Novel Input of Continuous Recognition of Hand Gesture

Prashant Richhariya, Piyush Chauhan, Anita Soni and Sanjay Srivas

April 2, 2022

Novel Input of Continuous Recognition of Hand Gesture

[1]**Mr. Prashant Richhariya**
Research Associate
UPES, Dehradun, India

[2]**Dr. Piyush Chauhan**
Professor
UPES, Dehradun, India

[3]**Dr. Anita Soni**
Professor
Technocrats, Group
Bhopal, India

[4]**Dr. Sanjay Srivas**
Joint Director
STPI, Pune, India

Abstract

This paper is to illustrate air-writing mechanism of different gesture by using deep learning techniques. Mainly, focus is given to observe and identify air-writing actions which perform unbreakable continuous gesture trajectory. In this paper, we attempt to eliminate unnecessary finger motions unrelated to literature or writing activities as compared to standard pattern recognition approach. For the experimental purpose, Xbox is used for finger tracking without using markers or gloves and also used datasets of writing and nonwriting finger actions.Window-based method for detecting and extracting the air-writing events from incessant watercourse of gesture data, which includes intended state unconnected in writing. A writing segment is created when multiple writing events occur in a row. Writing portion is used as a feed in deep learning network in recognizing different hand gesture trajectory, to create an exclusive method to develop air-writing system which includes both finding and acknowledgement stages. We propose dynamic hand gesture system which achieves an overall accuracy of 97% for word-based recognition and 94% for letter-based recognition using leave-one-out cross validation.

***Keywords*: Air-WR, HMM, Finger writing, Air-writing detection, Gesture trajectory, DHGR**

I. Introduction

Motion with a limb on a touch-based border is straightforward since it uses the same symbol as script with a pen. Current advancements in following knowledge enable tracking of pointer and limbgestures without the use of user-worn sensors, and writing motion is no longer limited on a physical plane. When conventional input devices, such as a keyboard or a mouse, are not available or sufficient, air-writing provides a feasible alternative interface for text entry. Air-writing, as opposed to other non-traditional input approaches such as capturing using a simulated console or comparable systems, allows for "eye-free" execution with minimal attention concentration [1].

Once we inscribe with a fingertip in the midair and use a controller-free following scheme [2], the gesture data includes every piece of the finger undertaking in an unbroken stream, whether we're script or drifting.

The targeted writing action can no longer be found quickly or unambiguously. As a result, detecting and extracting the writing signal from a continuous motion data stream might be difficult. The user may simply write in the air with his or her fingertip because to the Xbox finger-precision tracking [2]. To make the Dive a feasible writing interface, however, an intelligence system must be built that is capable of detecting and recognizing both air-writing and other errant movements. While some exact digit actions can be utilized as in-line delimiter signals to offer endpoint information for a writing activity. Air-writing is hampered when these explicit delimiters are used. In this paper, we propose a system which identifies, segments, and distinguishes the script module from a nonstop motion tracking signal.

In this paper, we propose an algorithm for recognizing air-writing on a motion-based user interface using typical finger movements. There are a lot of stray motions in the air which are unconnected to and often vague from the envisioned literatures or arguments. We re-design recognition kernel to handle hypothetically flawed discovery segments [3,4]. We also analyze the total system performance from writing activity detection to recognition by evaluating air-writing recognition performance based on the observed script routes. Fig. 1 depicts an indication of the limb air-writing knowledge. The continuous motion data is transformed to independently noticed script sections during the detection stage. The identified sections are evaluated at the recognition step for a final outcome, more over a acknowledgement choice or a refusal.

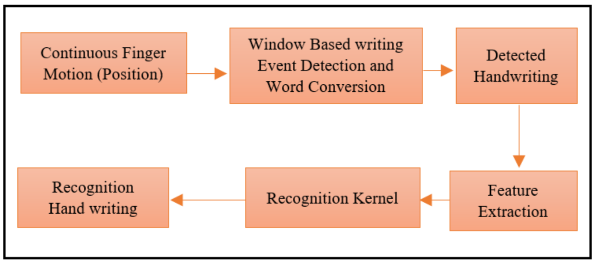


Fig.1: Architecture of air-writing

The following is an outline of the paper. The related work of gesture recognition and detection discussed in Section II. The gesture following scheme, the dive and data video recording techniques for air-writing are described in Section III. The comparative results are discussed in Section IV. Followed by the conclusion in Section V.

II. Related work

Hand tracking needs to be fast and precise in order to be useful in interactive apps. Hand chasing has remained planned using glove-based trailing [5]. However, putting on a glove for a long period of engagement may be inconvenient and uncomfortable. Motion tracking without a controller is thought to provide the greatest natural user experience. Two consumer-grade cameras are utilized in [6] to enable bimanual communication.

With 3D cameras like Kinect, this technology can path the user's pointers to limb precision, and it can be refined to millimeter-level accuracy. In a desktop context, Dive a small USB marginal expedient, is designed to follow fingers (or stick-like things like a pen or a chopstick) precisely [7]. Leap is distinguished from Kinect by its smaller tracking volume and superior resolution. Kinect is built for body and face tracking in a living room-like environment.

Online handwriting identification takes a spatio_temporal approach to the problem, focusing on the writing trajectory rather than the contour [8]. Hidden Markov Models (HMMs) are particularly well-known for their use in spatiotemporal design acknowledgement, such as operational script acknowledgement [9,10]. Running handwriting consists of a series of letters that are linked together without the use of explicit pen-up moves. For online cursive handwriting identification, employed string representations to manage the inter-letter patterns [11]. Current connected script information, such as without PEN is mostly collected via pen-based plans that follow the 2D route using pen-up/pen-down data. Different sorts of tracking devices are required when individuals transcribe in the air, such as a vision-based hand following scheme, inertial linked to a glove or a penetration device [12], [13], [14].

The toner material is straight comprised though script with a pen or a touchpad. In print writing, the pen-awake/pen-miserable activities visibly define the hits, but in running hand writing, they partition the word boundaries. The motion of air-writing is recorded with a continuous stream of sensor data, resulting in uni-stroke writing with no engagement information. Delimitation can be achieved in this scenario using more over obvious subdivision (p-to-w) or unconscious discovery (staining). Buttons can be used to accomplish explicit segmentation, where the operator holds a switch to transcribe and issues it to break [15]. After knobs are not accessible, such as in a controller-free scheme, one choice is to utilize a convinced bearing or sign, such as p-to-w, to indicate the terminations of a script. Additional ways to obvious subdivision are proposed by means of Kinect to make an effort zone for gesticulation limitation [16]. In a user spreads out to write in the air, and penetration evidence is threshed older to part the duplicate.

A method for the off-line recognition of cursive handwriting based on hidden Markov models (HMMs) is described. The features used in the HMMs are based on the arcs of skeleton graphs of the words to be recognized. An algorithm is applied to the skeleton graph of a word that extracts the edges in a particular order.[18] Gestures are expressive, important body motions involving physical movements of the fingers, hands, arms, head, face, or body with the intent of: 1) conveying meaningful information or 2) interacting with the environment [19]. Gestures can bestatic (the user assumes a certain pose or con-figuration) or dynamic (with preparation, stroke, and retraction phases). Some gestures also have both static and dynamic elements, as in sign languages

In assumption, obvious subdivision still put on to methods that need preambles in any form to carryparticipation. Involuntary discovery of script, on the other pointer, does not require planned definition and can make air-writing more suitable with controller_free systems. Amma et al. [20] suggested an air-writing spotting algorithm based on acceleration and angular speed using inertial sensors attached to a glove, with a recall of 99 percent and a low precision of 25% for air-writing spotting. Nonetheless, rather than using automatically discovered segments, the recognition performance was assessed using manually segmented writing.

III. Video recording and air script

A. Imaginary script

There is no tangible flat to inscribe on, and there is no tactile input or imagined script route with air-writing. We employ the container script method described in which the user writes a single-stroke term with each communication covered in its own simulated case [21]. The box-script method sinks the variety of pointer gesture and decreases user exhaustion when likened to outdated left-to-right inscription.

On two totals, the current work fluctuates after First, in its place of using a hand-held tool, the writing is done with a fingertip. The Leap is used to collect motion data in this study, and it delivers motion data with a variety of features derived from air-writing data taken with a hand-held controller [22]. This is referred to as "controller-free air-writing." Second, because there is no button for the user to signify the start and finish of writing, the push-to-write paradigm is no longer appropriate. The trajectory of the motion will invariably comprise both legitimate writing segments and stray bits unrelated to writing, a scenario not seen in [23]. Before effective letter or word recognition can be successfully done, the valid writing segments must be discovered and separated from the errant ones. We exclusively evaluate uppercase letters A to Z in this study, with each letter having a specific stroke order, as in [24]. Lowercase letters, allographs, and varied stroke orders should all be included in the ultimate goal of air-writing recognition. In short, these variances all result in different spatio-temporal patterns, necessitating the modelling of additional recordings. As a result, we begin by simplifying the air-finger writing problem without sacrificing generality.

B. Information capture

First, we generate a 1200-word language using the Google 1T data set, which comprises the greatestmutual 1200 two-, three-, and four- communication terms and four- dispatch prefixes [25]. We prudently choose 100 words from the 1200 words as the shared set, which comprises the 26 letters and 21 string types. The remaining 900 words are knotted and split into 18 groups of 50 words each. The unique terminology delivery allows us to assess how user reliance and out-of-vocabulary words effect recognition with little user data, as indicated in [26].

We created a user interface that uses an index-finger motion for pointing and a pinch gesture for clicking to replace the mouse cursor. This interface refreshes the Leap tracking data at 60 Hz, resulting in no apparent delay in writing and pointing-and-clicking actions. Ordinary control motions (of the index finger) in such a framework are typically: 1) idle or slow swaying motion, 2) reach out to a target for clicking [27,28]. Non-writing motions are not supposed to be random movements which will not appear in this type of UI [29].

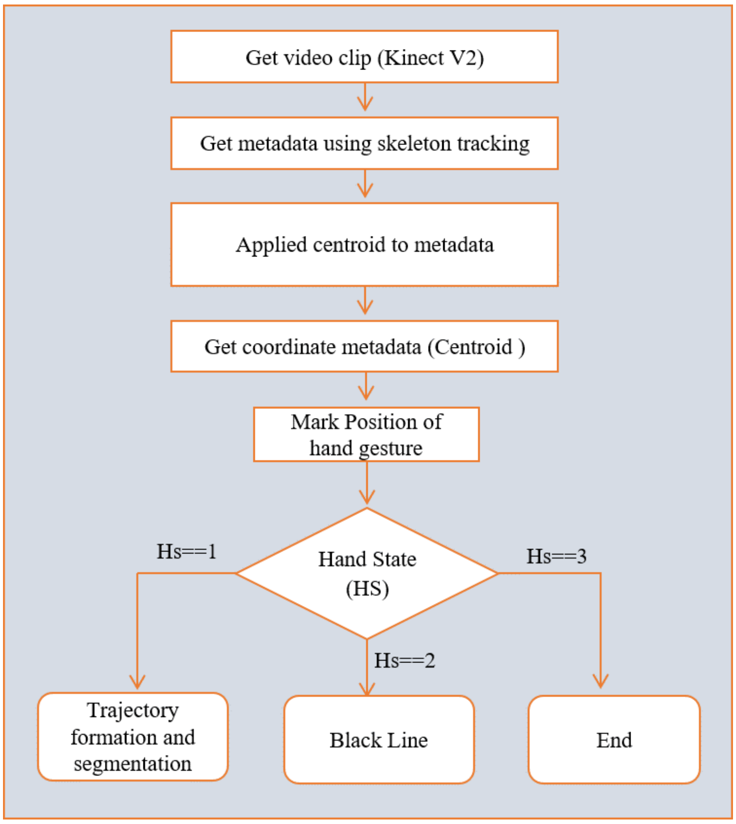


Fig. 2: Trajectory formation

Module 1:

Trajectory formation and segmentation

Get the coordinatesof the centroid from l, l+1 Frame
Calculate the angle between the two points using triangular algorithm
Find the angular changes
Find the Key Frame using angular change
Draw the trajectory between those two frames joining the Centroid coordinates

Module 2:

Black line formation

Get the coordinates of the centroid from l, l+1 frame
Calculate the angle between two points using the triangular algorithm
Find the angular change
Find the Key Frames using the angular change
Draw the Black line by Joining centroid coordinates

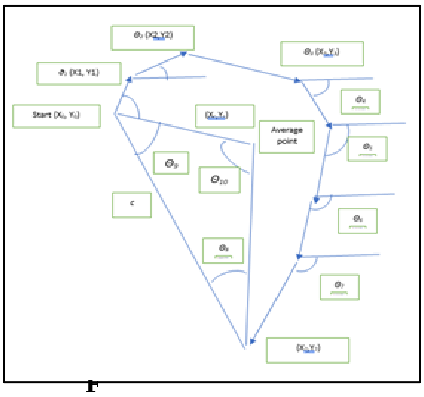


Fig. 4: Movement of Angles

Module 3:

Stop the drawing of the trajectory and proceed to gesture recognition

C. Gesture recognition

Initialization

Pen_on

Initially, we need to create the video input object from the color sensor.

Videoinput (“KinectV2”, “BGR_512*340)

Now we need to use the skeleton data which can be accessed as metadata on the depth stream. We can use “getData” to access it.

[frame, ts, metadata] =getData(vid)

There is much type of metadata fields for our purpose we are going to use: Iskeleton Tracked.

With this we can get the joint location for the person standing in the frame and so on. We can get the coordinates for the centroid.

metadata. IsSkeletonTracked

metadata. jointwordcoordinate (, 12)

We have many fields in the Jointwordcoordinate each having a unique index number. For the centroid of the right hand the no is 12.

Now, we need to create a condition so that the program will be able to know when the trajectory is to be drawn or when the black line/dotted lines are to be drawn or when the tracking is over [30].

[1] Pen_on: pen_on condition is for when the trajectory is to be drawn
[2] Pen_off: pen_off condition is for when the black/ dotted line is to be drawn
[3] Pen_down: pen_down condition is for when the writing is over and we need to go to the next step

These conditions are decided by the hand state of the user

a) If the user hand is closed pen_on condition is applied.
b) If the user hand is in an unknown state i.e., one finger out or making any othershape than hand open or closed then pen_off condition is selected.
c) If the hand is in open state, then the pen_down condition is applied

#Pen_down consists of two major steps:

[1] Trajectory segmentation
[2] Trajectory formation

In this, we are going to use the key frame detection method based on shape-based trajectory segmentation (STS) [31].

In order to detect these key frames, the coordinate vector of the hand’s centroid is defined as

P(t)= {px(t), py(t)}

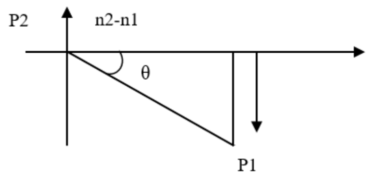
P(t) = metadata to Joint World Coordinate(:,:,12)

Here p () is a coordinate vector

Px, Py are the components
t is the frame no.
T is the total no. of frames

$$0<=t<=T$$

Now we need to find the angle between two adjacent points p1 & p2
Let Ø(t)--→angle between two points at t



From the previous point p1 we find the X distance and y distance of point p2.

Now

$$\mathbf{\tan\theta = y2-y1/x2-x1}$$
$$\mathbf{\theta=\tan^{-1}(y2-y1/x2-x1)}$$
$$\mathbf{\theta(t)= \tan^{-1}(y(t+1)- y(t)/x(t+1)-x(t))}$$

Now, to identify key frames we need to find the angular change θ, k(t, t') between two time indices

It can be found by using

$$\mathbf{K(t, t') =|Ø(t')-Ø(t)|\ (mod\ 2\pi)}$$

Here, we will now detect two types of angular changes abrupt and gradual changes. To detect abrupt changes i.e., the discontinuous shifts in the trajectory are detected as

The angular changes that occur between two sequential angle value k (t-1, t).

If
{
it is greater than a predefined threshold th_A, it is considered as abrupt change.
}
Else
{
if it is greater than threshold th_B is considered as gradual change.
$r(l) = t$, if $k(t-1, t) < th_A$ or $k(t-1, t) < th_G$
}
Where l is the key frame and 1<=l <=L. where L is the total number of key frames.

The value of th_A & th_G are obtained via experiment.This is done to produce a robust result and perform the segmentation of Trajectory. Now, to describe the generation of segmented trajectory based on the detected key frames r(l) using detected key frames as start and end point, when given a set of key frame detection result r(l) a single stream of trajectory v (lc, lp) is generated

$$V(lc, lp) =[p(r(lp)), p(r(lp)+1) \ldots\ldots\ldots\ldots p(r(lc))]$$

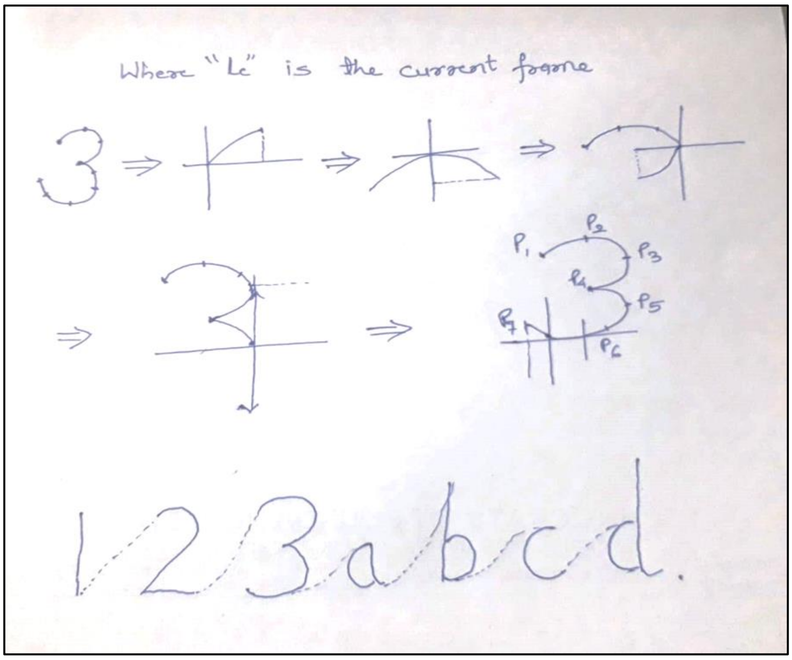


Fig.4: Gesture drawing for finding different trajectory

The function arctan 2, (y x) is a four-quadrant arctangent function, such that ()

$$\text{arctan 2 (y x)} = \begin{cases} \arctan(X/Y) & Y,X >= 0 \\ \pi - \arctan(Y/-X), & Y>0, X<0 \\ \pi + \arctan(-Y/-X) & X, Y<=0 \\ 2\pi - \arctan(-Y/X) & Y<0, X>0 \end{cases}$$

The last three features' angles θ_8, θ_9, θ_{10} are angles of the triangle consisting of the start point (X_0, Y_0), end point (X_7, Y_7) and Average point (X_c, Yc) and are obtained using the cosine theorem,

$$\theta_8 = \arccos (b2+c2-a2)/2bc$$

$$\theta_9 = \arccos (a2+c2-b2)/2ac$$

$$\theta_{10} = \arccos (a2+b2-c2)/2ab$$

where (a, b, c) are edge lengths of the triangle.

IV. Result for drawing trajectory

The most important motion parameter that can discriminate phase from a gesture is the change in speed or acceleration. During the movement, speed of the hand increases to a very high value from almost zero value and then shortly comes down to almost zero. That means, the hand moves with very high acceleration (positive or negative) during the movement phase. On the other hand, during gesturing the speed of the hand gradually increases from a pause, may remain constant for some time and then gradually comes down to almost zero. Therefore, acceleration feature may be a good measure to check for motion. Velocity feature for co-articulation detection: The speed of the hand is generally very high while making a stroke. But, that during the preparation and retraction stages is generally very small. That means, the average velocity of the hand during co-articulation is generally very large.

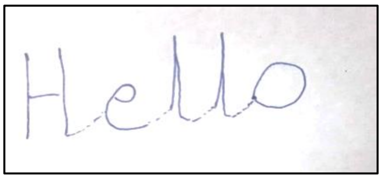


Fig. 5: Multi Stroke Trajectory

In the evaluation, HMM topology is a fully connected with 3 types of states. About the number of state, we take in our consideration the number of segment parts that are contained in possible gesture. The main direction part of all gesture in configuration consists of 8 directions (up, down, left, right, up-right, up-left, down-right and down-left). Intuitionally, we use 8 hidden states with 2 auxiliary states as initialization and output. However, the number of HMM cannot defined precisely by that consideration, it needs to be tested with variation. Therefore, we also evaluate the HMM with 5 and 15 states for comparison with 10 states.

Table 1: Formation HMM Trajectories

Gesture Model	Correct Recognition		
	5 States	10 States	15 States
1	95	85	88
2	99	91	88
3	93	96	87
4	95	97	97
5	28	68	69
6	26	78	60
7	98	75	97
8	78	69	76
9	55	69	73
10	69	98	96
11	78	66	99
12	92	89	92

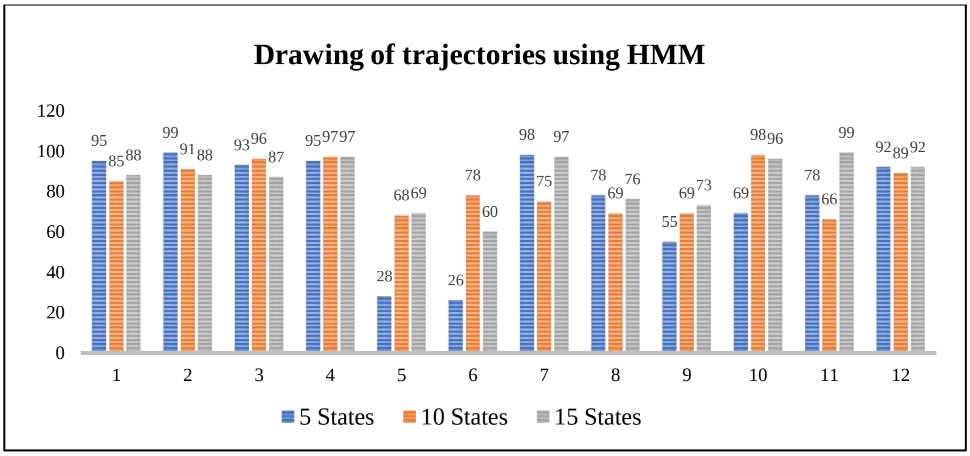


Fig. 5: Trajectories comparison of different States / Stokes

V. Conclusion

We proposed a new gesture recognition system that can recognize the dynamic Indian Sign Language gestures independent of skin color and physical structure of signers. In addition, the dynamic hand gesture recognition of Indian Sign Language using hand motion trajectory Features system works in different places with normal intensity of light. The proposed features, measured from the motion trajectory of the right hand, were able to encode the gesture and to differentiate between 20 different Indian gestures collected from eight different signers, with an average recognition rate of 95.67%. The error rate was mainly due to the high similarity between the gestures.

The proposed system was able to improve the error rate in recognizing the difficult Kaggle dataset from 46% to 28.6%. In future work we will search for other clues to be able to increase the recognition correctness particularly in case of extremely like gesticulations. Also, treatment gesticulations that use both hands or delay with the expression will be lectured in future work.

References

[1] M. Chen, G. AlRegib, and B.-H. Juang, “Air-writing recognition, part 1: Modeling and recognition of characters, words and connecting motions,” IEEE Trans. HumanMach. Syst, in press.

[2] “Leap motion,” http://www.leapmotion.com/, 2015.

[3] R. Plamondon and S. Srihari, “Online and off-line handwriting recogni- tion: a comprehensive survey,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 22, no. 1, 2000, pp. 63 –84

[4] R. Y. Wang and J. Popovic´, “Real-time hand-tracking with acolor glove,” ACM Trans. Graphics, vol. 28, no. 3, 2009.

[5] R. Wang, S. Paris, and J. Popovic´, “6d hands: markerless hand-tracking for computer aided design,” in Proc. of the 24th annual ACM symposium on User interface software and technology, 2011, pp. 549–558.

[6] J. Makhoul, T. Starner, R. Schwartz, and G. Chou, “On-line cursive handwriting recognition using hidden markov models and statistical grammars,” in Proc. of the workshop on Human Language Technology, 1994, pp. 432–436.

[7] J. Hu, M. Brown, and W. Turin, “Hmm based online handwriting recognition,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 18, no. 10, 1996, pp. 1039–1045.

[8] B.-K. Sin and J. H. Kim, “Ligature modeling for online cursive script recognition,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 19, no. 6, 1997, pp. 623–633.

[9] Guyon, L. Schomaker, R. Plamondon, M. Liberman, and S. Janet, “Unipen project of on-line data exchange and recognizer benchmarks,” in Proc. of the 12th intl. conf. on Pattern Recognition, vol. 2, 1994, pp. 29–33 vol.2.

[10] Schick, D. Morlock, C. Amma, T. Schultz, and R. Stiefelhagen, “Vision-based handwriting recognition for unrestricted text input in mid-air,” in Proc. of the 14th ACM intl. conf. on Multimodal interaction, 2012, pp. 217–220.

[11] Amma, D. Gehrig, and T. Schultz, “Airwriting recognition using wearable motion sensors,” in Proc. of the 1st Augmented Human Intl. Conf., 2010, pp. 10:1–10:8.

[12] Amma, M. Georgi, and T. Schultz, “Airwriting: Hands-free mobile text input by spotting and continuous recognition of 3d-space handwrit- ing with inertial sensors,” in Proc. of the 16th International Symposium on Wearable Computers, 2012, pp. 52–59.

[13] X. Zhang, Z. Ye, L. Jin, Z. Feng, and S. Xu, “A new writing experience: Finger writing in the air using a kinect sensor,” MultiMedia, IEEE, vol. 20, no. 4, 2013, pp. 85–93.

[14] M. Chen, G. AlRegib, and B.-H. Juang, “Feature processing and modeling for 6d motion gesture recognition,” IEEE Trans. Multimedia, vol. 15, no. 3, 2013, pp. 561–571.

[15] P. O. Kristensson, T. Nicholson, and A. Quigley, “Continuous recogni- tion of one-handed and two-handed gestures using 3d full-body motion tracking sensors,” in Proc. of the ACM intl. conf. on Intelligent User Interfaces, 2012, pp. 89–92.

[16] T. Brants and A. Franz, “Web 1T 5-gram version 1,” Linguistic Data Consortium, Philadelphia, 2006.

[17] O. Golubitsky and S. M. Watt, “Online stroke modeling for handwriting recognition,” in Proc. of conf. of the center for advanced studies on collaborative research: meeting of minds, 2008, pp. 6:72–6:80.

[18] R.G. Casey and E. Lecolinet. A survey of methods and strategies in character segmentation. In IEEE transactions on pattern analysis and machine intelligence, 1996.

[19] Mitra, S.; Acharya, T., "Gesture Recognition: A Survey," Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on , vol.37, no.3, pp.311,324, May 2007

[20] Bazzi, “Modelling out-of-vocabulary words for robust speech recog- nition,” Ph.D. dissertation, Massachusetts Institute of Technology, 2002.

[21] F. Guo and S. Chen, “Gesture recognition techniques in handwriting recognition application,” in ICFHR, 2010.

[22] M. Bouillon, P. Li, E. Anquetil, and G. Richard, “Using confusion reject to improve (user and) system (cross) learning of gesture commands,” in ICDAR, 2013.

[23] P. Li, N. Renau-Ferrer, E. Anquetil, and E. Jamet, “Semicustomizable gestural commands approach and its evaluation,” in ICFHR, 2012.

[24] X. J. Jin, Q. F. Wang, X. Hou, and C. L. Liu, “Visual gesture character string recognition by classification-based segmentation with stroke deletion,” in ACPR, 2013.S. Liwicki and M. Everingham, “Automatic recognition of fingerspelled words in British sign language,” in CVPR, 2009.

[25] J. Tian, C. Qu, W. Xu, and S. Wang, “Kinwrite: Handwritingbased authentication using kinect,” in NDSS, 2013.

[26] J. P. Wachs, M. Kolsch, H. Stern, and Y. Edan, “Vision-based ¨hand-gesture applications,” in Communications of the ACM, 2011.

[27] H. Bunke, T. Von Siebenthal, T. Yamasaki, and M. Schenkel,“Online handwriting data acquisition using a video camera,” in ICDAR, 1999.

[28] G. A. Fink, M. Wienecke, and G. Sagerer, “Video-based on-line handwriting recognition,” in ICDAR, 2001.

[29] Z. Feng, S. Xu, X. Zhang, L. Jin, Z. Ye, and W. Yang, “Realtime fingertip tracking and detection using Kinect depth sensor for a new writing-in-the air system,” in ICIMCS, 2012.

[30] X. Zhang, Z. Ye, L. Jin, Z. Feng, and S. Xu, “A new writing experience: Finger writing in the air using a kinect sensor,” in IEEE MultiMedia, 2013.

[31] D. Lee and S. Lee, “Vision-based finger action recognition by angle detection and contour analysis,” in ETRI, 2011.