# Handwriting and Gestures in the Air, Recognizing on the Fly

**Sharad Vikram**
Computer Science Division
UC Berkeley
sharad.vikram@berkeley.edu
**Lei Li**
Computer Science Division
UC Berkeley
leili@cs.berkeley.edu

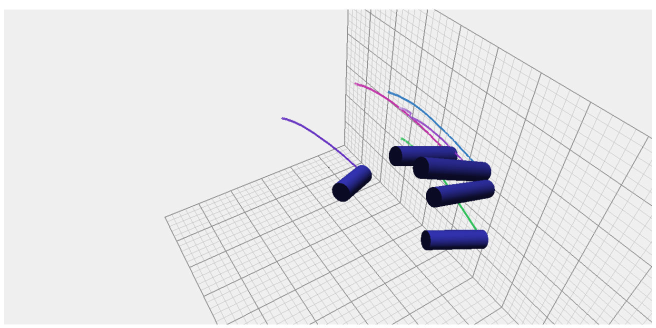


**Figure 1:** The Leap Motion in use


*CHI'13*, April 27 – May 2, 2013, Paris, France.
ACM 978-1-XXXX-XXXX-X/XX/XX.

## 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, time series, dynamic time warping

## ACM Classification Keywords

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

## General Terms

Human factors

## Introduction

Interaction with computers can go far beyond keyboard typing and mouse pointing. Recent advances in computer vision technology can recognize hand gestures and body shape, as seen in Kinect games. With the new computer vision device such as Leap motion detector, it is possible to track the positions of each finger precisely. We propose a new method to control computers by interpret finger movements as commands or character input using finger tracking 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 fingers. 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.”

**Figure 2:** A 2D view of LEAP Motion data

With new innovations in computer vision, precise 3D finger data can be obtained at over 100 frames per second, so this paper also proposes a method of online character recognition, using a data-driven approach and a similarity search. We treat the input data from the computer vision device as a multivariate time series.

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 and store data. The experiment will consist of collecting the same data from 100 people to account for differences in handwriting.
The proposed approach to identify characters in these time series uses 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.

## 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].

## Capturing Finger Movements

**Figure 3:** A basic view of the LEAP Motion

Data will be recorded with the LEAP Motion device, 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. 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.

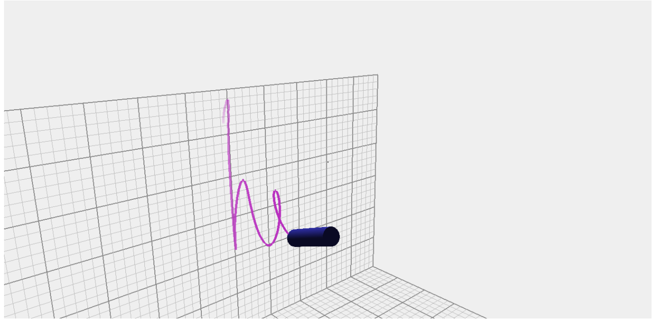


**Figure 4:** The visualizer for the LEAP Motion with a finger writing "he"

## Building Database

The dataset to be collected consists of two parts. The first is candidate time series. 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 series. The data time series are words to be tested. These words will be taken from 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 30000 recordings.

Thus, the total size of the dataset will be 56000 recordings.

The data will be recorded with the LEAP Motion, 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. After the user has finished recording, the data will be uploaded, so at this stage, the user requires an internet connection.

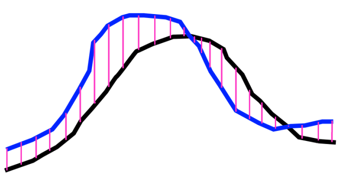
## Similarity for Character Trajectories

The proposed algorithm to identify characters in real time is a dynamic time warping similarity search.

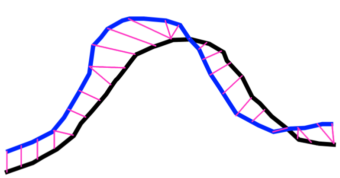
The input is a data time series, $D$, and a collection of candidate time series, $C = \{c_1, c_2, c_3, ..., c_n\}$. Each time series is multivariate as each element of the time series is a 3D point, $\{x_i, y_i, z_i\}$ The first goal is to have some sort of similarity search algorithm.

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

**Algorithm 1**: High level database search algorithm



(a) Euclidean distance



(b) Warping distance

**Figure 6:** Similarity measures

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 the candidate and all subsequences are z-normalized in the process.

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

**Algorithm 2**: Similarity search algorithm

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 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 neighbour similarity search across an input time series can be run, searching for the closest subsequence match to the each candidate. In the scenario described in this paper, each candidate is a recording 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

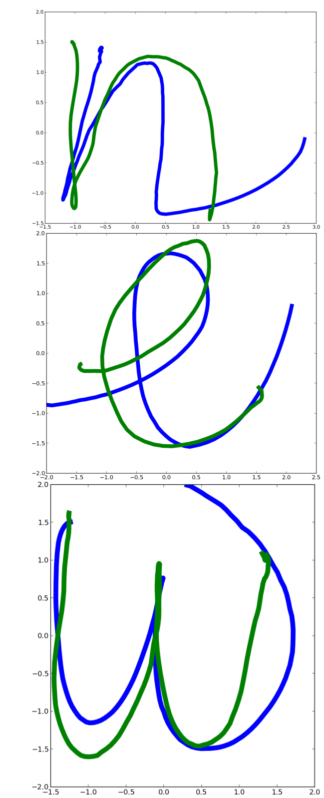


**Figure 7:** The "n", "e", and "w" matches for the "new" time series. Candidates are in green and subsequences from the data are in blue.

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 slightly modified version of the UCR Suite. [6] They are:

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-neighbour matches for each candidate in the database.

**Table 1:** Times of non-parallelized similarity search with data length 785

| | | Database Size | |
|---|---|---|---|
| | | 30 | 168 |
| DTW Window | 0 | 1.46s | 6.03s |
| | 1 | 1.55s | 6.56s |
| | 5 | 1.77s | 7.51s |
| | 10 | 2.16s | 9.81s |

An example test time series was the word "new," taken from The Gettysburg Address. The "new" time series is of length 492 and we ran it with a DTW window of 0. It took 1.02 seconds.

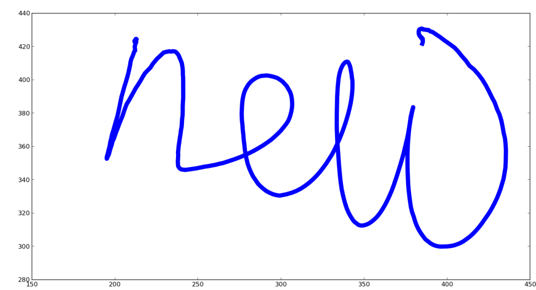


**Figure 8:** The "new" time series data

The matches for "n", "e", and "w" are pictured in the left margin.

## Conclusion and future work

This paper presents a new type of user input for writing: given finger data from the LEAP Motion device, 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 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.

## Acknowledgements

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.