Handwriting and Gestures in the Air, Recognizing on the Fly

Sharad Vikram
Computer Science Division
University of California, Berkeley
sharad.vikram@berkeley.edu
Lei Li
Computer Science Division
University of California, Berkeley
leili@cs.berkeley.edu
Stuart Russell
Computer Science Division
University of California, Berkeley
russell@cs.berkeley.edu

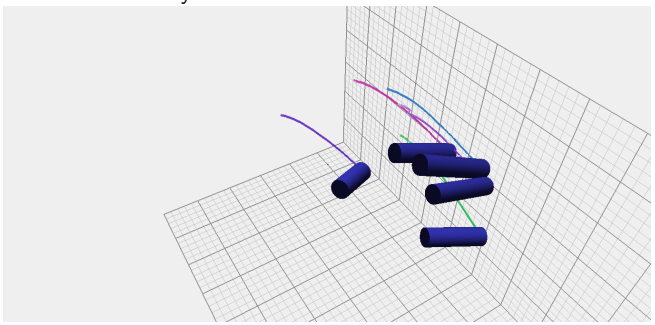


Figure 1: Finger trajectory tracked by Leap Motion controller.

CHI 2013 Extended Abstracts, April 27–May 2, 2013, Paris, France.
ACM 978-1-4503-1952-2/13/04.

Abstract
Recent technologies in vision sensors are capable of capturing 3D finger positions and movements. We propose a novel way to control and interact with computers by moving fingers in the air. The positions of fingers are precisely captured by a computer vision device. By tracking the moving patterns of fingers, we can then recognize users' intended control commands or input information. We demonstrate this human input approach through an example application of handwriting recognition. By treating the input as a time series of 3D positions, we propose a fast algorithm using dynamic time warping to recognize characters in online fashion. We employ various optimization techniques to recognize in real time as one writes. Experiments show promising recognition performance and speed.

Author Keywords
handwriting recognition, hand gesture, time series, dynamic time warping

ACM Classification Keywords
H.5.2 [User Interfaces]: Input devices and strategies.

General Terms
Human factors

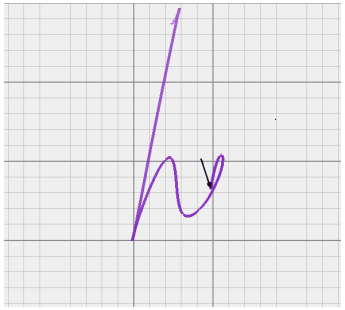


Figure 2: A 2D view of "he" sequence captured by Leap Motion controller.

Introduction

Interaction with computers can go far beyond typing on the keyboard, moving the mouse, and touching the screen. Recent advances in computer vision technology can recognize hand gestures and body shape, as seen in Kinect games. With new computer vision devices such as the Leap Motion controller, precise 3D finger data can be obtained at over 100 frames per second. Therefore, it is possible to track the detailed positions of each finger precisely and efficiently. We advocate a new way to control computers by interpreting finger movements as commands or character input using finger tracking devices. In this paper, we propose a novel method to recognize handwritten characters in the air using such devices.

Traditional character recognition technology is widely applied to such problems as converting scanned books to text and converting images of bank checks into valid payments. These problems can be divided into offline and online recognition.

We introduce a new problem: the online recognition of characters in a stream of 3D points from finger gestures. Many OCR techniques utilize images of completed words, whereas this paper deals with interpreting the data while it is generated, specifically for the scenario of writing "in the air." In this paper we proposes a method of online character recognition, using a data-driven approach. Our method utilizes similarity search technique on multi-dimensional time series. Figure 2 shows a sample sequences of 3D positions transformed from the tracked finger movement data. Our proposed approach to identify characters in these time series exploits the dynamic time warping (DTW) algorithm. A series of recent optimizations make a DTW similarity search feasible in real time. This paper benchmarks the performance of such a similarity search with the given application of handwriting recognition.

The task of identifying characters in a time series requires data to test and train on. Therefore, a new dataset needs to be created, partitioned into multiple candidate time series, specifically the characters in the alphabet, and multiple testing time series, which are words to be recognized. To construct this dataset, the Leap Motion, a commercial computer vision device, is used to record data. The experiment will consist of collecting the same data from over 100 people to account for differences in handwriting.

Our approach aims to deal with a less restricted kind of input than current recognition techniques require. Related work shows that much of modern handwriting recognition relies on a pen up/pen down gesture to The difference between our approach and other current handwriting recognition approaches is the medium in which writing takes place. In many handwriting recognition scenarios, the writing has already taken place and is being statically analyzed. In our approach, we are dealing with live, free form input. Other related work shows the necessity of some pen up/pen down gesture to determine the beginning and end of meaningful input. Our approach attempts gives a less restricted form of input, where no strict gesture is necessary to identify meaningful data.

Related work

Most existing online handwriting recognition techniques depend on a pen up/pen down gesture to window the input data. Essentially, there is a known beginning and end to user input. This paper does not make this assumption. We are using an input device that constantly streams the location of the fingers within its field of view

so the pen up/down gesture is not as easily identified.

One technique used in the process is the segmentation of the data points. This is difficult as it is hard to determine the beginning and end of segments, so typically unsupervised learning and data-driven approaches are used [4]. The statistical approaches to this problem use Hidden Markov Models or use a combination of HMMs and neural networks to recognize characters [5]. Hilbert Warping has been proposed as an alignment method for handwriting recognition [3]. Other scenarios have been proposed, including one where an LED pen is tracked in the air. This allows for 3D data to be interpreted, but also makes sure that the beginning and end of input are clearly defined [1]. Finally, treating the handwriting problem like speech recognition, i.e. treating the input points as a signal, allows in place algorithms with handwriting feature vectors to be used, but the same problem of segmentation arises [8]. These techniques have problems with accuracy in identification.

Another area of application of these techniques is sketch recognition, or digitizing drawings. The methods typically involve searching for sketch primitives and then combining them, which also rely on pen up/pen down gestures [2].

Figure 3: A basic view of the Leap Motion controller.

Capturing finger movements

Data will be recorded with the Leap Motion controller (Figure 3), a commercial technology that captures precise data about the hands. The Leap Motion plugs into computers via USB and sends information about any hands it sees in its field of view, which is a cone of about 8 cubic feet above it. It then determines the location of fingers within the field of view, the angle of the hand, the existence and position of any "tools," such as pens or pencils, and an approximation of the curvature of the palm.

This paper only uses the finger and tool position data taken from the Leap Motion. The Leap can capture these finger points at approximately 100 fps using USB 2.0 port or about 150 using a USB 3.0 port. The importance of this type of input device is that when the user is writing, there is no explicit "beginning" and "end" of input. There is no predefined gesture that indicates when writing characters starts and stops (Figure 4). Instead, the use of this input device requires that the entire stream of data points be searched for instances of letters and only when no matches are found can it be determined that writing has stopped.

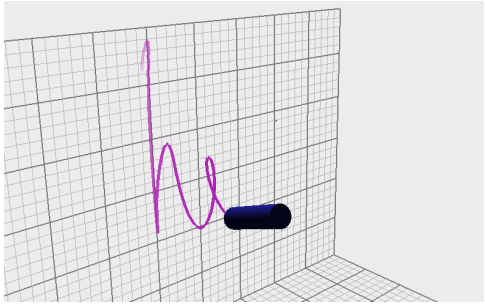
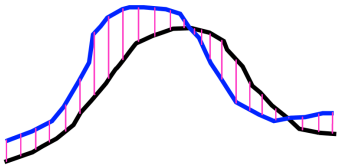


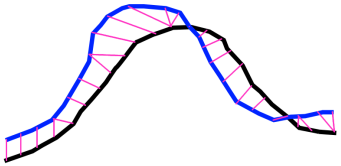
Figure 4: Writing "he" using Leap Motion controller. The captured data are shown in Figure 2.

Building a database

The dataset to be collected consists of two parts. The first is character candidate dataset. The candidates consist of the letters of the alphabet, written in both uppercase and lowercase. Each letter will be replicated five times, for a total of 260 recordings per person. Around 100 people will participate in the instrument for a total of 26000 recordings. The second part is data time



(a) Euclidean distance



(b) Warping distance

Figure 6: Similarity measures. Note DTW matches shape better.

series. The data time series are words to be tested. These words will be taken from a standard corpus, for example, Lincoln's Gettsyburg Address: "Four score and seven years ago our fathers brought forth on this continent, a new nation, conceived in Liberty, and dedicated to the proposition that all men are created equal." Each word will be recorded individually. This will also be replicated by 100 people, for a total of 3000 recordings. Thus, the total size of the dataset will be 29000 recordings. The data will be recorded with the Leap Motion controller, using a browser application.

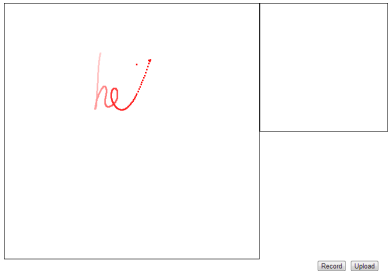


Figure 5: The data recording apparatus.

The browser application shows a 2D preview of the data being recorded and prompts users to confirm the character or word they just wrote (Figure 5). After the user has finished recording, the data will be uploaded to our server through Internet.

Similarity for character trajectories

Our proposed algorithm searches similar character writing sequences from the database using dynamic time warping distance (DTW).

The input is a data time series, D, and a collection of candidate time series, $C = \{c_1, c_2, c_3, ..., c_n\}$. Each time series is multivariate as each element of the time series is a 3D point, $\{x_i, y_i, z_i\}$ The first goal is to have a database similarity search algorithm (Alg. 1).

Algorithm 1: High level database search algorithm.

input : C, D
output: Matches of all the candidates to D
Distances←Map();
for c *in* C **do**
    distance, location ←`Search(c, D)`;
    Distances[c]←(distance,location);
return *Distances*

The similarity search will consist of a sweep across the data time series, checking every subsequence against the candidate and returning the best match. Both candidates and all subsequences are z-normalized in the process.

Algorithm 2: Similarity search algorithm.

input : c, D
output: distance, location
best ← $-\infty$;
for loc *in* D **do**
    distance ←`Similarity(c,D)`;
    if distance $<$ best **then**
        location ← loc;
        best ←distance;
return *distance, location*

The dynamic time warping algorithm is used as a similarity metric between vectors. It is a generalization of the Euclidean distance metric but chooses the closest point within a certain time window, rather than creating a

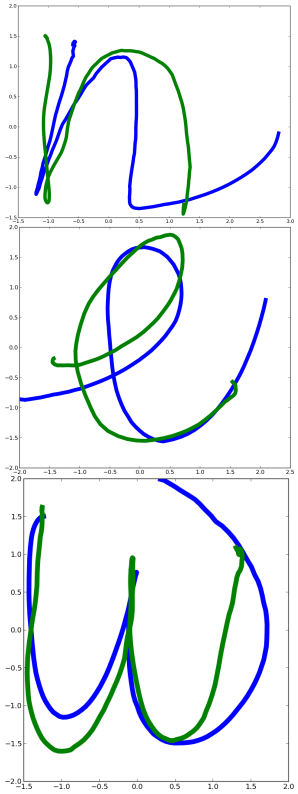


Figure 7: The "n", "e", and "w" matches for the "new" time series. Candidates are in green and subsequences from the data are in blue.

one-to-one mapping of points. When the time window is 0, DTW reduces to Euclidean distance.

Optimization for real time

Given a set of candidate vectors, a nearest neighbor similarity search across an input time series can be run, searching for the closest subsequence match to the each candidate. Each candidate is a recorded 2D sequence of a letter, which we will call a query and the input finger points will be called the data. The complexity of the DTW metric is in $O(nr)$ where n is the length of the vectors being compared (or query size) and r is the time window size. A similarity search of a given query vector across a given data vector of length m would be in $O(nrm)$. With a database of query vectors of size k, the entire search would be in $O(nrmk)$ time. Recent optimizations on DTW similarity search can make this entire operation feasible in real time. The optimizations used by this paper are a improved version of the UCR Suite [6], including:

1. Approximated normalization of query and subsequences, and mean and standard deviations are updated rather than recalculated
2. Cascading the LB Kim and LB Keogh
3. Using LB Keogh on the subsequence in addition to the query
4. Sorting the normalized query to abandon earlier when calculating LB Keogh

A key difference in the proposed method and the UCR Suite is that the UCR Suite was implemented for a univariate time series. Thus, to implement these optimizations, the lower bounding measures had to be extended to a trivariate time series x, y, z. [7] We can also speed up the process by parallelizing the similarity search.

Experiment and results

Preliminary tests on the algorithm have been run. Using database sizes of 30 and 168 recordings, with DTW windows of 0, 1, 5, and 10, similarity searches have been run, getting one-nearest-neighbor matches for each candidate in the database.

Table 1: Running time of plain similarity search with data length 495 (the word "new").

		Database Size	
		30	168
DTW Window	0	1.02s	6.03s
	1	1.55s	6.56s
	5	1.77s	7.51s
	10	2.16s	9.81s

Figure 8 shows an example test time series of the word "new," taken from The Gettysburg Address. In this query, we were looking for the closest database matches to "new" and their locations. The "new" time series is of length 492 and we ran it with a DTW window of 0. It took 1.02 seconds (Table 1). The matches for "n", "e", and "w" are pictured in the left margin (Figure 7).

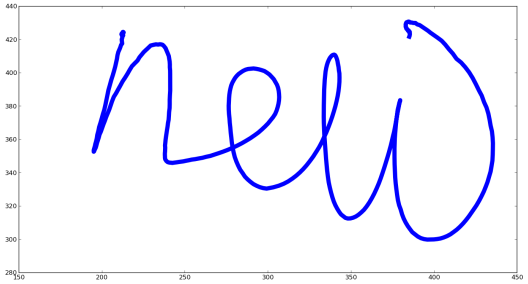


Figure 8: The "new" character time series.

Conclusion and future work

This paper presents a new type of user input for writing: given finger data from the Leap Motion controller, identify characters and words that are written in the air. This problem is novel because no pen up/pen down gesture exists that determines the beginning and end of data. Rather, characters must be recognized in real time. We propose a data-based similarity search algorithm using dynamic time warping and its recent optimizations to do some simple matching. Future work will include extending the recognition algorithm to arbitrary gestures and the use of the Leap Motion controller in different user scenarios than handwriting recognition. These include using a web browser, listening to music, and a replacement to the mouse and keyboard altogether. For example, users can use their computer as normal by moving their finger as the mouse. When a text input area is selected by the mouse, the handwriting input mode would be used, and the stream of finger data points would be interpreted as letters and sent as input to the computer. This process will require the use of 3D data and will thus increase the complexity of the problem. However, such a 3D gesture system would enable us to use more complex types of input, such as American Sign Language and would help disabled people interact with computers much more easily.

Acknowledgment

We would like to thank Leap Motion for the development kit and use of the device. This material is based upon work partly supported by DARPA under Grant No. 20112627. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the funding parties.

References

[1] Asano, T., and Honda, S. Visual interface system by character handwriting gestures in the air. In *IEEE RO-MAN* (2010), 56–61.

[2] Hammond, T., and Paulson, B. Recognizing sketched multistroke primitives. *ACM Trans. Interact. Intell. Syst. 1*, 1 (Oct. 2011), 4:1–4:34.

[3] Ishida, H., Takahashi, T., Ide, I., and Murase, H. A hilbert warping method for handwriting gesture recognition. *Pattern Recognition 43*, 8 (2010), 2799 – 2806.

[4] Plamondon, R., and Srihari, S. Online and off-line handwriting recognition: a comprehensive survey. *Pattern Analysis and Machine Intelligence, IEEE Transactions on 22*, 1 (jan 2000), 63–84.

[5] Plötz, T., and Fink, G. Markov models for offline handwriting recognition: a survey. *International Journal on Document Analysis and Recognition 12*, 4 (2009), 269–298.

[6] Rakthanmanon, T., Campana, B., Mueen, A., Batista, G., Westover, B., Zhu, Q., Zakaria, J., and Keogh, E. Searching and mining trillions of time series subsequences under dynamic time warping. In *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '12, ACM (New York, NY, USA, 2012), 262–270.

[7] Rath, T. M., and Manmatha, R. Lower-bounding of dynamic time warping distances for multivariate time series.

[8] Starner, T., Makhoul, J., Schwartz, R., and Chou, G. On-line cursive handwriting recognition using speech recognition methods. In *Acoustics, Speech, and Signal Processing, 1994. ICASSP-94., 1994 IEEE International Conference on*, vol. v (apr 1994), V/125 –V/128 vol.5.