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

WANG-HSIN HSU ${ }^{1,2}$, YI-YUAN CHIANG ${ }^{1 *}$, and JUNG-SHYR WU ${ }^{2}$<br>${ }^{1}$ Department of Computer Science and Information Engineering<br>Vanung University<br>No. 1, Vanung Rd., Chungli, Taoyuan, 32061<br>TAIWAN<br>${ }^{2}$ Department of Electrical Engineering<br>National Central University<br>Chungli, Taoyuan, 32054<br>TAIWAN<br>kimble@vnu.edu.tw, yychiang@vnu.edu.tw, jswu@wireless.ee.ncu.edu.tw


#### Abstract

Based on accelerometer, we propose a 3D handwriting recognition system in this paper. 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: weighted 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 $96.85 \%$.


Key-Words:- Accelerometer, gesture recognition, handwriting recognition, LCS, SVM.

## 1 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 glove-based devices (e.g., CyberGlove), or vision-based gesture recognition [3], [11] 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 [2], [6], [7], [8], [10].

Although in the literature there are already exist some approaches of using acceleration signals for gestures recognition, most work focuses on recognizing

[^0]the simple gestures such as Arabic numerals [6], [7], [8], simple linear movements and direction [10]. 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 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: weighted longest common subsequence (LCS) and support vector machine (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 $96.85 \%$.

The rest of this paper is organized as follows. Theoretical backgrounds, including the weighted LCS and SVM are described in Section 2. In Section 3, we
discuss the feasibility of applying weighted LCS in 3D handwriting recognition. The proposed recognition system is presented in Section 4. Then, the effectiveness of this scheme is demonstrated through experimental analysis in Section 5 followed by Conclusions in Section 6.

## 2 Theoretical Backgrounds

### 2.1 The Weighted LCS Algorithm

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 [4].

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 $L C S$ 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 $L C S$ 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

$$
c[i, j]=\left\{\begin{array}{c}
0  \tag{1}\\
\text { if } i=0 \text { or } j=0 \\
c[i-1, j-1]+1 \\
\text { if } i, j>0 \text { and } x_{i}=y_{j} \\
\max (c[i, j-1], c[i-1, j]) \\
\text { if } i, j>0 \text { and } x_{i} \neq y_{j}
\end{array}\right.
$$

Based on equation (1), we have an recursive algorithm to compute the length of an LCS of two sequences.

```
LCS-Length (X, Y)
    \(m=\) length [X]
    \(\mathrm{n}=\) length \([\mathrm{Y}]\)
    for \(i=1\) to m
        c \([i, 0]=0\)
    for \(j=0\) to \(n\)
        c \([0, j]=0\)
    for \(i=1\) to \(m\)
        for \(j=1\) to \(n\)
                if \(x(i)=y(j)\)
                                    \(c[i, j]=c[i-1, j-1]+1\)
                                    b[i,j]='up-left'
                elseif c[i-1,j]>=c[i,j-1]
                        c[i,j]=c[i-1,j]
                                    b[i,j]='up'
                else
                    \(c[i, j]=c[i, j-1]\)
                                    b[i,j]='left'
    return(c)
Print-LCS (b, X,i,j)
            if \(i==0\) or \(j==0\)
            return
        if b[i,j]=='up-left'
            Print-LCS (b, X,i-1,j-1)
                        print \(x(i)\)
        elseif b[i,j]=='up'
            Print-LCS (b, X,i-1, j)
        else
            Print-LCS (b, X,i,j-1)
```

Another structure, useful in molecular biology, is the weighted sequence [1], [13]. This is defined as a sequence $S=s_{1}, s_{2}, \ldots, s_{l}$ where a value is associated to each $s_{i}$ for $i=1, \ldots, l$. While comparing two weighted sequences we define a weighted function, $W$, assigning a value to each possible match between two characters one from the first sequence and the other from the second sequence. The LCS variant for these weighted sequences aims at maximizing
the weight of the common subsequence, instead of its length as follows.

Definition 2 (WLCS) Given two sequences $X=<$ $x_{1}, x_{2}, \ldots, x_{m}>$ and $Y=<y_{1}, y_{2}, \ldots, y_{n}>$ and a weight function

$$
W: x_{i} \times y_{j} \rightarrow w .
$$

The weighted LCS (WLCS) problem is to find a common subsequence $Z=<z_{1}, z_{2}, \ldots, z_{l}>$ of $X$ and $Y$ such that

$$
\sum_{k=1}^{l} W\left(x_{i k}, y_{j k}\right)
$$

is maximal.

### 2.2 SVM for Classification

In this subsection we briefly review the basis of the theory of SVM in classification problems [5], [12]. Suppose we are given a set of labeled training data:

$$
\begin{equation*}
\left\{\left(\mathbf{x}_{1}, y_{1}\right),\left(\mathbf{x}_{2}, y_{2}\right), \ldots,\left(\mathbf{x}_{l}, y_{l}\right)\right\} \subset \mathcal{X} \times\{ \pm 1\} \tag{2}
\end{equation*}
$$

where $\mathcal{X}$ denotes the space of the input patterns. For each input pattern $\mathbf{x}_{i} \in \mathcal{X}$ belongs to either of two classes and is given a label $y_{i} \in\{ \pm 1\}$ for $i=1, \ldots, l$. In a support vector classification problem [5], the goal is to find a optimal hyperplane $f(\mathbf{x})$ that separates the training data with a maximal margin. That is, we want to find a linear function $f$ taking the form

$$
\begin{equation*}
f(\mathbf{x})=\mathbf{w} \cdot \mathbf{x}+b \quad \text { with } \quad w \in \mathcal{X}, b \in \mathbb{R} \tag{3}
\end{equation*}
$$

and satisfying the following constraints

$$
\begin{equation*}
y_{i}\left(\mathbf{w} \cdot \mathbf{x}_{i}+b\right) \geq 1, \quad i=1, \ldots, l \tag{4}
\end{equation*}
$$

The task can be written as a optimization problem:

$$
\begin{align*}
& \operatorname{minimize} \Phi=\mathbf{w} \cdot \mathbf{w} \\
& \text { subject to } y_{i}\left(\mathbf{w} \cdot \mathbf{x}_{i}+b\right) \geq 1, \quad i=1, \ldots, l \tag{5}
\end{align*}
$$

The key idea to solve (5) is to construct a Lagrangian function

$$
\begin{equation*}
L(\mathbf{w}, b, \boldsymbol{\Lambda})=\frac{1}{2} \mathbf{w} \cdot \mathbf{w}-\sum_{i=1}^{l} \lambda_{i}\left[y_{i}\left(\mathbf{w} \cdot \mathbf{x}_{i}+b\right)-1\right] \tag{6}
\end{equation*}
$$

where $\boldsymbol{\Lambda}^{T}=\left(\lambda_{1}, \lambda_{2}, \ldots, \lambda_{l}\right)$ is the vector of nonnegative Lagrangian multipliers corresponding to the constraints (4).

For the cases where the training data cannot be separated without error, one may want to find a soft margin hyperplane to separate the training data set
with a minimal number of errors. That is, minimize the penalty for outliers

$$
\Phi(\xi)=\sum_{i=1}^{l} \xi_{i}
$$

subject to the constraints

$$
\begin{align*}
y_{i}\left(\mathbf{w} \cdot \mathbf{x}_{i}+b\right) & \geq 1-\xi_{i}, & & i=1, \ldots, l  \tag{7}\\
\xi_{i} & \geq 0, & & i=1, \ldots, l . \tag{8}
\end{align*}
$$

The Lagrangian functional for this problem is

$$
\begin{aligned}
& L(\mathbf{w}, \xi, b, \boldsymbol{\Lambda}, \mathbf{R}) \\
& =\frac{1}{2} \mathbf{w} \cdot \mathbf{w}+C \sum_{i=1}^{l} \xi_{i} \\
& -\sum_{i=1}^{l} \lambda_{i}\left[y_{i}\left(\mathbf{w} \cdot \mathbf{x}_{i}+b\right)-1+\xi_{i}\right] \\
& -\sum_{i=1}^{l} r_{i} \xi_{i}
\end{aligned}
$$

where $\boldsymbol{\Lambda}^{T}=\left(\lambda_{1}, \lambda_{2}, \ldots, \lambda_{l}\right)$ and $\mathbf{R}^{T}=$ $\left(r_{1}, r_{2}, \ldots, r_{l}\right)$ are the vectors of nonnegative Lagrangian multipliers associated with the constraint (7) and (8), respectively.

The Kuhn-Tucker theorem plays an important role in the theory of optimization. According to this theorem, complementarity conditions are provided

$$
\begin{array}{r}
\lambda_{i}\left[y_{i}\left(\mathbf{w} \cdot \mathbf{x}_{i}+b\right)-1+\xi_{i}\right]=0, \\
r_{i} \xi_{i}=\left(C-\lambda_{i}\right) \xi_{i}=0 .
\end{array}
$$

Patterns for which $\lambda_{i}>0$ are termed the support vectors. Non-zero slack variables can only occur when $\lambda_{i}=C$. In this case, the point $\mathbf{x}_{i}$ are mis-classified if $\xi_{i}>1$. If $\xi_{i}<1$, they are classified correctly, but lie closer to the separating hyperplane $1 /|\mathbf{w}|$.

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 $K$ :

$$
\begin{equation*}
K\left(\mathbf{x}_{i}, \mathbf{x}_{j}\right)=\left\langle\phi\left(\mathbf{x}_{i}\right), \phi\left(\mathbf{x}_{j}\right)\right\rangle, \tag{9}
\end{equation*}
$$

where $\langle\cdot, \cdot\rangle$ means a dot product. The decision function $f$ in (3) becomes

$$
\begin{align*}
f(x) & =\phi(\mathbf{w}) \cdot \mathbf{x}+b \\
& =\sum_{i=1}^{l} y_{i} \lambda_{i} \phi(\mathbf{x}) \cdot \phi\left(\mathbf{x}_{i}\right)+b \\
& =\sum_{i=1}^{l} y_{i} \lambda_{i} K\left(\mathbf{x}, \mathbf{x}_{i}\right)+b, \tag{10}
\end{align*}
$$

where $\mathbf{x}_{i}$ is the image of a support vector in input space and $\lambda_{i}$ is the weight of a support vector in the feature space.

Functions that satisfy the Mercer's theorem [12] can be used as dot-products and thus can be used as kernels. Common examples of kernel functions are the polynomial kernel

$$
K\left(\mathbf{x}_{i}, \mathbf{x}_{j}\right)=\left(\mathbf{x}_{i} \cdot \mathbf{x}_{j}+1\right)^{d}
$$

and the Gaussian kernel

$$
K\left(\mathbf{x}_{i}, \mathbf{x}_{j}\right)=\exp \left(-\frac{1}{\sigma^{2}}\left\|\mathbf{x}_{i}-\mathbf{x}_{j}\right\|^{2}\right)
$$

## 3 The Feasibility of Applying Weighted LCS in 3D Handwriting Recognition

### 3.1 Data Sets and Data Preprocessing

Data is collected from an handheld device with accelerometer, such as Apple iPhone, Google HTC Phone, etc. 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.

We obtain three acceleration time series $\mathbf{a}_{x}, \mathbf{a}_{y}$, $\mathbf{a}_{z}$ 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, $\mathbf{v}_{x}=\int_{t_{0}}^{t_{N}} \mathbf{a}_{x} d t$ and $\mathbf{s}_{x}=\int_{t_{0}}^{t_{N}} \mathbf{v}_{x} d t$, where $\mathbf{v}_{x}$ and $\mathbf{s}_{x}$ are respectively the velocity and position time series of $x$-axis. The other two position time series $\mathbf{s}_{y}$ and $\mathbf{s}_{z}$ could be derived using the same method.

In order to use a single sequence instead of three to represent a letter of alphabet or a word, we have further transform the three position time series into one sequence. The method is described as follows. Suppose that

$$
\begin{aligned}
\mathbf{s}_{x} & =\left\{s_{x}(1), s_{x}(2), \ldots, s_{x}(n)\right\} \\
\mathbf{s}_{y} & =\left\{s_{y}(1), s_{y}(2), \ldots, s_{y}(n)\right\} \\
\mathbf{s}_{z} & =\left\{s_{z}(1), s_{z}(2), \ldots, s_{z}(n)\right\}
\end{aligned}
$$

are given, we could have a difference sequence as below

$$
\begin{gathered}
d \mathbf{s}_{X}(t)=\left\{\tau\left(\mathbf{s}_{X}(t)-\mathbf{s}_{X}(t-1)\right):\right. \\
\quad t=2, \ldots, n\}, \\
X \in\{x, y, z\},
\end{gathered}
$$

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 $d \mathbf{s}_{X}(t), X \in\{x, y, z\}$ into a single sequence composed of $\{0,1, \ldots, 7\}$ as follows:

$$
S\left(t_{i}\right)=d \mathbf{s}_{x}\left(t_{i}\right) \cdot 2^{0}+d \mathbf{s}_{y}\left(t_{i}\right) \cdot 2^{1}+d \mathbf{s}_{y}\left(t_{i}\right) \cdot 2^{2}
$$

### 3.2 Performance Criteria

The classification performance can be evaluated using mis-classification rate such as apparent error rate and/or graphical representation tools such as the receiver operating characteristic (ROC) curve [9].

Let the training data be denoted by $Y=$ $\left\{y_{i}: i=1, \ldots, n\right\}$, the pattern $y_{i}$ consisting of two parts, $y_{i}^{T}=\left(x_{i}^{T}, z_{i}^{T}\right)$, where $\left\{x_{i}: i=1, \ldots, n\right\}$ are the measurements and $\left\{z_{i}: i=1, \ldots, n\right\}$ are the corresponding class labels, now coded as a vector, $\left(z_{i}\right)_{j}=1$ if $x \in \operatorname{class} \omega_{j}$ and zero otherwise. Let $\omega\left(z_{i}\right)$ be the corresponding categorical class label. Let the decision rule designed using the training data be $\eta(x ; Y)$ and let $Q(\omega(z), \eta(x ; Y))$ be the loss function

$$
Q(\omega(z), \eta(x ; Y))= \begin{cases}0 & \text { if } \omega(z)=\eta(x ; Y) \\ 1 & \text { otherwise }\end{cases}
$$

The apparent error rate, $e_{A}$, is obtained by using the design set to estimate the error rate,

$$
e_{A}=\frac{1}{n} \sum_{i=1}^{n} Q(\omega(z), \eta(x ; Y))
$$

Another assessment tool for performance is the ROC curve. For a given classifier and an instance, there are four possible outcomes: true positive, false negative, true negative, and false positive. The true positive rate is

$$
\text { tp rate }=\frac{\text { Positive correctly classified }}{\text { Total positives }} .
$$

The false positive rate is

$$
\text { fp rate }=\frac{\text { Negatives incorrectly classified }}{\text { Total negatives. }}
$$

Additional terms associated with ROC curves are

$$
\begin{aligned}
\text { sensitivity } & =\text { recall } \\
\text { specificity } & =\frac{\text { True negatives }}{\text { False positives+True negatives }} \\
& =1-\text { fp rate } .
\end{aligned}
$$

Then, ROC curves are two-dimensional graphs in which $t p$ rate is plotted on the $Y$ axis and fp rate is plotted on the $X$ axis. All the ROC curves pass throughout $(0,0)$ and $(1,1)$ points and as the separation increases the curve moves into the top left corner.

### 3.3 Experimental Results

We choose some letters of alphabet for testing the significance of LCS-based pattern matching method. In the first experiment, signals for four capital letters ' A ', 'B', 'C', and 'D' are collected. For each letter, 10 patterns are collected. Then, utilizing the data preprocessing method mentioned in Sec. 3.1, the acceleration data are transformed into a sequence composed of $\{0,1, \ldots, 7\}$. We name these sequences as $S_{A, i}$, $S_{B, i}, S_{C, i}$, and $S_{D, i}$, where $i \in\{1, \ldots, 10\}$ indicates the pattern number. Using the weighted LCS algorithm, we compute the following length of WLCS:

$$
\left.\begin{array}{rl}
W \operatorname{LCS}\left(S_{X, i}, S_{X, j}\right), & \\
& \in\{A, B, C\}, \\
i, j & \in\{1, \ldots, 10\}, \\
i & \neq j, \\
W L C S\left(S_{X, i}, S_{Y, i}\right), & X, Y
\end{array}\right)\{A, B, C\}, 子 \begin{aligned}
& X \\
& X
\end{aligned}
$$

The average length of the longest common subsequences between two letters are shown in Table 1 and the histograms are shown in Figure 1. Besides, the ROC curve are shown in Figure 2. The cut-off point for best Sensitivity and Specificity is 88.00 . Using the cut-off, accuracy can be achieved at $82.62 \%$. From these experimental results, we find that the LCS-based pattern matching method for 3D handwriting recognition is feasible. In next section, we will propose the system architecture based on WLCS and SVM for the 3D handwriting recognition.

## 4 The Proposed 3D Handwriting Recognition System

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

### 4.1 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. 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 300 Hz . 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.

### 4.2 Data Preprocessing

We obtain three acceleration time series $\mathbf{a}_{x}, \mathbf{a}_{y}, \mathbf{a}_{z}$ from the previous step. In order to obtain the position time series, we can use integration calculus twice on the acceleration time series as mentioned in Sec. 3.1.

While the position time series $\mathbf{s}_{x}, \mathbf{s}_{y}$, and $\mathbf{s}_{z}$ have been derived, we have to transform them into a sequence which composed of a finite set of symbols. Suppose that $\left\{\mathbf{s}_{X}(t)\right\}_{t=t_{0}}^{t_{N}}, X \in\{x, y, z\}$ are given, we could have a difference sequence as below

$$
\begin{align*}
& d \mathbf{s}_{X}(t)=\left\{\tau\left(\mathbf{s}_{X}(t)-\mathbf{s}_{X}(t-1)\right):\right. \\
&\left.t=\left(t_{0}+1\right), \ldots, t_{N}\right\},  \tag{11}\\
& X \in\{x, y, z\}, \tag{12}
\end{align*}
$$

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 $d s_{X}(t), X \in\{x, y, z\}$ into a sequence composed of $\{0,1, \ldots, 7\}$ as follows:

$$
\begin{equation*}
S\left(t_{i}\right)=d \mathbf{s}_{x}\left(t_{i}\right) \cdot 2^{0}+d \mathbf{s}_{y}\left(t_{i}\right) \cdot 2^{1}+d \mathbf{s}_{y}\left(t_{i}\right) \cdot 2^{2} . \tag{13}
\end{equation*}
$$

The geometric meaning of transformation (13) is to mapping the difference sequence (11) into eight octant of three-dimensional Euclidean coordinate system. The above steps are depicted as shown in Figure 4.

### 4.3 Data Training

For data training, we have to prepare two sets of time series: key patterns and non-key patters. After preprocessing, we have two sets of sequences, named $K E Y$ and $N O N K E Y$. Then, we apply WLCS to all pairs of sequences selected from $\{K E Y \times K E Y\} \cup$ $\{K E Y \times N O N K E Y\}$ and we get a weight value for each pair of sequences. For example,

$$
K E Y=\left\{k_{1}, k_{2}, \ldots, k_{m}\right\}
$$

and

$$
N O N K E Y=\left\{n k_{1}, n k_{2}, \ldots, n k_{n}\right\}
$$

any selected pair of sequences can be one of the following type:

$$
\left(k_{i}, k_{j}\right), \quad i \in\{1, \ldots, m\}, j \in\{1, \ldots, n\}
$$

or

$$
\left(k_{i}, n k_{j}\right), \quad i \in\{1, \ldots, m\}, j \in\{1, \ldots, m\} .
$$

Then, the type associated with the weight values form a set for the input of support vector classifier. Thai is, the training process is performed by WLCS and SVC.

### 4.4 Pattern Recognition

After the step of data training, we have an SVM model with the form as equation (10). For a new input pattern, $a_{0 x}, a_{0 y}, a_{0 z}$, 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 (10), the model would tell us whether the new pattern is the key word $K$ or not.

## 5 Experimental Analysis

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, Test, Nathan, Wonderful $\}$. For each word, we collect at least 10 patterns from the handheld device (HTC G1 mobile phone). Figure 5 shows the three axes acceleration data of pattern 'Kimble'. Table 2 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 average accuracy of classification could be achieved at about $96.85 \%$. The performance of each classifier could be found through the ROC curves as shown in Figure 6 to Figure 10. The average accuracy of classification could be achieved at about $96.85 \%$.

## 6 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: weighted 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 average accuracy of classification could be achieved at about $96.85 \%$.

## References.

[1] Amir, A, Z. Gotthilf, and B. R. Shalom, "Weighted LCS," in IWOCA 2009, LNCS 5874, pp. 36-47, 2009.
[2] Baek, J. and Byoung-Ju Yun, "A SequenceAction Recognition Applying State Machine for User Interface," IEEE Transactions on Consumer Electronics, Vol. 54, No. 2, 2008.
[3] Chen, Qing, N. D. Georganas, and E. M. Petriu, "Hand gesture recognition using Haar-like features and a stochastic context-free grammar," IEEE Trans. on Instrumentation and Measurement, Vol. 57, No. 8, pp. 1562-1571, 2008
[4] Cormen, T. H., C. E. Leiserson, and R. L. Rivest, Introduction to Algorithms, The MIT Press, 1985.
[5] Cortes, Corinna and V. Vapnik, "Support-vector networks," Machine Learning, Vol. 20, pp. 273297, 1995.
[6] Sung-Jung Cho, Eunseok Choi, et. al., "Twostage Recognition of Raw Acceleration Signals for 3-D Gesture- Understanding Cell Phones", 10th IWFHR, La Baule, France, Oct. 2006.
[7] Eun-Seok Choi, Won-Chul Bang, et. al,"Beatbox Music Phone: Gesture Interactive Cell phone using Tri-axis Accelerometer," ICIT IEEE Int. Conference on Industrial Technology, 2005.
[8] Sung-Do Choi, A.S. Lee, Soo-Young Lee, "OnLine Handwritten Character Recognition with 3D Accelerometer", 2006 IEEE International Conference on Information Acquisition, pp.845850,2006.
[9] Fawcett, Tom, "An introduction to ROC analysis," Pattern Recognition Letters, Vol. 27, pp. 861-874, 2006.
[10] S. Kallio, J. Kela and J.Mantyjarvi, "Online gesture recognition system for mobile interaction", 2003 IEEE International Conference on Systems, Man and Cybernetics, vol 3, pp.20702076.
[11] Mitra, S. and T. Acharya, "Gesture recognition: a survey," IEEE Trans. on Sys., Man, and Cyber., Part C, Vol. 37, No. 3, pp. 311-324, 2007.
[12] Vapnik, V., The Nature of Statistical Learning Theory, Springer Press, New York, 1999.
[13] Wagner, R.A. and M.J. Fischer, "The string-tostring correction problem," J. ACM 21, pp. 168173, 1974.

Table 1: The average length of the longest common subsequence between two words in the KEY WORD set.

| Average length of LCS | A | B | C | D |
| ---: | ---: | ---: | ---: | ---: |
| A | 109 | 88 | 60 | 80 |
| B | - | 83 | 59 | 74 |
| C | - | - | 71 | 55 |
| D | - | - | - | 97 |

Table 2: The average length of the longest common subsequence between two words in the KEY WORD set.

| Average length of LCS | Kimble | Apple | Test | Nathan | Wonderful |
| ---: | ---: | ---: | ---: | ---: | ---: |
| Kimble | 274 | 222 | 146 | 220 | 224 |
| Apple | - | 249 | 150 | 206 | 208 |
| Test | - | - | 192 | 154 | 167 |
| Nathan | - | - | - | 279 | 254 |
| Wonderful | - | - | - | - | 328 |



Figure 1: Histograms. (a) Of sequences between letters ' $A$ ' and ' $A$ '. (b) Of sequences between letters ' $A$ ' and ' $B$ '. (c) Of sequences between letters ' $A$ ' and ' $C$ '. (d) Of sequences between letters ' $A$ ' and ' $D$ '.

| $—$ | ROC curve |
| :---: | :--- |
| $\square$ | Random classifier |
| $O$ | Cut-off point |


| $—$ | Mirrored ROC curve |
| :---: | :---: |
| $\square$ | Random classifier |
| 0 | Cut-off point |

ROC curve


False positive rate (1-Specificity)


True negative rate (Specificity)

Figure 2: Performance assessment using the ROC curve.


Figure 3: The architecture of the proposed 3D handwriting recognition system.


Figure 4: Data Preprocessing.


Figure 5: Three axes acceleration data of pattern 'Kimble'. (a) x-axis acceleration data. (b) $y$-axis acceleration data. (c) z -axis acceleration data.

| $\ldots$ | ROC curve |
| :---: | :--- |
| $\square$ | Random classifier |
| Cut-off point |  |

ROC curve


| $—$ | Mirrored ROC curve |
| :---: | :--- |
| $\square$ | Random classifier |
| $\bigcirc$ | Cut-off point |

Mirrored ROC curve


Figure 6: The ROC curve for the classifier recognizing the word 'Kimble'. Cut-off point for best Sensitivity and Specificity is 248 . Accuracy for these trials could be achieved at $100 \%$.

| $—$ | ROC curve |
| :---: | :--- |
|  | Random classifier |
| 0 | Cut-off point |

ROC curve


| $—$ | Mirrored ROC curve |
| :---: | :--- |
| $\square$ | Random classifier |
| 0 | Cut-off point |



Figure 7: The ROC curve for the classifier recognizing the word 'Apple'. Cut-off point for best Sensitivity and Specificity is 236. Accuracy for these trials could be achieved at $98.65 \%$.

| $\ldots$ | ROC curve |
| :---: | :--- |
| $\square$ | Random classifier |
| 0 | Cut-off point |

ROC curve



Mirrored ROC curve


Figure 8: The ROC curve for the classifier recognizing the word 'Test'. Cut-off point for best Sensitivity and Specificity is 178 . Accuracy for these trials could be achieved at $98.43 \%$.

| $—$ | ROC curve |
| :---: | :---: |
| $\square$ | Random classifier |
| $\bigcirc$ | Cut-off point |



| $—$ | Mirrored ROC curve |
| :---: | :--- |
| $\square$ | Random classifier |
| Cut-off point |  |



Figure 9: The ROC curve for the classifier recognizing the word 'Nathan'. Cut-off point for best Sensitivity and Specificity is 255 . Accuracy for these trials could be achieved at $87.19 \%$.

| $—$ | ROC curve |
| :---: | :--- |
| $\square$ | Random classifier |
| Cut-off point |  |





Figure 10: The ROC curve for the classifier recognizing the word 'Wonderful'. Cut-off point for best Sensitivity and Specificity is 280. Accuracy for these trials could be achieved at $100 \%$.


[^0]:    * Corresponding Author.

