Input Action Classification in a 3D Gesture Interface for Mobile Devices

Kayo Ogawa, Naoko Sakata, and Tomoko Muraiso
Faculty of Science
Japan Women’s University
Tokyo, Japan
kayo@optnet.jwu.ac.jp

Takashi Komuro
School of Science and Engineering
Saitama University
Saitama, Japan
komuro@mail.saitama-u.ac.jp

Abstract—In this research we propose a new motion classification method to improve operability of a 3D gesture interface that assists text input on mobile devices. A certain range of time-series finger scale data is cropped and is classified using linear discriminant analysis. To confirm possibility of linear separation, data were visualized using principle component analysis. Experimental result with changing cropping ranges and sampling rates showed that the recognition rate improved when the cropped time is longer, and more than 97.9% recognition rates were achieved with cropping time of 0.77s/0.38s from both/one sides of the peak.

Keywords- action recognition; virtual keyboard; time-series analysis; machine learning

I. INTRODUCTION

Mobile devices have become small and it is difficult to have wide operation area on their surface. For example, keypads and touch panels that are used in mobile phones, have small operation area, which degrades the operability of the devices. On the other hand, a camera-based interface which detects finger motion in the air and uses it as input has been proposed [1]. In such an interface, detection of input action, which corresponds to button pushing in a keypad interface and screen touching in a touch panel interface is an important issue. Since gesture operation in the air has no physical contact, analysis of finger motion is needed. Especially to realize high speed text input, it is necessary to detect small pushing movement of a finger.

In the previous method, relative distance of a finger from the camera was obtained from the scale (size ratio) of the finger image, and a band-pass filter and thresholding were applied to the time series distance data to detect input action. However, false recognition often occurred such as “not detected when pushed” and “detected when not pushed”, which restricted comfortable operation.

In this research we propose a new motion classification method to improve operability of a 3D gesture interface that assists text input on mobile devices.

Classification of body action is roughly divided to the case of analyzing a relatively long sequence and the case of classifying action at a certain moment. In the former case, tools like hidden Markov model (HMM) that is strong against variation in the time direction is used [2][3]. Meanwhile, in the latter case, whole time-series data is often inputted together to the classifier [4][5].

Since this research aims to detect brief moment of forward and back movement of a finger, we use whole time-series data as input. We use time-series motion parameters instead of images as they are extracted at the time of finger tracking.

II. IN-AIR TYPING INTERFACE

Figure 1 shows the appearance of the 3D gesture interface system for mobile devices that was used in the experiment. By performing finger tracking and detection of input action using a small monocular camera, keyboard typing in the air is realized. The camera is attached on a small 4.3 inch display. Image processing and recognition is performed in a PC and the result is displayed on the small display.

Figure 1 In-air typing interface

In order to obtain accurate finger position, this system registers a user’s finger image when the user starts to operate and tracking is performed by using it. Four parameters are estimated to track the fingertip: translation along the plane perpendicular to the camera’s optical axis, rotation around the optical axis, and scale. Using a high frame rate camera, it is possible to follow the fast moving finger. An example of tracking is shown in Fig. 2.
III. EXPERIMENT DATA

The ten-key application shown in Fig. 3 was used in the experiment. We chose ten-key which is simpler and requires less training than full keyboard to get data from many subjects. The frame rate of the camera was 130 fps (frames per second).

To obtain data of natural input operation, we asked the subjects to type their own mobile phone number (11 digits), the irregular series of numbers which the subjects can type on their own will. The number of subjects was six and each subject was asked to type the numbers five times respectively.

The ten-key application shown in Fig. 3 was used in the experiment. We chose ten-key which is simpler and requires less training than full keyboard to get data from many subjects. The frame rate of the camera was 130 fps (frames per second).

To obtain data of natural input operation, we asked the subjects to type their own mobile phone number (11 digits), the irregular series of numbers which the subjects can type on their own will. The number of subjects was six and each subject was asked to type the numbers five times respectively.

The finger scale data was used for analysis. An example of the obtained data is shown in Fig. 4. Some peaks can be seen in the finger scale. Between the peaks, the finger was moving to the next key and the scale change is small.

To analyze the motion feature, both input motion and non-input motion were cropped. In cropping of input motion, 50 points before and after the peak of the scale, total 100 points (0.77[s]) were taken as one sample. In cropping of non-input motion, 100 points around the center position between peaks were taken. Data whose peak position could not read were removed. As a result, the number of data for input motion was 161 and that for non-input motion was 146.

In addition, for investigating the effect of camera frame rate, data at the sampling rates of 65Hz and 33Hz were created by resampling the 130Hz data. Also for investigating the effect of cropping time, data with the range of 0.46s, the shortest time with which one can identify input motion and 0.23s, which is the half-value width of the peak value, were created.

To perform action detection in real time without delay, action classification has to be done in the middle of input action. Therefore the data with the points only before the peak were also created (we call it “one side” data) for comparison to the “both sides” data with the points before and after the peak. The detail of the data used in the experiment is shown in Table 1.

IV. DATA VISUALIZATION

Principal component analysis (PCA) can compress data to principal components while minimizing the loss of information that the data essentially have. Each principal component is orthogonal to the others and independent features can be extracted. The principal components are expressed by the following equation.

\[ Z_m = \sum_{p=1}^{P} a_{pm} x_p \]  

(1)
$Z_m (m=1,…,M)$ is the $m$-th principal component, $a_{pm}$ is the combination coefficient, $x_p (p=1,…,P)$ denotes the data, and $P$ is the number of variables. $a_{pm}$ is calculated so that the variance of principal component becomes maximum (eigenvector), and the principal component score of each sample is obtained by substituting the value and measured data into the equation above. By plotting them in the principal component space, it is possible to visualize the feature of the data.

PCA was applied to each data set. Each of both sides, 0.77s data sampled at all frequencies, both sides, 0.46s data at 130Hz and 65Hz, one side, 0.38s data at 130Hz and 65Hz, and one side, 0.23s data at 130Hz was compressed to two dimensional data. Each of other data was compressed to one dimensional data. Here, we considered the components whose eigenvalue was more than 1 as significant components.

Table 2 shows the result of both sides / 0.77s / 130Hz data and one side / 0.38s / 130Hz data, and Fig. 5 shows the eigenvectors for each data. As seen from the wave shape, the first eigenvector shows the gradual scale change due to key selection and the second eigenvector shows the difference between input and non-input motions. Therefore, we expect that the data, of which up to second principal components are significant, are suitable for classification.

Figure 6 shows the graph of plotting principal component scores. From the result, it was confirmed that the plot clusters of input and non-input motion are clearly separated and are linearly separable.

<table>
<thead>
<tr>
<th>TABLE II. PCA RESULTS</th>
<th>(a) both sides / 0.77s / 130Hz</th>
</tr>
</thead>
<tbody>
<tr>
<td>principal component</td>
<td>eigenvalue</td>
</tr>
<tr>
<td>1</td>
<td>92.122</td>
</tr>
<tr>
<td>2</td>
<td>6.109</td>
</tr>
<tr>
<td>3</td>
<td>.835</td>
</tr>
<tr>
<td>4</td>
<td>.437</td>
</tr>
<tr>
<td>5</td>
<td>.139</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>(b) one side / 0.38s / 130Hz</th>
</tr>
</thead>
<tbody>
<tr>
<td>principal component</td>
</tr>
<tr>
<td>1</td>
</tr>
<tr>
<td>2</td>
</tr>
<tr>
<td>3</td>
</tr>
<tr>
<td>4</td>
</tr>
<tr>
<td>5</td>
</tr>
</tbody>
</table>

(a) both sides / 0.77s / 130Hz
(b) one side / 0.38s / 130Hz

Figure 5 Eigenvectors

(a) both sides / 0.77s / 130Hz
(b) one side / 0.38s / 130Hz

Figure 6 Plotted principal component scores
V. ACTION CLASSIFICATION

Since the possibility of linear separation was confirmed, we applied linear discriminant analysis (LDA) to all the data. Leave-one-out cross validation was used to separate training data and testing data. The results are shown in Table 3. Average rates were calculated considering the numbers of input and non-input data. The graph of average rates with changing cropping ranges and frame rates is shown in Fig. 7.

From the results, it was confirmed the there were little difference of recognition rates between both sides and one sides cropping. There were also little difference according to sampling frequencies. The major factor that affects the recognition rate was the cropping time and the recognition rates were more than 97.9% with the cropping time of 0.77(0.38)s, more than 96.8% with the cropping time of 0.46(0.23)s and 89.8% with the cropping time of 0.23(0.12)s. The longer the cropping time was, the higher the recognition rate became.

VI. CONCLUSION

In this research we investigated the method of motion classification in 3D gesture recognition for mobile devices. PCA was applied to the time-series scale data and linear separation of input and non-input motion was confirmed. By applying LDA to the data, more than 97.9% recognition rates were achieved with cropping time of 0.77s/0.38s from both/one sides of the peak.

Future works will include real time input action detection in a real application by performing classification with sliding the cropping range of the time series data. Since we obtained high recognition rates equally for both sides and one side data, it is highly likely that real time recognition with no delay can be realized.

REFERENCES