The Text Input System Using Spatial Finger Touch Method

YANG-KEUN AHN, KWANG-SOON CHOI, YOUNG-CHOONG PARK
Realistic Media Platform Research Center
Korea Electronics Technology Institute
121-835, 8th Floor, #1599, Sangam-Dong, Mapo-Gu, Seoul
REPUBLIC OF KOREA
ykahn@keti.re.kr

Abstract: - The present study proposes a way of entering letters by recognizing gestures involving slightly folding and unfolding the fingertips in a smart TV environment. To recognize fingertip gestures, an iterated local search algorithm (ILSA)-based gesture recognition algorithm was designed, and, to test this algorithm, an interface that operates in a Leap Motion camera was realized. To verify whether it was actually possible to enter letters remotely, two programs were employed based on the realized interface: LeapKII-SW, which recognizes gestures, and LeapKII-GUI, which prints out the letters recognized. These programs were then tested in an environment similar to smart TVs and, through a comparative analysis with existing studies, were confirmed to exhibit an enhanced character input speed.

Key-Words: - Gesture, Gesture Recognition, Hand Gesture, Character Input, Spatial Touch, Finger Click, Leap Motion, Smart TV

1 Introduction

With the ongoing development of information and communications technology (ICT), it has become possible for users to enjoy diverse media contents at any time and in any place through a variety of devices. However, in terms of interaction, users must still use unnatural user interfaces (UI) while also enduring the inconvenience of having to learn new interfaces each time they access another device. Consequently, to support more natural and convenient interactions between users and diverse devices and media contents, research on gesture-based UIs which make use of the user's movements have been actively conducted both at home and abroad.[1]

Ishikawa Oku's laboratory at the University of Tokyo has proposed a vision technology-based space touch method using high-speed cameras installed on mobile devices.[2] However, the suggested template-based fingertip recognition method for recognizing and tracking the fingertips requires camera devices with high frame rates to track the fingertips, and an initialization stage for recognition.

To detect the fingertips, Takeoka, et al. have proposed the Z-touch method using high-speed cameras and line lasers.[3] Due to its processing speed, however, Z-touch has failed to accurately estimate the consecutive depth values of the fingertips. Tsukada, et al. have enhanced the accuracy of the estimation of the coordinates of the fingertips by overlapping two panels in a layer form. However, equipment costs have increased with the use of two layers.

Using many cameras, Itai Katz at Stanford University has increased the accuracy of fingertip recognition and the depth values of the fingertips in comparison with existing interfaces and also reduced equipment costs.[4]

By using geometric characteristics, where the position and the overall direction of the hand do not change when finger gestures are entered, Joseph J. La Violar at the NSF Center was able to quickly recognize diverse gestures even without pattern recognition.[5] Depending on the gestures to be recognized, the introduction of a pattern recognition algorithm into gesture recognition methods has become optional.

LeapKII, the character input interface proposed in the present study, is a system where, to recognize the fingertips accurately, Leap Motion devices are used to accurately obtain their positions, and extreme points are found in the obtained fingertip trajectories, thus recognizing gestures. Gestures are recognized using numerical analysis instead of pattern recognition, which requires considerable calculations, with the further advantage that gestures can be entered consecutively. In addition, by applying the suggested method to actual smart device equipment, its performance was evaluated through a comparative analysis of existing studies.

2 Proposed Method

As illustrated in the Fig. 1, in the overall algorithm, LeapKII-SW analyzes fingertip trajectories, recognizes gestures, and transmits command packets to LeapKII-GUI, thus printing out letters. Detailed algorithms will be explained in Sections 2.1 and 2.2.

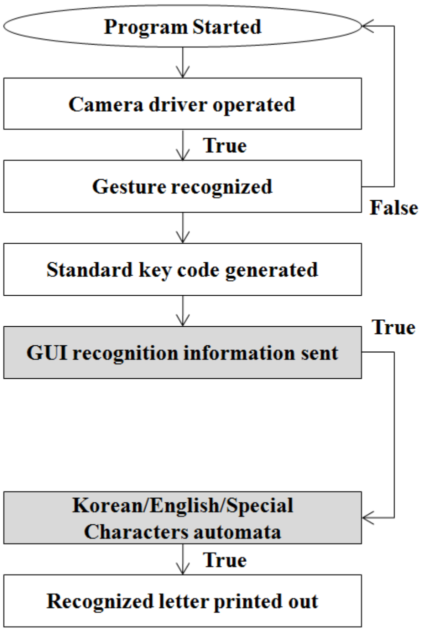

Fig. 1 System flowchart

2.1 LeapKII-SW: Gesture Recognition

LeapKII-SW is a program that analyzes the user's fingertip trajectories to recognize defined gestures, and then commands GUI to print out letters, operating as in the Fig. 2.

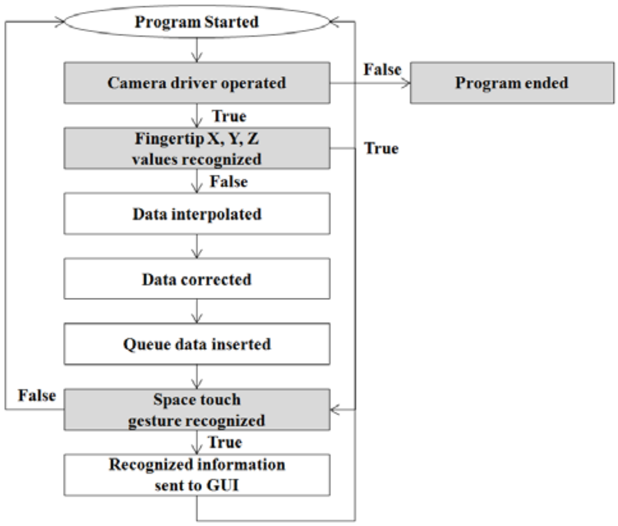

Fig. 2 LeapKII-SW flowchart

2.2 Fingertip Recognition

Unlike interfaces that use costly infrared cameras, the proposed interface uses affordable Leap Motion cameras to recognize the user's fingertip trajectories. Leap Motion cameras are located near the lower part of the user's hand, and the LeapKII program receives the coordinate data by using the embedded controller. Because the coordinates received have not been assigned according to the locations of the hands, the data are sorted to match the order of the fingers by using ascending sorting.

```
procedure bubbleSort( A : list) defined as:
 for each i in 1 to length(A) do:
     for each j in length(A) downto i + 1 do:
      if A[ j ] < A[ j − 1 ] then
        swap( A[ j ],  A[ j − 1 ] )
      end if
     end for
 end for
end procedure
```

Fig. 3 Ascending-order pseudocode

The Fig. 4 shows the fingertip trajectories of the thumb, middle finger, and index finger recognized by LeapKII-SW. In the diagram, the fingertip trajectory value of the middle finger between the 10th frame and the 17th frame is 0. Leap Motion recognizes fingertip trajectories by using two cameras, so that when the gesture of slightly folding and unfolding the middle finger is entered, two fingers overlap, making it difficult to recognize each fingertip trajectory accurately.

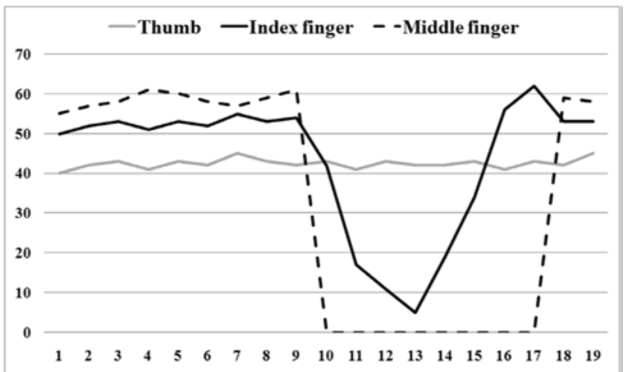

Fig. 4 Fingertip trajectories

2.3 Data Interpolation

When errors occur, the cubic Bézier curve is used to estimate fingertip trajectories to ensure accurate recognition.

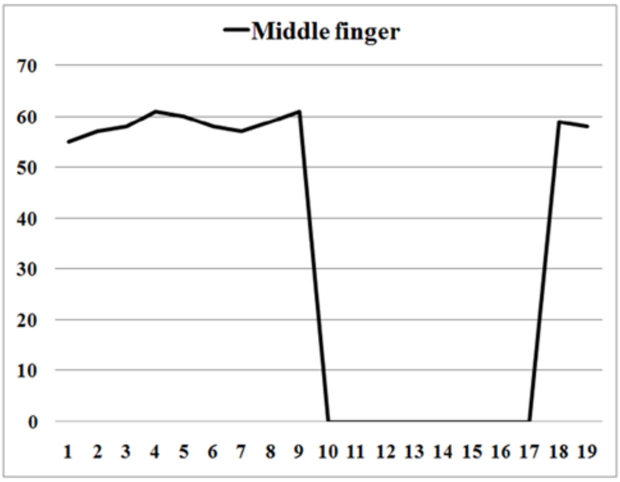

Fig. 5 Failure to recognize fingertip trajectories

The basic Bézier curve estimation algorithm is as in the equation 1:

(1) $P(t) = \sum_{i=0}^{N-1} B_i J_{ni}(t)$
Bezier curve

The results interpolated with estimations are shown in the Fig. 6, and display a pattern similar to that of the trajectories of slightly folding and unfolding the fingertips, because they have been formed through the Bézier curve.

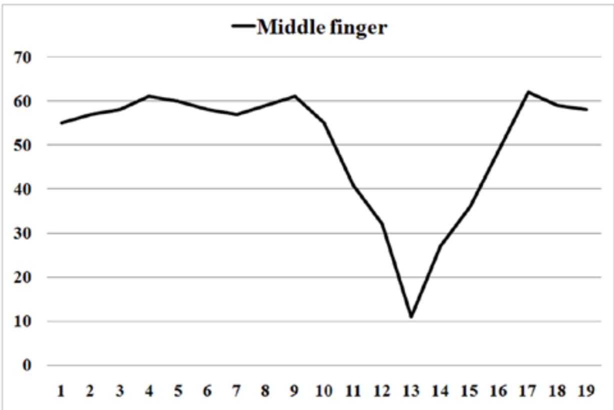

Fig. 6 Interpolated fingertip trajectories

2.4 Gesture Recognition

The gestures to be recognized were those of slightly folding and unfolding one finger, with the hand completely open.

When the gestures are entered using the middle finger, the data on the finger trajectories form a graph curving downward, as in the Fig. 7. In graphs of this form, the shaded area is called a partial graph with extreme points; here the 13th point, marked with a circle in the partial graph, is called the local minimum. In the present study, the occurrence of a touch event is defined as recognition of the local minimum.

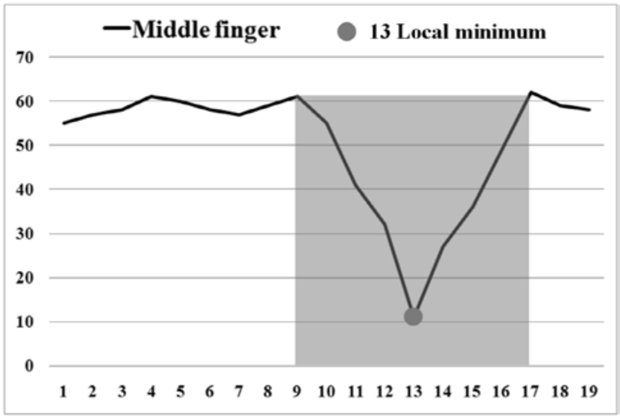

Fig. 7 Local minimum graph

In the present study, to raise the accuracy of recognition, several constraints were established in terms of the local minimum. First, the partial graph with the local minimum must have a range of 10 frames (minimum) and 50 frames (maximum). Second, the point detected must differ in size from the maximum values at either end by 20 or more. Third, data before the point to be detected must not include the point to be detected.

Based on the definitions above, the present study used an improved golden section method to detect the local minimum.[6] While the local minimum search method using the golden section established the section point as the midpoint between the starting and ending points, the present study established the end points of the section in which the local maximum occurred as the section points. This was to prevent graphs with local minimums from overlapping.

By using section points such as those above, consecutive trajectories from consecutive trajectory data were divided into partial graphs, and testing determined whether or not a local minimum existed.

Such an algorithm is repeated with each point on the graph, which makes normal recognition possible even when the thumb, index finger, and middle finger alternately enter gestures.

2.5 Layer Recognition

In the LeapKII program, there are six layers, with the X and Z axes of Leap Motion as the standards, as defined in the Fig. 8. In the diagram, the circle indicates the location of the user's fourth finger and determines the layer. When entering the gestures of other fingers, it was used as the standard, indicated as a circle in the diagram below, because the sizes of changing coordinate values were the smallest.

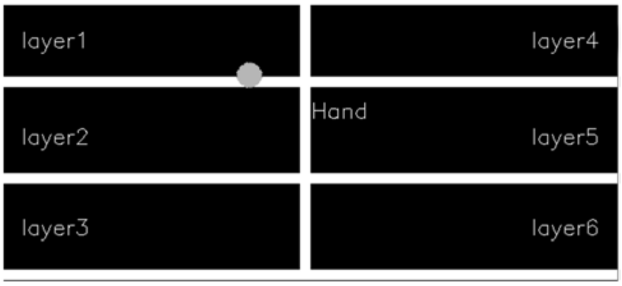

Fig. 8 Layer view

2.6 Standard Key Code Transmission

When it recognizes gestures, LeapKII-SW transmits character output command codes to the GUI program. An example of a standard transmission packet according to the gestures recognized, layers, and finger values are shown in the Fig. 9.

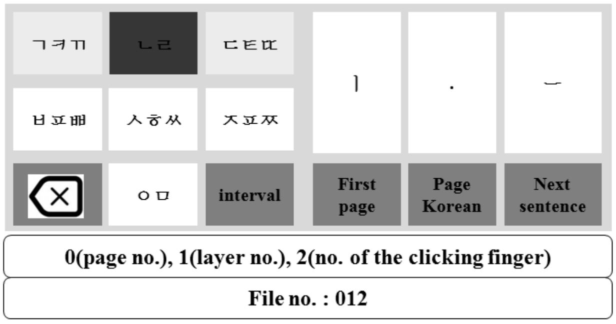

Fig. 9 Standard file transmission packet

3 Character Output Algorithm: LeapKII-GUI

As in the Fig. 10, the LeapKII-GUI interface receives and prints out standard key codes on the screen. Because the character input method follows the Cheonjiin keyboard standard, when stroke-adding key codes are transmitted, previous letters are deleted and expanded letters are entered. For example, when 'ㅣ'and stroke-adding letter '•'are entered in succession, in the end, the letter 'ㅏ'is printed out.

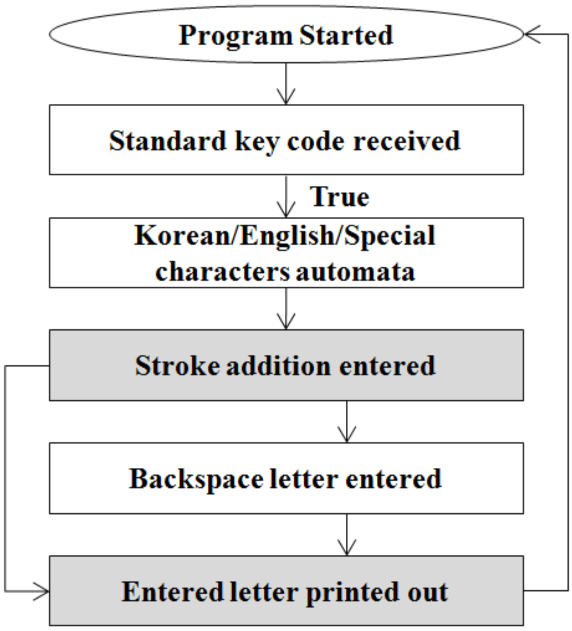

Fig. 10 LeapKII-GUI flowchart

4 Experiments

For the experiment, Leap Motion cameras were connected to a personal computer (PC) with a central processing unit (CPU) with a3.4 GHz Intel(R) Core(TM) i7-2600K and an 8-Gbyte memory at 150 FPS. Tests were conducted, and the software processing time is shown in Table 1 below. Most of the time was taken to recognize fingertip trajectories, and approximately 10 ms was additionally required after the finger had been pressed to find the local minimum, to recognize touch gestures.

Table 1 Required processing time

Type	Processing time (msec)
Fingertip recognition	13 msec
Preprocessing	2 msec
Search for the local minimum	11 msec
Total	26 msec

The user consecutively entered gestures 453 times in LeapKII with the intention of entering particular gestures, and the gestures were correctly recognized 379 times and erroneously 74 times. While recognition programs accurately perform the same algorithms for the given data, trajectories cannot be accurately recognized when the fingertips overlap in the videos recorded by Leap Motion cameras, thus leading to problems.

5 System Structure

The present system uses a virtual keyboard printed out on the display and defined gestures are used to enter letters. The system overall consists of LeapKII-SW, which recognizes gestures, and LeapKII-GUI, which converts the gestures recognized into letters and prints them out, as in the Fig. 11.

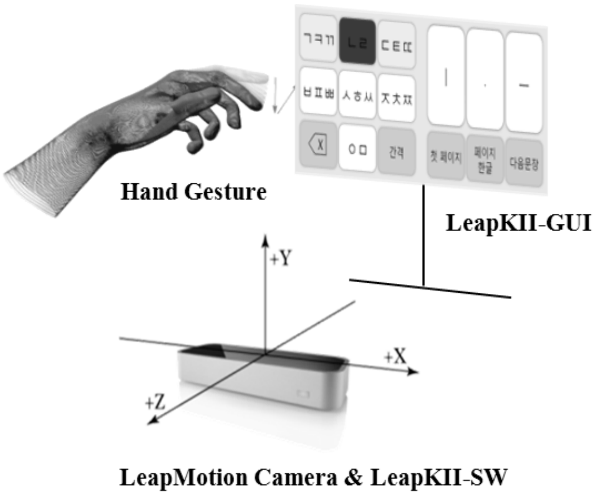

Fig. 11 LeapKII system structure

5.1 Basic Requirements: Gesture Definitions

As in the Fig. 12, the gestures entered by users were the movements of slightly folding and unfolding the fingers. In consideration of the recognition rate and users'convenience, only three fingers, the thumb, index finger, and middle finger, were used to enter gestures.

Fig. 12 Gesture fingertip trajectories

5.2 Basic Requirement: Virtual Keyboards

The letters printed out according to the gestures entered are shown in the Fig. 13. The user selects the layer by moving the hand at the top of the Leap Motion cameras and can print out letters by entering the gestures.

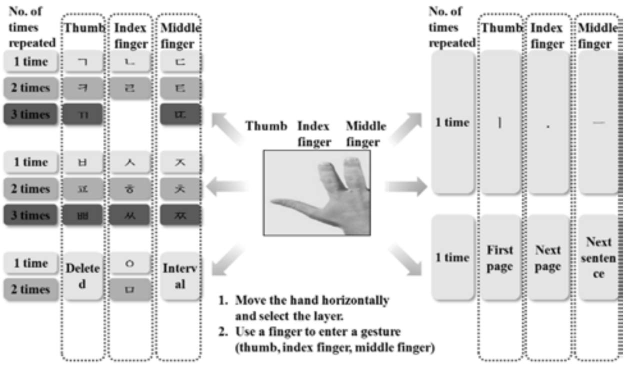

Fig. 13 Virtual keyboard

6 Performance Evaluation and Future Plans

The present study has suggested an enhanced version of an algorithm from existing research for space touch-based gesture recognition, and utilized LeapKII-SW/GUI for experiments. In the future, experiments will be conducted to make remote character input possible by actually interlocking Leap Motion cameras to smart TVs.

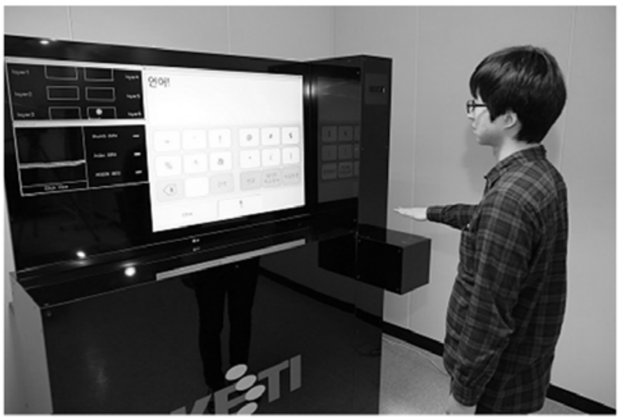

Fig. 14 LeapKII interface structure

References:

[1] Dongpyo Hong, Woontack Woo, "Recent Research Trend of Gesture-based User Interfaces," Telecommunications Review, Vol. 18, No. 3, June 2008, pp. 403-413.

[2] Y. Hirobe, T. Niikura, Y. Watanabe, T. Komuro, M. Ishikawa, “Vision-based Input Interface for Mobile Devices with High-speed Fingertip Tracking,” Adj. Proc. ACM UIST, October, 2009.

[3] Y. Takeoka, et al., "Z-touch: An Infrastructure for 3D Gesture Interaction in the Proximity of Tabletop Surfaces," ITS 2010: Devices & Algorithms, November, 2010, pp. 91-94.

[4] Erez Ponser, "A Single Camera-based Floating Virtual Keyboard with Improved Touch Detection," Electrical & Electronics Engineers in Israel (IEEEI), 2012 IEEE 27th Convention of, November, 2012.

[5] Joseph J. La Viola, Jr., "A Survey of Hand Posture and Gesture Recognition Techniques and Technology", Brown University Technical Report, Jun, 1999.

[6] Kiefer, J. (1953), “Sequential minimax Search for a Maximum,” Proceedings of the American Mathematical Society 4(3), 1953, pp. 502-506.