Integrating LCS and SVM for 3D Handwriting Recognition on Handheld Devices using Accelerometers

Wang-Hsin Hsu, Yi-Yuan Chiang, Wen-Yen Lin, Wei-Chen Tai, and Jung-Shyr Wu

Abstract—Based on accelerometer, we propose a 3D handwriting recognition system in this paper. The system consists of 4 main parts: (1) data collection: a single tri-axis accelerometer is mounted on a handheld device to collect different handwriting data. A set of key patterns have to be written using the handheld device several times for consequential processing and training. (2) data preprocessing: time series are mapped into eight octant of three-dimensional Euclidean coordinate system. (3) data training: LCS and SVM are combined to perform the classification task. (4) pattern recognition: using the trained SVM model to carry out the prediction task. To evaluate the performance of our handwriting recognition model, we choose the experiment of recognizing a set of English words. The accuracy of classification could be achieved at about 93%.

Keywords—Accelerometer, gesture recognition, handwriting recognition, LCS, SVM.

I. INTRODUCTION

In recent years mobile devices have become popular as a result of the growth of sensor-enabled mobile devices. Users can utilize diverse digital contents anywhere, anytime due to its portability. If the mobile terminal can aware of user’s current context then it could react in some appropriate manner to suit the user without the need of user interaction.

To implement the handwriting recognition system, many different techniques, such as vision-based gesture recognition, touch-based gesture recognition have been utilized. In recent years, a new kind of interaction technology that recognizes users’ movement has emerged due to the rapid development of sensor technology. An accelerometer measures the amount of acceleration of a device in motion. Analysis of acceleration signals enables three kinds of gesture interaction methods: tilt detection, shake detection and gesture recognition [1], [3], [4], [5], [6].

Although in the literature there are already exist some approaches of using acceleration signals for gestures recognition, most work focuses on recognizing the simple gestures such as Arabic numerals [3], [4], [5], simple linear movements and direction [6]. In our work, we attempt to recognize a set of handwritten English words.

We propose a 3D handwriting recognition system in this paper. The system consists of 4 main parts: (1) data collection: a single tri-axis accelerometer is mounted on a handheld device to collect different handwriting data. A set of key patterns have to be written using the handheld device several times for consequential processing and training. (2) data preprocessing: time series are mapped into eight octant of three-dimensional Euclidean coordinate system. (3) data training: LCS and SVM are combined to perform the classification task. (4) pattern recognition: using the trained SVM model to carry out the prediction task. To evaluate the performance of our handwriting recognition model, we choose the experiment of recognizing a set of English words. The accuracy of classification could be achieved at about 93%.

The rest of this paper is organized as follows. The proposed 3D handwriting recognition system is described in Section 2. The effectiveness of this scheme is demonstrated through experimental results in Section 3 followed by Conclusions in Section 4.

II. THE PROPOSED 3D HANDWRITING RECOGNITION SYSTEM

The architecture of the proposed 3D handwriting recognition system is presented as shown in Figure 1 which consists of 4 main parts: data collection, data preprocessing, data training, and pattern recognition. We detail them in what follows.

A. Data Collection

A single tri-axis accelerometer is mounted on a handheld device to collect different handwriting data. A set of key patterns have to be written using the handheld device several times for consequential processing and training. In order to acquire an adequate training result, we collect larger than 10 samples for each key pattern. The output signal of the accelerometer is sampled at 300Hz. Since acceleration signals are sampled in equal-time interval, the length of raw data is variable according to different key pattern and different input speed. Data from the accelerometer has the following attributes: time, acceleration along x-axis, y-axis, and z-axis.
B. Data Preprocessing

We obtain three acceleration time series \(a_x, a_y, a_z\) from the previous step. In order to obtain the position time series, we can use integration calculus twice on the acceleration time series. That is, \(v_x = \int_{t_0}^{t_N} a_x dt\) and \(s_x = \int_{t_0}^{t_N} v_x dt\), where \(v_x\) and \(s_x\) are respectively the velocity and position time series of \(x\)-axis. The other two position time series \(s_y\) and \(s_z\) could be derived using the same method.

While the position time series \(s_x, s_y,\) and \(s_z\) have been derived, we have to transform them into a sequence which composed of a finite set of symbols. Suppose that \(\{s_X(t)\}_{t=t_0}^{t_N},\ X \in \{x, y, z\}\) are given, we could have a difference sequence as below

\[
ds_X(t) = \{ \tau(s_X(t) - s_X(t - 1)) : t = (t_0 + 1), \ldots, t_N \},
\]

\[
X \in \{x, y, z\},
\]

where \(\tau : \mathbb{R} \rightarrow \{0, 1\}\) is defined as

\[
\tau(x) = \begin{cases} 
1, & \text{if } x \geq 0, \\
0, & \text{if } x < 0.
\end{cases}
\]

Then, we can transform \(ds_X(t), X \in \{x, y, z\}\) into a sequence composed of \(\{0, 1, \ldots, 7\}\) as follows:

\[
S(t_i) = ds_x(t_i) \cdot 2^0 + ds_y(t_i) \cdot 2^1 + ds_z(t_i) \cdot 2^2.
\]

The geometric meaning of transformation (3) is to mapping the difference sequence (1) into eight octant of three-dimensional Euclidean coordinate system.

C. Data Training

1) The Longest Common Subsequence (LCS)[2]: Given a sequence \(X = < x_1, x_2, \ldots, x_m >\), another sequence \(Z = < z_1, z_2, \ldots, z_k >\) is a subsequence of \(X\) if there exists a strictly increasing sequence \(< i_1, i_2, \ldots, i_k >\) of indices of \(X\) such that for all \(j = 1, 2, \ldots, k\), we have \(x_{i_j} = z_j\). Given two sequences \(X\) and \(Y\), we say that a sequence \(Z\) is a common subsequence of \(X\) and \(Y\) if \(Z\) is a subsequence of both \(X\) and \(Y\).

In the longest common subsequence problem, we are given two sequences \(X = < x_1, x_2, \ldots, x_m >\) and \(Y = < y_1, y_2, \ldots, y_n >\) and wish to find a maximum length common subsequence of \(X\) and \(Y\). The LCS problem has an optimal structure property as below.

**Proposition 1:** Let \(X = < x_1, x_2, \ldots, x_m >\) and \(Y = < y_1, y_2, \ldots, y_n >\) be sequences, and let \(Z = < z_1, z_2, \ldots, z_k >\) be any LCS of \(X\) and \(Y\).

- If \(x_m = y_n\), then \(z_k = x_m = y_n\) and \(Z_{k-1}\) is an LCS of \(X_{m-1}\) and \(Y_{n-1}\).
- If \(x_m \neq y_n\), then \(z_k \neq x_m\) implies that \(Z\) is an LCS of \(X_{m-1}\) and \(Y\).
- If \(x_m \neq y_n\), then \(z_k \neq y_n\) implies that \(Z\) is an LCS of \(X\) and \(Y_{n-1}\).

The characterization of Proposition 1 states that an LCS of two sequences contains within it an LCS of prefixes of the two sequences. Thus, the LCS problem has an optimal substructure property. A recursive solution also has the overlapping-substructure property, as we will see in below.

Let us define \(c[i, j]\) to be the length of an LCS of the sequences \(X_i\) and \(Y_j\). If either \(i = 0\) or \(j = 0\), one of the sequences has length 0, so the LCS has length 0. The optimal substructure of the LCS problem gives the recursive formula

\[
c[i, j] = 0 \\
c[i - 1, j - 1] + 1 \\
\max(c[i, j - 1], c[i - 1, j])
\]

if \(i, j > 0\) and \(x_i = y_j\),

if \(i, j > 0\) and \(x_i \neq y_j\),

(4)

Based on equation (4), we have an recursive algorithm to compute the length of an LCS of two sequences.

2) The Support Vector Machines (SVM): Suppose we are given a set of training data:

\[
\{(x_1, y_1), (x_2, y_2), \ldots, (x_i, y_i)\} \subset \mathcal{X} \times \{\pm 1\},
\]

where \(\mathcal{X}\) denotes the space of the input patterns. In \(\varepsilon\) support vector classification [7], the goal is to find a function \(f(x)\) that has at most \(\varepsilon\) deviation from the actually obtained targets \(y_i\) for all training data, and at the same time is as flat as possible. That is, 
we want to find a linear functions \(f\) with maximal margin, taking the form

\[
f(x) = \langle w, x \rangle + b \quad \text{with} \quad w \in \mathcal{X}, b \in \mathbb{R}
\]

where \(\langle \cdot, \cdot \rangle\) denotes the inner product in the input space \(\mathcal{X}\). The task can be written as a convex optimization problem:

\[
\min \frac{1}{2} ||w||^2 + C \sum_{i=1}^{l} (\xi_i + \xi_i^*)
\]

subject to \(y_i - \langle w, x_i \rangle - b \leq \varepsilon + \xi_i\)

\(\langle w, x_i \rangle + b - y_i \leq \varepsilon + \xi_i^* \quad \forall i\)

\(\xi_i, \xi_i^* \geq 0\)

The key idea to solve (7) is to construct a Lagrangian function from the objective function and the corresponding constraints. The vector \(w\) has the form

\[
w = \sum_{i=1}^{l} (\alpha_i - \alpha_i^*) x_i
\]

(8)
TABLE I
THE AVERAGE LENGTH OF THE LONGEST COMMON SUBSEQUENCE BETWEEN TWO WORDS IN THE KEY WORD SET.

<table>
<thead>
<tr>
<th>Average length of LCS</th>
<th>Kimble</th>
<th>Apple</th>
<th>Nathan</th>
<th>Model</th>
<th>Apostal</th>
</tr>
</thead>
<tbody>
<tr>
<td>Kimble</td>
<td>273</td>
<td>135</td>
<td>171</td>
<td>124</td>
<td>207</td>
</tr>
<tr>
<td>Apple</td>
<td>—</td>
<td>250</td>
<td>110</td>
<td>172</td>
<td>177</td>
</tr>
<tr>
<td>Nathan</td>
<td>—</td>
<td>—</td>
<td>261</td>
<td>181</td>
<td>199</td>
</tr>
<tr>
<td>Model</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>247</td>
<td>148</td>
</tr>
<tr>
<td>Apostal</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>281</td>
</tr>
</tbody>
</table>

and \( f \) can be written as

\[
 f(x) = \sum_{i=1}^{l} (\alpha_i - \alpha^*_i) \langle x_i, x \rangle + b. \tag{9}
\]

For the cases of nonlinear classification, the training patterns \( x_i \) are preprocessed by a mapping \( \Phi : \mathcal{X} \rightarrow \mathcal{F} \) into some feature space \( \mathcal{F} \) and applying the standard SV classification algorithm. The mapping \( \Phi \) need not to be known since it is implicitly defined by the choice of kernel functions \( K \):

\[
 K(x_i, x_j) = \langle \Phi(x_i), \Phi(x_j) \rangle. \tag{10}
\]

The decision function \( f \) in (9) becomes

\[
 f(x) = \sum_{i=1}^{l} (\alpha_i - \alpha^*_i) K(x_i, x) + b. \tag{11}
\]

The input points \( x_i \) with \( (\alpha_i - \alpha^*_i) \neq 0 \) are called support vectors.

D. Pattern Recognition

After the step of data training, we have an SVM model with the form as equation (11). For a new input pattern, \( a_{0x}, a_{0y}, a_{0z} \), we have to process them using the data preprocessing method and we could get a sequence \( S_0(t) \). Then, the LCS algorithm could be applied. We compute the length \( l \) of the longest common subsequence between \( S_0(t) \) and a particular key word, \( K \). Using the length \( l \) as the input of SVM model (11), the model would tell us whether the new pattern is the key word \( K \) or not.

III. EXPERIMENTAL RESULTS

To evaluate the performance of our handwriting recognition model, we choose the experiment of recognizing a set of English words. The set of English words contains \{Kimble, Apple, Nathan, Model, Apostal\}. For each word, we collect at least 10 patterns from the handheld device (HTC G1 mobile phone). Table 1 is a statistic of the average length of the LCS between these words. It is easy to see that the patterns indicating the same word have larger length than the patterns indicating the different words. Then, using SVM, the accuracy of classification could be achieved at about 93%.

IV. CONCLUSIONS

In this paper, we propose a handwriting recognition system based on a single tri-axis accelerometer mounted on a cell phone for human computer interaction. The system is consists of 4 main parts: (1) data collection: a single tri-axis accelerometer is mounted on a handheld device to collect different handwriting data. A set of key patterns have to be written using the handheld device several times for consequential processing and training. (2) data preprocessing: time series are mapped into eight octant of three-dimensional Euclidean coordinate system. (3) data training: LCS and SVM are combined to perform the classification task. (4) pattern recognition: using the trained SVM model to carry out the prediction task. The experimental results show that the accuracy of classification could be achieved at about 93%.

REFERENCES