Infrared Camera-Based Hand Gesture Space Touch System Implementation of Smart Device Environment

Yang-Keun Ahn, Kwang-Soon Choi, Young-Choong Park, Kwang-Mo Jung

***Abstract*—**This paper proposes a method to recognize the tip of a finger and space touch hand gesture using an infrared camera in a smart device environment. The proposed method estimates the tip of a finger with a curvature-based ellipse fitting algorithm, and verifies that the estimated object is indeed a finger with an ellipse fitting rectangular area. The feature extracted from the verified finger tip is used to implement the movement of a mouse and clicking gesture. The proposed algorithm was implemented with an actual smart device to test the proposed method. Empirical parameters were obtained from the keypad software and an image analysis tool for the performance optimization, and a comparative analysis with conventional research showed improved performance with the proposed method.

***Keywords*—**Infrared camera, Hand gesture, Smart device, Space touch.

I. Introduction

WITH the recent expansion of the mobile market and smart devices, smart mobile devices are extensively used in many different places. With this trend, the amount of information to be displayed on-screen is increasing, while the error rate in controlling smart devices through direct touch is also increasing due to the miniaturization of such devices for improved portability.

Recent studies have proposed a method resolve this problem. In the method, a finger tip in the space is recognized to provide an operation pointer and achieve gesture recognition. Ishikawa Oku laboratory of Tokyo University proposed a vision-technology-based space touch method utilizing a high-speed camera installed in the mobile device [1]. However, the template-based finger tip recognition method proposed to recognize and track the tip of a finger requires a camera with a high frame rate and an initializing stage. Takeoka et al. proposed the Z-touch method, which utilizes a high-speed camera and line-laser to extract the tip of a finger [2]. However, Z-touch could not accurately estimate the continuous depth values of a finger tip due to the processing speed. Tsukada et al. improved the accuracy of finger tip estimation by overlapping two panels in the form of layers. However, this leads to additional cost [3].

To accurately recognize a finger tip, this paper proposes a method that utilizes the ellipse fitting method to accurately estimate the location of a finger tip and estimate a relatively accurate depth value from a single infrared camera. In addition, the proposed method was implemented on a smart device to evaluate its performance through a comparative analysis with conventional research.

There are two main assumptions regarding hand shapes for gesture recognition.

- Assumption 1. The finger tip for virtual touch is always at the top of the image.
- Assumption 2. The finger tip area for virtual touch is larger than a certain area.

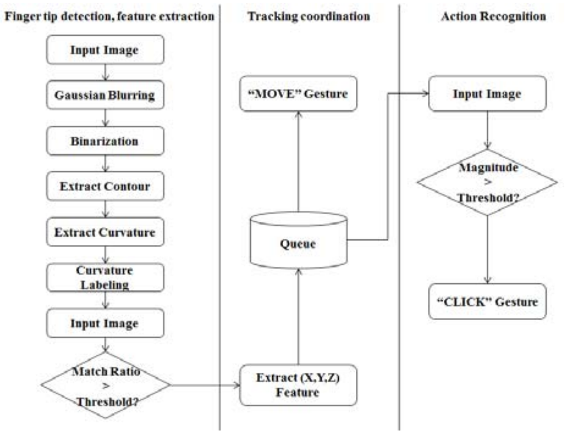

Fig. 1 Flowchart of system control

The system this paper proposes is operated through the finger tip and feature extraction part, coordinate tracking part, and motion recognition part. Fig. 1 illustrates the control flow chart of the system.

II. Pre-Processing

A smoothing computation for noise reduction is conducted on the input image. Among various smoothing methods, Gaussian kernel is known to be an effective noise reducing method. Noise is reduced through Gaussian smoothing, and the object is separated through binarization. Infrared-camera-based images utilize infrared lighting, in which the brightness level increases with respect to the distance of the object, and they are easily binarized as constants. Binarized images are expressed as number of clusters with a labeling algorithm, and small clusters are removed according to assumption 2 of section I. Outlines are extracted from the final clusters to extract the finger tips from the images. The outline extraction method used in this

Yang-Keun Ahn, Kwang-Soon Choi, Young-Choong Park, and Kwang-Mo Jung are with the Korea Electronics Technology Institute Seoul, Korea (e-mail:ykahn@keti.re.kr).

paper is an outline extraction function provided by the open library OpenCV [4], and the binarization threshold of 80 is used.

III. Curvature-Based Finger Tip Extraction

The user hand shape used in this system is illustrated in Fig. 2. When the hand shape is defined as illustrated in Fig. 2, the tip of the finger has a certain curvature. The finger tip candidate areas are estimated and verified with the curvature information of finger tips. The algorithm from [5] was used as the algorithm for curvature estimation and ellipse fitting [5]. The finger tip feature used in gesture recognition is estimated from the verified area.

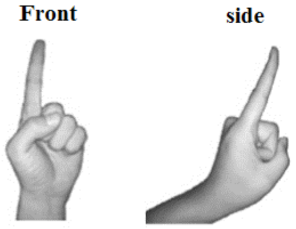

Fig. 2 Hand shape for recognition of motion

A. Curvature-Based Candidate Area Extraction

Curvature is an indicator of the degree of bending of a curved surface, and it is computed from (1).

$$K_i = \frac{\overrightarrow{p_i p_{i-1}} \cdot \overrightarrow{p_i p_{i+1}}}{||\overrightarrow{p_i p_{i-1}}|| \, ||\overrightarrow{p_i p_{i+1}}||} \qquad (1)$$

p_i represents the ith location of the outline. And each p_{i-l}, p_{i+l} represents the locations of the pixels a certain distance l away from the location of the ith pixel in the set of the outline pixels. $\cdot$ represents the inner product of two vectors. Curvatures are computed from every outline pixel, and the pixels with low curvatures are removed because the finger tip areas have shapes with the feature of strong curvatures. Fig. 3 illustrates pixels with the curvature estimated from the extracted outline images.

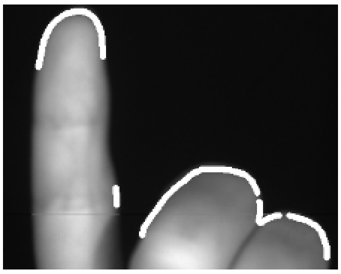

Fig. 3 Curvature pixel from the extracted outline images (white line)

Curvatures are widely distributed in the finger tip area of the figure above. However, the joint areas also have curved shapes; thus, those areas are also extracted as the curvature areas. Curvature pixels are connected with the neighboring pixels to represent them as clusters to remove the pixels in the areas other than the finger tip areas. The connections with neighboring pixels are indexed as clusters when a curvature exists in reference to the neighboring pixels. The sets of clustered curvatures are sorted in descending order with respect to the Y axes of the average locations of each cluster after the locations of the pixels of each cluster are averaged for finger tip verification. The meaning of sorting in a descending order is expressed in assumption 1 of Section II.

B. Finger Tip Verification Utilizing Ellipse Fitting

A curvature pixel-based ellipse suite algorithm is used for an accurate finger tip estimation and finger tip verification robust to rotation. The accurate location of a finger tip is estimated by utilizing an ellipse fitting algorithm in the finger tip area, and it is used as the template matching input image for finger tip verification. A function provided by OpenCV is used for ellipse fitting [4]. The ellipse fitting algorithm is a least-square-based ellipse fitting method that extracts the lengths of the major and minor axes of an ellipse as well as the rotational angle. The topmost point of the major axis of the ellipse is estimated as a $(\mathrm{x}, y)^T$ feature of the finger tip, and an ellipse fitting rectangular is generated with the rotational angle and the major and minor axes of the ellipse.

The ellipse fitting rectangular area is employed as the input image of the template matching algorithm of finger tip verification. The sorted finger tip candidate clusters are rotated and size converted to fit the template images using ellipse fitting, and an image is selected as a finger tip area and the features are extracted when the matching rate is above a certain threshold δ. Template matching is a matching method of measuring the similarities between two images by employing a normalized correlation matching method [6]. The equation for the normalized template matching method is shown in (2).

$$R_{ncr} = \frac{R_{cr}}{P}$$
$$R_{cr} = \sum_{x,y} [T(x,y) \cdot I(x,y)]^2$$
$$P = \sqrt{\sum_{x,y} T(x,y)^2 \sum_{x,y} I(x,y)^2} \qquad (2)$$

Here, P represents the normalization factor, $T(x,y)$ represents pixel value in the (x,y)-coordinate of the template image, $I(x,y)$ represents the pixel at the (x,y)-coordinate of the input image. R_{ncr} represents the matching rate and R_{cr} r represents the degree of correlation matching. Fig. 4 illustrates the template matching process

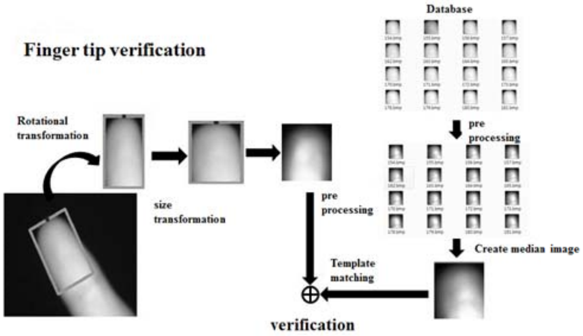

Fig. 4 Finger tip verification

C. Finger Tip Verification Utilizing Ellipse Fitting Template Image Generation

100 finger tip images are collected to generate a template image. In an infrared-based camera, the reflectivity increases as the hand gets closer to the camera. Therefore, Histogram equalization is required for the collected finger tip images to be robust against lighting variations. A template image is generated from an average image of the 100 collected learning images after the histogram equalization.

Fig. 5 illustrates a DB collected for histogram equalization pre-processed average image and histogram equalized DB. The employment of an infrared image with little noise and few texture features leads to a robust matching rate to the template generation. However, the lighting varies with respect to distance, which leads to variation in contrast. Therefore, pre-processing is conducted with the histogram equalization algorithm to ensure that the matching is robust against light variation.

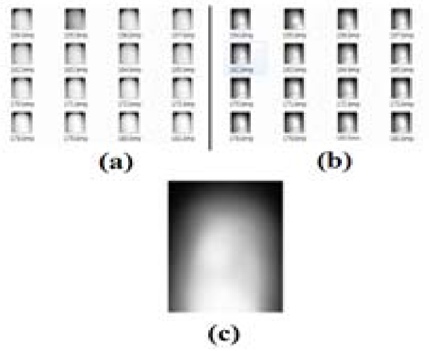

Fig. 5 Generate median Image:
(a) Collected finger-tip area DB (b) DB after histogram equalization (c) median Image

D. Feature Extraction

The features for the gesture recognition are extracted from the verified finger tip areas. Reference [1] implemented the Lucas-Kanade template tracker to employ the width length of the template area as the depth feature z to recognize the click gesture. However, since this method utilizes the template tracker, it is difficult to obtain the feature when tracking in the template area fails, and the features of the z-axis are not definite. Therefore, the width features of the finger tips are defined as the z-axis features, and the minor axis of the ellipse fitting area of the finger tips is employed.

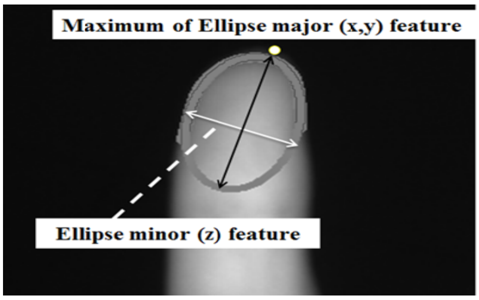
Maximum of Ellipse major (x,y) feature
Ellipse minor (z) feature

Fig. 6 Finger-tip detection and feature extraction result

IV. Hand Gesture Recognition

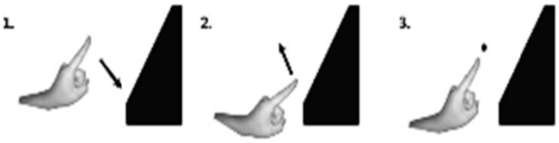

Fig. 7 Hand gesture for character input

The hand gesture for character input is illustrated in Fig. 7. Make a fist and extend the index finger. Then position the palm to be parallel to the camera, and make an instant gesture towards the direction of the camera. Take a gesture similar to pressing a button in the space.

A. High-Frequency Components Extraction Using Time Differentiation

The gesture for character input implemented click motion of a mouse. High-frequency components are extracted to treat the radical variations of the extracted z-features. The high-frequency components utilized here refer to the interval during which the z value changes radical with respect to time axis T. Fig. 8 shows the variation of the Z-axis value with respect to time axis T in relation to a certain gesture. The difference between the z values of the intervals with the movement event and the click event is evident.

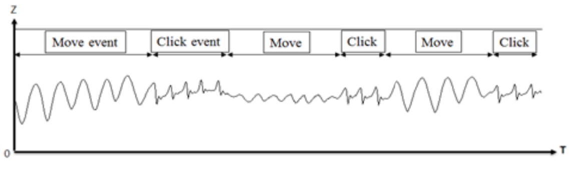

Fig. 8 Finger-tip Z-axis value variation analysis

To effectively estimate the instant variations of the z values, this paper utilizes a certain mask filter to approximate the frequency strength. Fig. 9 illustrates the mask filter for the estimation of the high-frequency component.

-1	-1	-1	-1	8	-1	-1	-1	-1

Fig. 9 Mask filter for estimation of high-frequency

However, the employed mask filter is an approximation mask filter that might mistakenly recognize the components from the movement interval as the high-frequency component due to the degrees of noise and variation. Therefore, the estimated z value differentiated with respect to time is used as the feature value employed in the frequency component analysis. The new z-axis features can be estimated from (3).

$$z'[t] = z[t-1] - z[t+1]$$
$$\mathrm{mag[t]} = \sum_{i=0}^{N} z'[i+t+1] \cdot m[i] \qquad (3)$$

Here, z$[t]$ refers to the signal of then Z-axis in t-time, and $z'[t]$ refers to the t-approximated differentiated value. $m[i]$ represents the high-frequency approximation mask filter, and mag[t] represents the strength of the high-frequency component approximated at t. Fig. 10 illustrates the frequency strength approximated after passing the mask filter featuring the differentiation of z signal with respect to time.

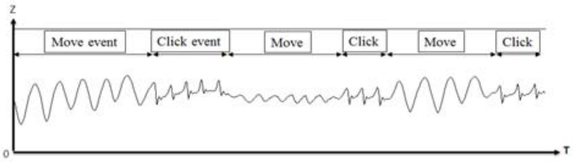

Fig. 10 The frequency strength approximated after passing the mask filter

V. System Composition

An actual character input system, SmartUX-MobileKII, was implemented to evaluate the performance of character input using the space touch technology this paper proposes. The system comprises movement and click gestures analyzed in a gesture recognition module regarding the input through an infrared camera, and the recognition results are displayed in the keypad. Fig. 11 shows the simple system composition and flow.

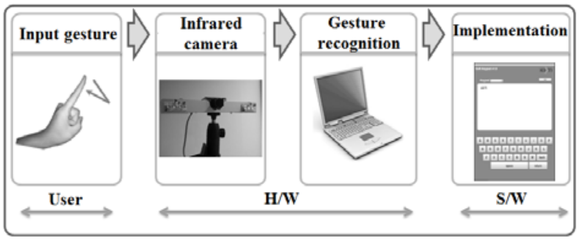

Fig. 11 Gesture recognition system

A. References Hardware Composition

Fig. 12 shows the hardware composition of SmartMobileKII. For the hardware composition, then HDMI-interface-based display, the output module of the keypad, is installed at the top, and the infrared camera for the image input is embedded at the bottom. The gesture analysis module is a regular desktop PC connected to the camera and USB interface, and the PC is connected to the output module via the HDMI interface.

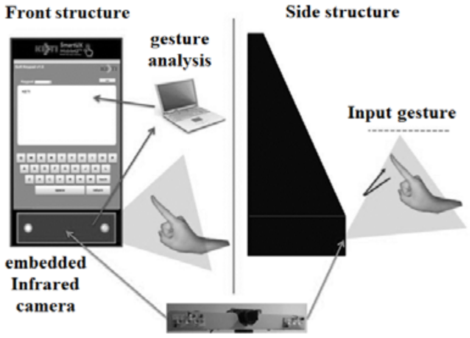

Fig. 12 System hardware structure

B. Software Composition

Fig. 13 shows the software composition of the SmartUX-MobileKII system. The software comprises a parameter display to control parameters (left top), main display to display the input images (middle top), binarization display to output the binarization result (left bottom), finger tip extraction display to display the finger tip extraction and feature area (middle bottom), and frequency analysis display for analysis of the features of z (right top and right bottom). The implemented system employed SmartUX-MobildeKII software to estimate the empirical parameters to upgrade the performance.

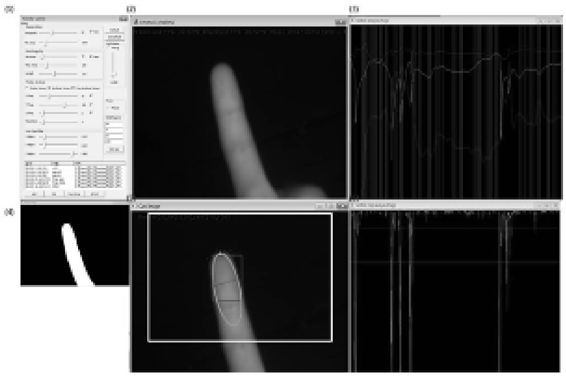

Fig. 13 Gesture analysis system GUI

VI. Experiment Result

The development environment comprises the Windows 7 OS-based development tool Visual Studio 2010, the MFC environment, and the hardware includes a FireFly MX 60 frame/sec infrared camera, an HDMI interface display, a desktop PC Intel i7-2600k CPU, 3.48 GB. The empirical parameters are summarized as follows: the binarization threshold of 80, the curvature extraction mask size used in finger tip extraction of 114, the minimum number of pixels with respect to assumption 2 of 20, the curvature threshold of 0.96, the template matching threshold of 0.93, and the frequency component threshold with respect to the z value in gesture recognition of 120. Each parameter was obtained using image analysis software. For S/W specification, the camera recognized gestures positioned between the minimum distance of 11cm and the maximum distance of 30cm from the camera, and the gesture recognition yielded the best performance within the optimal distance of 20cm±5cm.

Experiments were conducted with the produced smart device for comparative analysis of the Lucas-Kanade template tracker [7] of reference [1], and the proposed method and to evaluate the speed and accuracy.

The frame rates per second at the resolutions of 320x240 and 640x480 were compared.

TABLE I
SPEED AND ACCURACY (FRAME/SEC)

	320x240	640x480
Lucas-Kanade[7]	53	23
Proposed method	60	52

The proposed method yielded averages of 60 and 52 frames at resolutions of 320x240 and 640x480, respectively, and the Lucas-Kanade tracker yielded averages of 53 and 23 frames at resolutions of 320x240 and 640x480 resolutions, respectively. As the resolution increased, the template image of Lucas-Kanade increased, and the computations required for parameter extraction also increased. Consequently, the frame rate of Lucas-Kanade is decreased by a relatively large magnitude.

Keypad input of 10 English pangrams provided by Wikipedia [8] was conducted using each gesture to analyze the accuracy, and the numbers of recognition failures and falsely detected strings were compared. Fig. 14 shows the results of the accuracy test.

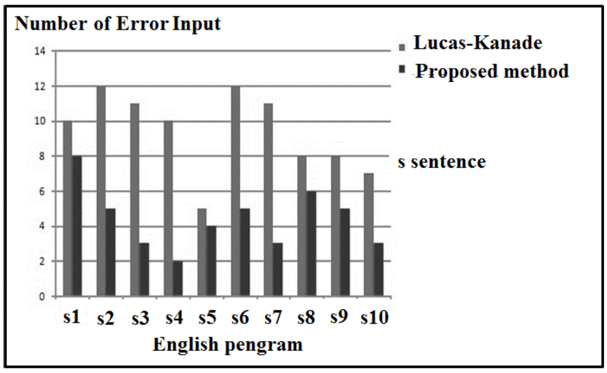

Fig. 14 Comparison of error input numbers

The proposed method yielded higher accuracy than Lucas-Kanade in the implemented smart device. Most of the input errors of the template-based tracking system arose due to the failure to recognize the instant gestures of the finger tips.

VII. CONCLUSION

This paper proposed a space touch-based gesture recognition algorithm which is an improvement on conventional methods; it was implemented using the SmartUX-MobileKII system for testing. In addition, the superiority of the proposed method to the template tracker used in the space touch of Tokyo University in terms of speed and accuracy was demonstrated through performance comparison. In a future study, a depth camera will be installed in an actual mobile device for gesture recognition using 3D hand models.

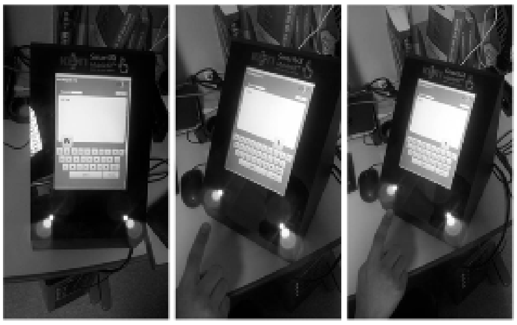

Fig. 15 SmartUX-MobileKI system

ACKNOWLEDGMENT

This research was funded by the MSIP (Ministry of Science, ICT & Future Planning), Korea in the ICT R&D Program 2013.

REFERENCES

[1] 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 2009, pp. 7-8.
[2] Y. Takeoka et al.: Z-touch: an infrastructure for 3d gesture interaction in the proximity of tabletop surfaces, Proceedings of ITS’10, 2010.
[3] Y. Tsukada et al.: Layerd touch panel: the input device with touch layers, Proceedings of CHI’02, 2002, pp. 584-585.
[4] Intel Corporation. Open Source Computer Vision Library reference manual. December 2000.
[5] T. Lee and T. Höllerer, "Handy AR: Markerless inspection of augmented reality objects using fingertip tracking," International Symposium on Wearable Computers, Citeseer, 2007, pp. 83-90.
[6] J. L. Rodgers and W. A. Nicewander, "Thirteen ways to look at the correlation coefficient," American Statistician 42, 1988, pp. 59-66.
[7] Baker, S. and Matthews, I. Lucas-Kanade 20 Years On: A Unifying Framework, International Journal of Computer Vision, 2004, vol.56, No3, pp. 221-255.
[8] "List of Pangrams", http://en.wikipedia.org/wiki/Li st_of_pangrams.