Fréchet Distance based Approach for Searching Online Handwritten Documents

R. Sriraghavendra
sriraghar@csa.iisc.ernet.in

Karthik K.
karthkk@gmail.com

Chiranjib Bhattacharyya
chiru@csa.iisc.ernet.in

Dept. of Computer Science & Automation, Indian Institute of Science, Bangalore-12

Abstract

We propose a novel, language-neutral approach for searching online handwritten text using Fréchet distance. Online handwritten data, which is available as a time series (x,y,t), is treated as representing a parameterized curve in two-dimensions and the problem of searching online handwritten text is posed as a problem of matching two curves in a two-dimensional Euclidean space. Fréchet distance is a natural measure for matching curves. The main contribution of this paper is the formulation of a variant of Fréchet distance that can be used for retrieving words even when only a prefix of the word is given as query. Extensive experiments on UNIPEN dataset[1] consisting of over 16,000 words written by 7 users show that our method outperforms the state-of-the-art DTW method. Experiments were also conducted on a multilingual dataset, generated on a PDA, with encouraging results. Our approach can be used to implement useful, exciting features like auto-completion of handwriting in PDAs.

1. Introduction

In recent times there has been a tremendous increase in the penetration of mobile devices the world over. Although mobile devices and PDAs support touch sensitive screens that are naturally suited for handwritten input, keypads continue to be the primary input interface since handwriting recognition technologies are not reliable enough in many cases. Of late the concept of *digital ink*, i.e. storing raw handwritten data directly without performing any recognition, has been gaining popularity [4]. This method reduces the need for recognition algorithms and enables the use of even languages with complicated scripts (many Asian languages for instance) for which good recognition algorithms are not available.

The usability of handwritten documents will be enhanced greatly if it is possible to search them. We consider the specific case of online handwritten text that is stored as time series, i.e. a sequence of (x,y,t) values denoting the x-y coordinates of the pen at various instances of time t. If the query string to be searched is also input in handwritten format by the same user, then such documents lend themselves to be searched without the need for recognition, as searching for the query string can be treated as searching for a given time series sequence in a collection of time series sequences.

Problem formulation: Given an online handwritten document in digital ink format and a handwritten string q, both written by the same user, we want to retrieve words in the document that have string q as their prefix. Such prefix-based search would enable the user to search for even long words by writing just the first few characters. More importantly, it can be used for dynamic auto-completion of handwritten text. As the user writes a word, the auto-completion feature can retrieve whole words occurring earlier in the document that have the string written so far as their prefix. This feature will greatly enhance the usability and convenience of the handwriting interface.

The rest of the paper is organized as follows. In section 2 we introduce Fréchet distance and other preliminaries. In section 3 we present our main contribution - an algorithm for prefix-based search in handwritten text, based on the discrete Fréchet distance. We present our experimental results in section 4 - we benchmark our method against state-of-the-art Dynamic Time Warping (DTW)[7]. Extensive experiments conducted on the open source UNIPEN dataset demonstrate that our method outperforms DTW. Finally in 5 we discuss some possible future work on our approach.

2. Fréchet Distance and its variants

The *Fréchet distance* [1, 2] (also called *continuous Fréchet distance*) is a well-known measure of similarity between two curves. It can be defined intuitively using the example of a man and his dog walking along two curves - both

---

are allowed to vary their speed but cannot move backward. The *Fréchet distance* between these two curves is the minimal length of the leash required if both the man and the dog are to move from the starting point to the ending point of the corresponding curves. The usefulness of this distance measure stems from the fact that it is not only intuitive but also captures the similarity of curves better than other similarity measures such as the Euclidean distance and the Hausdorff distance [2]. Below we present a formal definition of Fréchet distance and related preliminaries.

2.1. Continuous Frechet distance

The *continuous Fréchet distance* (or just *Fréchet distance*) between two continuous, parameterized curves in d dimensions, $P : [0, m] \mapsto \mathbb{R}^d$ and $Q : [0, n] \mapsto \mathbb{R}^d$ ($m, n > 0$) is defined as [5]:

$$\mathbf{d_F}(P, Q) = \min_{(\alpha, \beta) \in \Gamma_{m,n}} \|P \circ \alpha - Q \circ \beta\|_\infty$$
$$= \min_{(\alpha, \beta) \in \Gamma_{m,n}} \max_{t \in [0,1]} \|P \circ \alpha(t) - Q \circ \beta(t)\|_2 \quad (1)$$

where $\Gamma_{m,n}$ is defined as:

$$\Gamma_{m,n} \equiv \mathcal{M}([0, 1], [0, m]) \times \mathcal{M}([0, 1], [0, n])$$

We use the notation $\mathcal{M}(X, Y)$ to denote the set of all continuous, weakly increasing and surjective mappings from a set X to another set Y. Thus α and β serve to reparameterize the curves P and Q using a common parameter t that takes values in [0,1].

Polygonal curves: A polygonal curve of length $m \in N$ is defined as a mapping $P : [0, m] \mapsto \mathbb{R}^d$, such that for all $i \in [0 : m - 1]$, $P|_{[i,i+1]}$ is affine [5], i.e. $\forall \lambda \in [0, 1]$ we have $P(i+\lambda) = (1-\lambda)P(i)+\lambda P(i+1)$. Alt and Godau [1] give an algorithm for computing the (continuous) Fréchet distance between two polygonal curves. If the two polygonal curves have n and m vertices, their algorithm has a time complexity of $O(nm \log(nm))$. A more general algorithm that works for *piecewise smooth curves* is given by Rote [6]. If the curves obey certain restrictions, this algorithm has a time complexity similar to the algorithm of [1]. Both the algorithms compute the Fréchet distance using complex parametric search techniques on free space diagrams [1, 2].

2.2. Discrete Frechet distance

Since a time complexity of $O(mn \log(mn))$ can be quite high in practice, approximations to continuous Fréchet distance have been developed. The *discrete Fréchet distance* [5] is one such approximation that is applicable to polygonal curves. Formally, the *discrete Fréchet distance* between polygonal curves P and Q of lengths m and n respectively

is defined as

$$\mathbf{D_F}(P, Q) = \min_{(\kappa, \lambda)} \|P \circ \kappa - Q \circ \lambda\|_\infty$$
$$= \min_{(\kappa, \lambda) \in \Psi_{m,n}} \max_{t \in [1:m+n]} \|P \circ \kappa(t) - Q \circ \lambda(t)\|_2, \quad (2)$$

where,

$$\Psi_{m,n} \equiv \mathcal{M}([1 : m + n], [0 : m]) \times \mathcal{M}([1 : m + n], [0 : n])$$
$$[a : b] \equiv \{a, a + 1, ..., b\} \text{ for any two integers } a, b \ni a \le b$$

In terms of the man & dog analogy, the *discrete Fréchet distance* between (polygonal) curves considers only the leash lengths required at time instances where both the man and the dog are at a vertex of their respective polygonal curve paths. On the other hand, the ($continuous$) *Fréchet distance* considers leash lengths required at all instances of time till both the man and the dog have reached the endpoints, i.e. even when either the man or the dog is not at a vertex. An advantage of this discrete variant is that it can be computed using a simple dynamic programming algorithm having a smaller time complexity than the complicated algorithms used to compute continuous Fréchet distance.

Computing the discrete Fréchet distance: For two polygonal curves P and Q of lengths m and n respectively, $\mathbf{D_F}(P, Q)$ is given by $d_{m,n}$, where $d_{m,n}$ is obtained using the following dynamic programming recurrences:

$$d_{0,0} := \|p_0 - q_0\|_2 \quad (3)$$
$$d_{0,j} := \max\{d_{0,j-1}, \|p_0 - q_j\|_2\}, \text{ for } j := 1 \text{ to } n \quad (4)$$
$$d_{i,0} := \max\{d_{i-1,0}, \|p_i - q_0\|_2\}, \text{ for } i := 1 \text{ to } m \quad (5)$$
$$d_{i,j} := \max\{\min\{d_{i,j-1}, d_{i-1,j}, d_{i-1,j-1}\}, \|p_i - q_j\|_2\}$$
$$\text{for } i := 1 \text{ to } m \text{ and } j := 1 \text{ to } n \quad (6)$$

In the above equations, $d_{i,j}$ represents the *discrete Fréchet distance* between the subcurve $P|_{[0:i]}$ of P and the subcurve $Q|_{[0:j]}$ of Q, and hence $d_{m,n}$ represents the *discrete Fréchet distance* between the two entire curves. It is easy to see that this algorithm has a time complexity of $O(mn)$. The space complexity can be limited to $O(m)$ by computing the matrix entries in column order, i.e. first computing $d_{i,0} \forall i := 0, 1, .., m$, then computing $d_{i,1} \forall i := 0, 1, ..., m$ and so on till last column $d_{i,m} \forall i := 0$ to m. In this way, at any point of the algorithm, we only need space for the column being filled and the previous column rather than for the entire matrix.

3. A new approach for searching using the Fréchet distance

When we want to retrieve words that have a given handwritten string as their prefix, the *discrete Fréchet distance* cannot be applied directly since it only reflects the similarity between two entire curves (i.e. fully aligned to each other), whereas the "prefix-based" query string needs to be

checked for similarity with any prefix of a word, including the whole word itself. In order to enable prefix-searching using *discrete Fréchet distance* as the inherent curve similarity measure, we formulate a variant called the *Prefix-Fréchet distance* which captures how well a "query" polygonal curve Q matches with any prefix of a sample polygonal curve P. (In the remainder of this paper, P and Q will denote polygonal curves of lengths m and n respectively)

3.1. Prefix variant of discrete Frechet distance

The *Prefix-Fréchet distance* between "sample curve" P and "query curve" Q, denoted by $\mathbf{prD_F}(P, Q)$, represents the smallest of the *discrete Fréchet* distances between any prefix of P (i.e. any subcurve of P having the same starting point as P) and curve Q, i.e. it is the *discrete Fréchet distance* between Q and the prefix of P that is most similar to Q in terms of *discrete Fréchet distance* (note that Prefix-Fréchet distance is *not symmetric* by definition). Formally,

$$\mathbf{prD_F}(P, Q) = \min_{[0:a] \subseteq [0:m]} \mathbf{D_F}(P|_{[0:a]}, Q)$$

Computing the Prefix-Fréchet distance: Using essentially the same recurrence equations (3)-(6) used to compute *discrete Fréchet distance*, $\mathbf{prD_F}(P, Q)$ too can be computed in $O(mn)$ time (with $O(m)$ space). The difference is that since we are looking for a prefix of P that best resembles Q in terms of *discrete Fréchet distance*, we need to consider all $d_{i,n}$'s, $i = 0, 1, ..., m$ and set $\mathbf{prD_F}(P, Q)$ to the minimum of these. Thus

$$\mathbf{prD_F}(P, Q) = \min_{i=0,1,..,m} d_{i,n}$$

3.2. Applying Prefix-Frechet distance to prefix-based search on handwriting

Searching for words with a given prefix is essentially a task of finding the Prefix-Fréchet distance between the curve corresponding to each word and the curve corresponding to the query, and retrieving the words whose distance is less than some threshold. However, a few additional steps are to be carried out before the Prefix-Fréchet distances are computed:

Scale invariance: Prefix-Fréchet distance is not scale-invariant. Thus if a query matches the prefix of a word in shape but is considerably different in scale the Prefix-Fréchet distance between the two can be very large, which is misleading. To minimize the effect of difference in scale, all sample words as well as the query are rescaled so that they all have the same constant height η. The width of each word (and the query) is scaled by the same ratio as its height. If $W = \langle (x_1, y_1), (x_2, y_2), ..., (x_N, y_N) \rangle$ is an original word

or query, we rescale it to get the corresponding normalized string W' as follows:

$$ymin = \min_{i=1,2,..,N} y_i \ , \ ymax = \max_{i=1,2,..,N} y_i$$
$$xmin = \min_{i=1,2,..,N} x_i \ , \ dy = ymax - ymin$$
$$\rho = \begin{cases} \eta/dy, & \text{if } dy \neq 0 \\ 1, & \text{otherwise} \end{cases}$$
$$x_i' = (x_i - xmin) * \rho, \ y_i' = (y_i - ymin) * \rho$$
$$(\forall \ i = 1, 2, ..., N)$$
$$W' = \langle (x_1', y_1'), (x_2', y_2'), ..., (x_N', y_N') \rangle \qquad (7)$$

Translation invariance: Any curve similarity measure used for handwriting search should reflect only the similarity in shapes of a word and the query. It should not depend on the position of the sample word within a page or line, i.e. it must be translation-invariant. Since Prefix-Fréchet distance as such is not translation-invariant, it is necessary to eliminate the effect of translation of curves before the Prefix-Fréchet distance between them is computed. This is easily achieved by shifting the curve corresponding to a sample word such that its starting point coincides with the starting point of the query curve. Below is our algorithm which eliminates the effect of translation and then computes the Prefix-Fréchet distance between P and Q. The two curves are assumed to have been normalized already for minimizing the effect of scaling.

$\sigma := q_0 - p_0;$ /* for shifting P */
$d_{0,0} := 0;$ /* since starting points are made to coincide */
for $j := 1$ to n
    $d_{0,j} := \max\{d_{0,j-1}, \|p_0 + \sigma - q_j\|_2\};$
    /* $(p_0 + \sigma - q_j)$ simplifies to $(q_0 - q_j)$ */
end
for $i := 1$ to m
    $d_{i,0} := \max\{d_{i-1,0}, \|p_i + \sigma - q_0\|_2\}$
    /* $(p_i + \sigma - q_0)$ simplifies to $(p_i - p_0)$ */
    for $j := 1$ to n
        $d_{i,j} := \max\{\min\{d_{i,j-1}, d_{i-1,j}, d_{i-1,j-1}\},$
                             $\|p_i + \sigma - q_j\|_2\};$
    end
end
$mindist := d_{0,n};$
$endpt := 0;$
for $i := 1$ to m
    if $mindist > d_{i,n}$ then
        $mindist := d_{i,n};$
        $endpt := i;$
    endif
end
return $(mindist, endpt)$
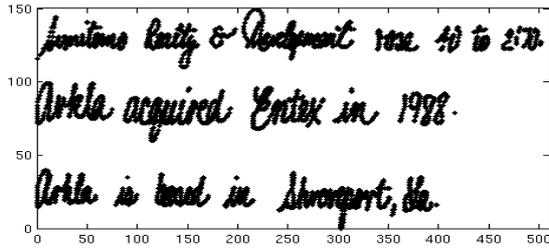
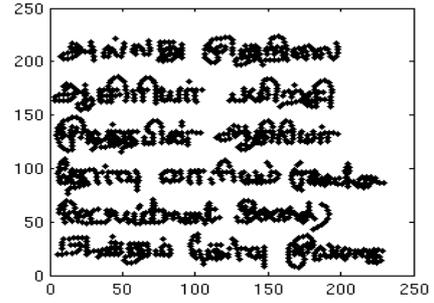**Figure 1. Some sentences written by a user in the Unipen dataset**

4. Experiments

Unipen train_r01_v07 dataset: The first set of experiments used data from category #8 of the Unipen [3] train_r01_v07 dataset, a well-known standard dataset that contains handwritten data collected using various types of handwritten input devices from a vast number of people with widely differing writing styles. Among all benchmarks in train_r01_v07 dataset, only benchmark #8 is a realistic, application-oriented test since the word segmentation problem too should be solved by the recognition or search technique[2]. The data from the 'bbd' group was chosen as it has the most number of sentences - 858 English sentences written by 7 writers, containing a total of almost 16450 words. Also the sentences are realistic as most of them are excerpts from news reports (see figure 1). *We are not aware of any previous related work that was evaluated against category 8 of Unipen dataset.* Our work possibly presents the *first results on this realistic benchmark*, not only for our own approach but also for the most widely used DTW.

Multilingual dataset: Our second dataset[3] consisted of sentences in 3 Asian languages (Hindi, Kannada, and Tamil) as well as a few icons/symbols written by two users. There were over 300 words, over 170 of them being words in Kannada/Hindi words or symbols/icons and the remaining being Tamil words with Arabic numerals interspersed. A sample of the data is plotted in figure 2.

4.1. Preprocessing

All handwritten documents in the UNIPEN benchmark#8 dataset are a collection of sentences rather than words. Since our search is based on prefixes of words, not of sentences, sentences need to be split into individual words before our prefix-matching algorithm can be applied. To segment sentences, we employed an algorithm that identifies indices of time series where the *split distance* d_s is

---

[2] Source: http://ontolingua.nici.kun.nl/scrawls/unipen-pub-guide.html
[3] Link: http://mllab.csa.iisc.ernet.in/~sriraghar/multiling.zip

Figure 2. A sentence (containing Tamil and English words) from the multilingual dataset
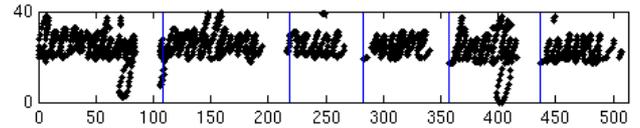

Figure 3. Splitting algorithm applied on a sentence, with $\tau = 10$. Vertical lines indicate the x-axis values of the first point in each word

greater than some threshold τ. The *split distance* at index i of the time series is defined as,

$$d_s = \min_{j>i}(x_j) - \max_{j<=i}(x_j) \qquad (8)$$

where x_j is the x-coordinate corresponding to index j of the time series, i.e. at time instant t_j. An example of the output of our segmentation algorithm is shown in figure 3.

4.2. Results

In this paper we concentrate on writer-dependent search tasks. Thus for each writer the query strings were obtained from the document written by the same writer. Each query string was created by selecting from the (x,y,t)-sequence of a sentence a subsequence corresponding to the prefix of a word in that sentence. Words having less than 3 characters and words occurring less than 4 times in the dataset were not considered for forming queries. On an average around 20 query strings were selected for testing per writer. For each sentence in the dataset, we found the Prefix-Fréchet distance between each word in it and the query word and assigned the minimum of these as the "distance" of the sentence. The sentences were then ranked in the increasing order of their distances and the precision and recall values calculated at each position (rank). As a benchmark we used the DTW distance computed using the same features, the x-y coordinates. Figure 4 shows the graph of precision-vs-recall
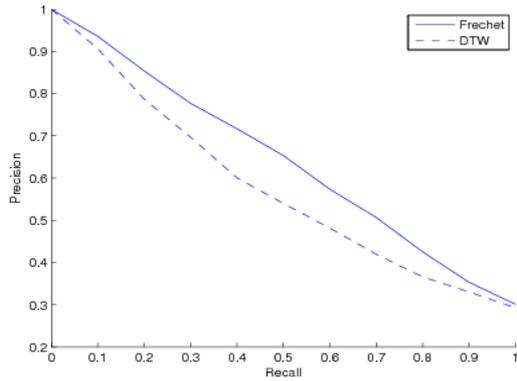
Figure 4. Average precision at various recall rates for DTW and prefix-Fréchet distances

Figure 5. Result of experiment on multilingual dataset: average word-retrieval accuracies

averaged over all words for a user, for DTW and for Prefix-Fréchet distance. It is seen that our Prefix-Fréchet distance based algorithm outperforms the DTW-based algorithm. For the multilingual dataset, we found the word-level retrieval accuracy for searches based on Prefix-Fréchet, DTW, discrete Fréchet and prefix-DTW[4] distances. We used all instances of all words in the dataset as queries and noted the closest match retrieved by the search algorithm from the *rest of the dataset, leaving out the instance* being queried. Since the dataset was created such that it had exactly two instances of any word, we could define retrieval error rate as the probability of that the closest match found is not the other instance of that word, i.e. a wrong match. Thus average error is $e = M/N$ where M is the total number of mistakes and N is the number of words in the dataset. Figure 5 shows the accuracies, i.e. $(1 - e)$. Scheme 1 in the figure corresponds to no normalization (i.e. no rescaling of words) and scheme 2 uses the rescaling approach discussed in section 3.2.

5. Conclusion

We proposed a Fréchet distance based approach for searching online handwritten documents to find words matching a query string handwritten by the user. Our method provides the flexibility of searching for words with a user-specified prefix and is hence suited for implementing novel features like auto-completion of handwriting. Experiments show that our approach works well on multilingual settings too and hence is ideally suited for handheld devices like PDAs. Using only raw features - the (x,y,t) values available directly from input data, our method outperforms the DTW approach that uses the same features. Our

---

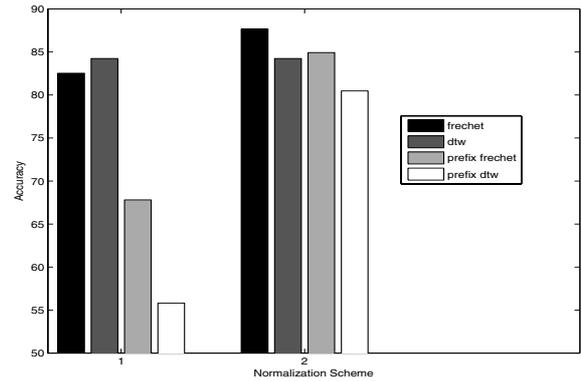[4]We define it analogous to prD_F, but based on DTW. $prDTW(P, Q) = \min_{[0:a] \subseteq [0:m]} DTW(P|_{[0:a]}, Q)$

future work will focus on extending our algorithm to allow search based on query strings occurring anywhere in a word (suffix, prefix or middle substring) and incorporating more sophisticated features (like those used in [4]) that capture more details relevant for searching handwritten text.

Acknowledgement

References

[1] H. Alt and M. Godau. Computing the Fréchet distance between two polygonal curves. *Int. J. Comput. Geometry Appl*, 5:75–91, 1995.

[2] B. Aronov, S. Har-Peled, C. Knauer, Y. Wang, and C. Wenk. Fréchet distance for curves, revisited. In *Algorithms - ESA 2006, 14th Annual European Symposium, September 11-13, 2006, Proceedings*, volume 4168 of *Lecture Notes in Computer Science*, pages 52–63. Springer, 2006.

[3] I. Guyon, L. Schomaker, R. Plamondon, M. Liberman, and S. Janet. UNIPEN project of on-line data exchange and recognizer benchmarks. In *International Conference on Pattern Recognition*, pages B:29–33, 1994.

[4] A. K. Jain and A. M. Namboodiri. Indexing and retrieval of on-line handwritten documents. In *ICDAR*, pages 655–659. IEEE Computer Society, 2003.

[5] A. Mosig and M. Clausen. Approximately matching polygonal curves with respect to the Fréchet distance. *Comput. Geom*, 30(2):113–127, 2005.

[6] G. Rote. Computing the Fréchet distance between piecewise smooth curves. *Comput. Geom. Theory Appl.*, 37(3):162–174, 2007.

[7] H. Sakoe and S. Chiba. A dynamic programming approach to continuous speech recognition. In *Proc. Int. Cong. Acoustics*, Budapest, Hungary, Paper 20C-13, 1971.