground truth (top row) and predictions (bottom row) for 3D and 2D single- and multi-stroke shapes

Figure 10. Random Forest model prediction accuracies (η) for symbols (first two rows) and 3D/2D primitives (bottom two rows) sketch data recorded using GeoMagic Touch.

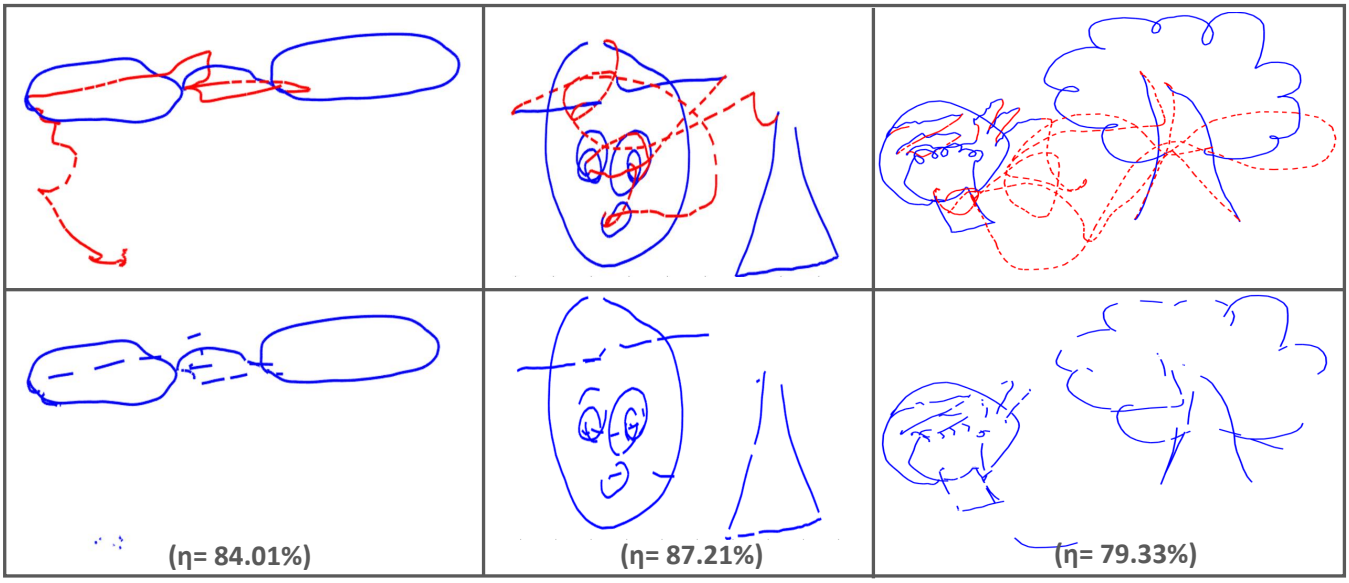
that sketching in mid-air without specifying explicit gestures, or using instrumented controllers, is both natural, as well as intuitive. Our work shows that both aspects are relevant and related. One of the major problems associated with using freehand 3D data in geometric modeling applications is the lack of controllability[17]. Scaling our approach and modeling geometric shape controllability using the *sketch-hover* analogy may lead to the development of mid-air curve input based 3D modeling applications.

Scalability across different interaction volumes

In order for mid-air interactions to scale, we must have different levels of details for recognizing when a user wants to affect the state of an interactive system - specifically in setups involving large displays. In this context, we offer a possibility of bypassing hand skeletal tracking by simply analyzing body-level activity (that can be robustly tracked by most available algorithms) to identify user intent for curve drawing [13]. This will further allow for implementation of novel and richer interfaces for large displays and interaction spaces enabling collaborative design experiences.



a. Ground truth (top row) and predictions (bottom row) for 2-dimensional free-form shapes



b. Ground truth (top row) and predictions (bottom row) for 2-dimensional free-form shapes

Figure 11. Random Forest model prediction accuracies (η) for multi-stroke free-form sketch data recorded using GeoMagic Touch.

CONCLUSIONS & FUTURE DIRECTIONS

In this paper, we presented an approach for capturing user intent for creating strokes using spatial input. Our idea was based on a simple premise that the user's intention to draw in mid-air is inherently present in the motion and geometric characteristics of user-generated hand trajectory. With the current accuracy of our classification algorithms, we believe it is possible to create interesting applications such as interactive art, especially for large displays with multiple users. However, for more precise applications such as 3D design, our approach needs improvements in overall accuracy. We intend to collect a richer and more extensive dataset that can be used for training classifiers based on neural networks (especially recurrent neural networks that can capture the time-series nature of the underlying data). Further, we plan to investigate how our approach could be utilized as an intermediate step toward improving known frameworks for recognition of spatial symbols and gestures. In the broader domain of mid-air curve input, our work opens a new problem context that can potentially lead to novel approaches for enabling users to express visual ideas through spatial interactions.

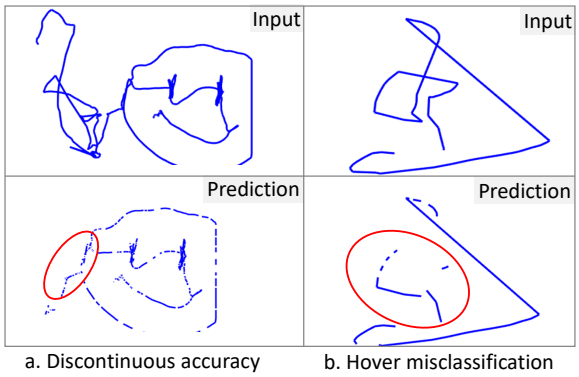


Figure 12. Random Forest model prediction results for 3D sketch data recorded using Leap Motion. Discontinuity in predictions (left), and false positives (right) observed during predictions.

ACKNOWLEDGMENTS

We thank the reviewers for their valuable feedback and comments. This work was supported by the Texas A&M University start-up fund and the Department of Mechanical Engineering at Texas A&M University.

REFERENCES

1. Chelsi Agarwal, Debi Prosad Dogra, Rajkumar Saini, and Partha Pratim Roy. 2015. Segmentation and recognition of text written in 3d using leap motion interface. In *Pattern Recognition (ACPR), 2015 3rd IAPR Asian Conference on*. IEEE, 539–543.

2. Lisa Anthony and Jacob O. Wobbrock. 2010. A Lightweight Multistroke Recognizer for User Interface Prototypes. In *Proceedings of Graphics Interface 2010 (GI '10)*. Canadian Information Processing Society, Toronto, Ont., Canada, Canada, 245–252. http://dl.acm.org/citation.cfm?id=1839214.1839258

3. Lisa Anthony and Jacob O. Wobbrock. 2012. $N-protractor: A Fast and Accurate Multistroke Recognizer. In *Proceedings of Graphics Interface 2012 (GI '12)*. Canadian Information Processing Society, Toronto, Ont., Canada, Canada, 117–120. http://dl.acm.org/citation.cfm?id=2305276.2305296

4. Relja ArandjeloviÄĞ and Tevfik Metin Sezgin. 2011. Sketch recognition by fusion of temporal and image-based features. *Pattern Recognition* 44, 6 (2011), 1225 – 1234. DOI:http://dx.doi.org/https://doi.org/10.1016/j.patcog.2010.11.006

5. Rahul Arora, Rubaiat Habib Kazi, Fraser Anderson, Tovi Grossman, Karan Singh, and George W Fitzmaurice. 2017. Experimental Evaluation of Sketching on Surfaces in VR.. In *CHI*. 5643–5654.

6. Ilhan Aslan, Andreas Uhl, Alexander Meschtscherjakov, and Manfred Tscheligi. 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, New York, NY, USA, 311–318.

7. Seok-Hyung Bae, Ravin Balakrishnan, and Karan Singh. 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, New York, NY, USA, 151–160. DOI: http://dx.doi.org/10.1145/1449715.1449740

8. Yu Chen, Jianzhuang Liu, and Xiaoou Tang. 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, 1–6.

9. Mary Czerwinski, George Robertson, Brian Meyers, Greg Smith, Daniel Robbins, and Desney Tan. 2006. Large Display Research Overview. In *CHI '06 Extended Abstracts on Human Factors in Computing Systems (CHI EA '06)*. ACM, New York, NY, USA, 69–74. DOI: http://dx.doi.org/10.1145/1125451.1125471

10. Fabio Dominio, Mauro Donadeo, and Pietro Zanuttigh. 2014. Combining Multiple Depth-based Descriptors for Hand Gesture Recognition. *Pattern Recogn. Lett.* 50, C (Dec. 2014), 101–111. DOI: http://dx.doi.org/10.1016/j.patrec.2013.10.010

11. Martin Field, Sam Gordon, Eric Peterson, Raquel Robinson, Thomas Stahovich, and Christine Alvarado. 2010. The effect of task on classification accuracy: Using gesture recognition techniques in free-sketch recognition. *Computers & Graphics* 34, 5 (2010), 499–512.

12. Google. 2017. Tilt Brush. https://www.tiltbrush.com/. (2017). [Online; accessed 01-October-2017].

13. Tovi Grossman, Ravin Balakrishnan, Gordon Kurtenbach, George Fitzmaurice, Azam Khan, and Bill Buxton. 2001. Interaction Techniques for 3D Modeling on Large Displays. In *Proceedings of the 2001 Symposium on Interactive 3D Graphics (I3D '01)*. ACM, New York, NY, USA, 17–23. DOI: http://dx.doi.org/10.1145/364338.364341

14. Tovi Grossman, Ravin Balakrishnan, Gordon Kurtenbach, George Fitzmaurice, Azam Khan, and Bill Buxton. 2002. Creating Principal 3D Curves with Digital Tape Drawing. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '02)*. ACM, New York, NY, USA, 121–128. DOI: http://dx.doi.org/10.1145/503376.503398

15. Christian Holz and Andrew Wilson. 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, New York, NY, USA, 811–820. DOI: http://dx.doi.org/10.1145/1978942.1979060

16. Gabe Johnson, Mark D. Gross, Jason Hong, and Ellen Yi-Luen Do. 2009. Computational Support for Sketching in Design: A Review. *Found. Trends Hum.-Comput. Interact.* 2, 1 (Jan. 2009), 1–93. DOI: http://dx.doi.org/10.1561/1100000013

17. D. Keefe, R. Zeleznik, and D. Laidlaw. 2007. Drawing on Air: Input Techniques for Controlled 3D Line Illustration. *IEEE Transactions on Visualization and Computer Graphics* 13, 5 (Sept 2007), 1067–1081. DOI: http://dx.doi.org/10.1109/TVCG.2007.1060

18. Narayanan C. Krishnan and Diane J. Cook. 2014. Activity Recognition on Streaming Sensor Data. *Pervasive Mob. Comput.* 10 (Feb. 2014), 138–154. DOI: http://dx.doi.org/10.1016/j.pmcj.2012.07.003

19. Beverley Laundry, Masood Masoodian, and Bill Rogers. 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, New York, NY, USA, 49–56. DOI: http://dx.doi.org/10.1145/1832838.1832847

20. Lars Lischke, Jürgen Grüninger, Khalil Klouche, Albrecht Schmidt, Philipp Slusallek, and Giulio Jacucci. 2015. Interaction Techniques for Wall-Sized Screens. In *Proceedings of the 2015 International Conference on Interactive Tabletops & Surfaces (ITS '15)*. ACM, New York, NY, USA, 501–504. DOI: http://dx.doi.org/10.1145/2817721.2835071

21. Stan Melax, Leonid Keselman, and Sterling Orsten. 2013. Dynamics Based 3D Skeletal Hand Tracking. In *Proceedings of Graphics Interface 2013 (GI '13)*. Canadian Information Processing Society, Toronto, Ont., Canada, Canada, 63–70. http://dl.acm.org/citation.cfm?id=2532129.2532141

22. Tao Ni, Greg S. Schmidt, Oliver G. Staadt, Mark A. Livingston, Robert Ball, and Richard May. 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, Washington, DC, USA, 223–236. DOI:http://dx.doi.org/10.1109/VR.2006.20

23. G. Noris, D. Sýkora, A. Shamir, S. Coros, B. Whited, M. Simmons, A. Hornung, M. Gross, and R. Sumner. 2012. Smart Scribbles for Sketch Segmentation. *Comput. Graph. Forum* 31, 8 (Dec. 2012), 2516–2527. DOI: http://dx.doi.org/10.1111/j.1467-8659.2012.03224.x

24. Brandon Paulson and Tracy Hammond. 2008. PaleoSketch: Accurate Primitive Sketch Recognition and Beautification. In *Proceedings of the 13th International Conference on Intelligent User Interfaces (IUI '08)*. ACM, New York, NY, USA, 1–10.

25. Z. Ren, J. Meng, and J. Yuan. 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, 1–5. DOI: http://dx.doi.org/10.1109/ICICS.2011.6173545

26. Alexander Schick, Daniel Morlock, Christoph Amma, Tanja Schultz, and Rainer Stiefelhagen. 2012. Vision-based handwriting recognition for unrestricted text input in mid-air. In *Proceedings of the 14th ACM international conference on Multimodal interaction*. ACM, 217–220.

27. Toby Sharp, Cem Keskin, Duncan Robertson, Jonathan Taylor, Jamie Shotton, David Kim, Christoph Rhemann, Ido Leichter, Alon Vinnikov, Yichen Wei, Daniel Freedman, Pushmeet Kohli, Eyal Krupka, Andrew Fitzgibbon, and Shahram Izadi. 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, New York, NY, USA, 3633–3642. DOI: http://dx.doi.org/10.1145/2702123.2702179

28. Gravity Sketch. 2017. Gravity Sketch. https://www.gravitysketch.com/. (2017). [Online; accessed 01-October-2017].

29. Poonam Suryanarayan, Anbumani Subramanian, and Dinesh Mandalapu. 2010. Dynamic Hand Pose Recognition Using Depth Data. In *Proceedings of the 2010 20th International Conference on Pattern Recognition (ICPR '10)*. IEEE Computer Society, Washington, DC, USA, 3105–3108. DOI: http://dx.doi.org/10.1109/ICPR.2010.760

30. Paul Taele. 2014. Intelligent sketching interfaces for richer mid-air drawing interactions. In *CHI'14 Extended Abstracts on Human Factors in Computing Systems*. ACM, New York, NY, USA, 339–342.

31. Andrea Tagliasacchi, Matthias Schröder, Anastasia Tkach, Sofien Bouaziz, Mario Botsch, and Mark Pauly. 2015. Robust articulated-ICP for Real-time Hand Tracking. In *Proceedings of the Eurographics Symposium on Geometry Processing (SGP '15)*. Eurographics Association, Aire-la-Ville, Switzerland, Switzerland, 101–114. DOI:http://dx.doi.org/10.1111/cgf.12700

32. Eugene M. Taranta II, Amirreza Samiei, Mehran Maghoumi, Pooya Khaloo, Corey R. Pittman, and Joseph J. LaViola Jr. 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, New York, NY, USA, 5850–5861. DOI: http://dx.doi.org/10.1145/3025453.3026002

33. Radu-Daniel Vatavu, Lisa Anthony, and Jacob O. Wobbrock. 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, New York, NY, USA, 273–280. DOI: http://dx.doi.org/10.1145/2388676.2388732

34. Sharad Vikram, Lei Li, and Stuart Russell. 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, 1179–1184.

35. Vinayak and Karthik Ramani. 2015. A gesture-free geometric approach for mid-air expression of design intent in 3D virtual pottery. *Computer-Aided Design* 69 (2015), 11 – 24.

36. Don Willems, Ralph Niels, Marcel van Gerven, and Louis Vuurpijl. 2009. Iconic and multi-stroke gesture recognition. *Pattern Recognition* 42, 12 (2009), 3303 – 3312. DOI:http://dx.doi.org/https://doi.org/10.1016/j.patcog.2009.01.030 New Frontiers in Handwriting Recognition.

37. Jacob O. Wobbrock, Andrew D. Wilson, and Yang Li. 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, New York, NY, USA, 159–168. DOI: http://dx.doi.org/10.1145/1294211.1294238