Hand Gesture Recognition in Multi-space of 2D/3D

Hansaem Lee†, Junseok Park††,

Department of Computer Software, University of Science & Technology, Daejeon, Republic of Korea Department
Wearable SW Research, ETRI, Daejeon, Republic of Korea

Summary
Recent human-computer interaction studies focus on natural interface based on hand/finger movements by tracking body motion using various sensors. In this paper, we propose 'Surface Gesture' recognition method using both 2D surface information and 3D spatial information. 'Surface Gesture' is composed of continuous gestures using 2D and 3D coordinates in multi spaces. We use a touchscreen to obtain the 2D coordinate and depth sensors to obtain finger coordinates in 3D space. We decompose 'Surface Gesture' into 3 particular steps and recognize 'Surface Gesture' with tagging a label on each step. By combining accurate 2D plane information and natural 3D space gestures, our method provides a more convenient and familiar way to interact with computers. We conducted an experiment to evaluate performance of the recognition engine. Experimental results show that the proposed method provides a 94% recognition rate on particular step and an 82% recognition rate on 'Surface Gesture'

Key words:
Leap Motion, Multi-space of 2D/3D, Hand gesture recognition.

1. Introduction

Conversation between people are communicating through activities that based on experience, such as language or gestures. Communication between the computer and the person, using external input device to process the voice or way to interact with a hand gesture by using a variety of sensors are being actively investigated. And progress has been made in a form that mimics a natural hand movement.
In this paper we propose a method 'Surface Gesture' which provides convenience and freedom for gesture operation. 'Surface Gesture' is composed of continuous sub-gestures using 2D and 3D coordinates in multi spaces. To recognize 'Surface Gesture', with results of sub-gestures. The gesture is to accurately touch targets using a 2D touch screen. And it ensures a natural interaction with the 3D sensor. This method can be useful when moving the object on the screen or the interactive interface with the different screens.
Existing methods only use 2D information system, hide the screen while typing gesture with hand. So the computer cannot know exactly what situation happened. This problem can be compensated by extending the path of the gesture 2D to 3D.

The paper is organized as follows. In Chapter 2 introduces the latest research trends, Chapter 3 describes the configuration of the system. In chapter 4 describes the experiments of the proposed method, we present the conclusions in Chapter 5.

2. Related Works

Method of finger movement recognition is using sensor row data and image data. One way is by using image data "3 Gear Devkit" [1]. This two-handed 6-DOF tracking system [2] supports the pre-computed database for high recognition rate. Although its high accuracy using the commercial cameras is merit, image processing is demerit.
Typical commercial products of this method using the sensor has a Leap Motion [4]. As this sensor setup in front of monitor, it tracks position of finger in the space of approximately 8 cubic feet of the top device. The device has a measurable accuracy [5-6] to 0.01mm. Since the sensor data is provided in forms of the coordinate values consisting of the actual distance, the image processing is not necessary, unlike the image data, which has the advantage of a reduced amount of operations.
Studies also have been made using a number of space. Autodesk announces HybridSpace [7] in the 2014 IEEE Symposium which proposed work area using both the 3D and 2D. When 3D stereoscopic tools that represents only a 2D plane can easily cause eye fatigue and difficulty to work. But this system proposed the modelling environment that solve both efficiency and stress.
Differences from the various studies mentioned above are compared to other studies using one kind of sensor, our method has a point that use two sensors. The use of 2D and 3D sensor simultaneously, complements the falling accuracy of the 3D sensor by using a 2D sensor, and provides an environment that can be used free gesture by using the 3D sensor. The use of two types of sensor provides effective gesture recognition environment.
In this paper, we propose a recognition method using a 2D, 3D space to ensure the success rate of the gesture recognition. To gather 2d data use touch screen that available multi-point recognize.

Manuscript received June 5, 2015
Manuscript revised June 20, 2015

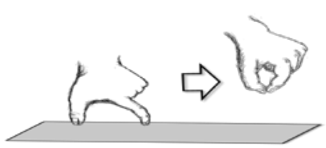
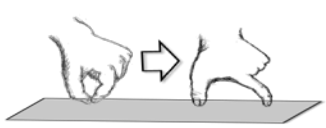
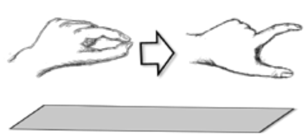
3. 'Surface Gesture' recognition system

In this chapter, we described a system for the surface gesture recognition.

3.1 Gesture

Surface gesture is one for use with a 2D plane and the 3D space. The advantage of this gesture, the finger tracking in 2D and 3D, compensate the low freedom of 2D and low accuracy of 3D. It allows a natural interaction between humans and computers by taking advantage of the degrees of freedom and accuracy of each.
Our goal is to recognize gestures pinch (Pinch-in) gestures, release (Pinch-out) gesture. These gestures are the details of the 'Surface Gesture'. Pinch gesture is performed in 2D space and 2D-3D (mixed-space pinch) is used. Release gesture is performed in 2D space and 3D is used. The mixed-pinch gesture will start in 2D, but the end is in 3D. (Table 1)

Table 1 Sub-gestures

	Description
2D pinch	
mixed-space pinch	
2D release	
3D release	

'Surface Gesture' recognition of the destination has four steps <Table 2>. The gesture is made in two of the detail gestures and one of free-moving motion. And each step can be divided clearly.

Table 2 Types of 'Surface Gesture'

	Start motion	***Mid motion***	***End motion***
2D to 2D	pinch (2D)	moving (3D)	Release (2D)
2D to 3D	pinch (2D)	moving (3D)	Release (3D)
3D to 2D	pinch (mixed-space)	moving (3D)	Release (2D)
3D to 3D	pinch (mixed-space)	moving (3D)	Release (3D)

3.2 Flag

Special point of the gesture recognizer, 'Surface Gesture' is divided into a series of sub-gestures. So recognize each sub-gesture, which determine 'Surface Gesture' based on them. For this purpose, we propose distinguish method using the Flag <Table 3>. The flag have true or false value, it retains a number of flag as a number of sub-gesture. A flag should turned into true value to recognize the sub-gesture. In 'Surface Gesture' recognition stage, the flags of sub-gesture are used to check what kind of 'Surface Gesture'.

Table 3 Gesture distinction using Flag

Flag	2D pinch	Mixed space pinch	2D release	3D release	result
State	○	×	○	×	2D to 2D
	○	×	×	○	2D to 3D
	×	○	○	×	3D to 2D
	×	○	×	○	3D to 3D

3.3 Algorithm

To recognize 'Surface Gesture' use following steps. First, recognize sub-gesture (pinch, release). And due to the nature of 'Surface Gesture', release motion is end of that gesture. So release motion detected, recognize surface gesture with flag setting that shows in 3.2.
This paragraph shows detail gesture recognition method in 2D and 3D.

3.3.1. Sub-Gesture Recognition

As previously mentioned 'Surface Gesture' is the set of a number of sub-gesture continuously made, it should be able to recognize the complete gesture. The complete gesture can be divided into the release and pinch, and pinch motion is divided in a 2D plane and mixed space. Release motion is divided 2D and 3D.

Touch point must be in around of interactive object. …… ①
The distance of each finger moved to be less than a threshold value …………… ②

Fig. 1 condition of 2D Pinch

Conditions required for 2D pinch is made (Fig. 1) as follows. The first touch point must be in around of an interactive object, and the second would be enough to move the pinch away for motion being performed. To verify the above two conditions the following process is required.

The finger data is needed the coordinates of each finger first reached touch screen (finger_1x_in, finger_1y_in), (finger_2x_in, finger_2y_in), … , (finger_nx_in, finger_ny_in),

The coordinates of the finger on the screen is turned off completely (finger_1x_in, finger_1y_in), (finger_2x_in, finger_2y_in) …, (finger_nx_in, finger_ny_in). To obtain the maximum value, the minimum value is as follows. (1). (n = touched finger number)

$$\begin{aligned} \max_x &= \max(\text{finger\_1}_{x_{in}}, \text{finger\_1}_{x_{out}}, \dots, \\ &\quad \text{finger\_n}_{x_{in}}, \text{finger\_n}_{x_{out}}) \\ \max_y &= \max(\text{finger\_1}_{y_{in}}, \text{finger\_1}_{y}, \dots, \\ &\quad \text{finger\_n}_{y_{in}}, \text{finger\_n}_{y_{out}}) \\ \min_x &= \min(\text{finger\_1}_{x_{in}}, \text{finger\_1}_{x_{out}}, \dots, \quad (1) \\ &\quad \text{fingern}_{\_x_{in}}, \text{finger\_n}_{x_{out}}) \\ \min_y &= \min(\text{finger\_1}_{y_{in}}, \text{finger\_1}_{y_{out}}, \dots, \\ &\quad \text{finger\_n}_{y_{in}}, \text{finger\_n}_{y_{out}}) \end{aligned}$$

At this time, when the center point of the object referred to the interaction point (Objectx,Objecty)

$$\begin{aligned} min_x &\le Object_x \le max_x \\ min_y &\le Object_y \le max_y \quad (2) \end{aligned}$$

When the formula (2) satisfied, the condition ① is established.

To satisfy the condition ②, the value of moving distance of finger should over threshold. To calculate the value of moving distance uses the Euclidean method (3)

$$distance_{fingern} = \sqrt{finger\_n_{x_{in}} + finger\_n_{x_{out}}} \quad (3)$$

The sum of the finger distance is greater than or equal to the threshold value if it is determined that pinch has succeeded (3-4).

$$distance_{finger\_1} + \cdots + distance_{finger\_n} \le Threshold \quad (4)$$

Conditions required for mixed space pinch is made (Fig 2) is as follows.

In the condition ① in Figure 5 established by formula 1 and 2. Assume that the position coordinate of the current finger (finger_1x_in, finger1y_in, finger_1z_in), (finger_2x_in, finger_2y_in, finger_2z_in), distance between the fingers is less than or equal to the threshold value when the operation of the pinch in the mixed space is determined.

The position of the finger occurs around the object, but did not established the pinch motion, ………①
If the distance between the finger on the 3D falls below a threshold value ………②
If the 3D sensor Pinch Strength value will be at least 0.9 ………③
If the sum of the angle formed by the two fingers drops below the threshold value ……④

Fig. 2 Condition of mixed space Pinch

$$distance_{finger\_n} = \sqrt{\begin{aligned} &(finger\_n_{x_{in}} - finger\_n + 1_{x_{in}})^2 + \\ &(finger\_n_{y_{in}} - finger\_n + 1_{y_{in}})^2 + \\ &(finger\_n_{z_{in}} - finger\_n + 1_{z_{in}})^2 \end{aligned}} \quad (5)$$

$$distance_{finger} \le Threshold$$

Pinch Strength value that used in the condition ③ is raw data of Leap Motion. This value measure distance of thumb and index finger, and provide a value between 0 and 1. 0 is totally closed state, 1 means to be fully opened. If the sensor send more than reference value, recognizer determine mixed space pinch operate.

The condition ④ confirms the two fingers used in the variation of the angle between a 2D plane. Change an angle of the fingers is provided by the 3D sensor, when change amount is performed higher than the threshold determine pinch is performed.

Recognition method of release and pinch is same.

Pinch gesture occurred …………… ①

When a touch is made to move the distance of each finger more than threshold value ……②

Fig. 3 Condition of Release motion in 2D

2D release motion is same with 2D pinch, so using formula 3, 4.

$$\sum(Distance) \geq Threshold_{distance}$$
$$\sum(angle) \geq Threshold_{angle} \quad (6)$$

Pinch gesture occurred ……①

The distance between the finger on the 3D is more than threshold, and the angle of difference is more than threshold …… ②

The distance between the finger on the 3D is more than threshold, and the pinch strength is more than threshold …… ③

Fig. 4 Condition of Release motion in 3D

The condition ③ is the same method for determining the condition ②, use pinch strength value instead of the angle variation.

$$\mathrm{diff}_{\mathrm{distance}} = distance_{current} - distance_{before}$$
$$\mathrm{Distance} = \mathrm{diff}_{\mathrm{distance}_1}, diff_{distance_2}, \ldots, diff_{distance_{15}}$$
$$\sum(Distance) \geq Threshold_{distance} \quad (7)$$
$$\mathrm{Pinchstrength} \geq \mathrm{Threshold}_{\mathrm{pinchstrength}}$$

3.3.2. ‘Surface Gesture’ recognition

Surface gesture is made up of a number of sub- gestures. It will recognize the gesture according to their characteristics. Gesture recognition method has been used a binary search tree using a “flag settings” and “gesture distinction” in the table 3.

4. Experimental

Experiments to confirm the usefulness of our method in Chapter 3 was performed. Subjects were targeted to the 20 to 30 adults, were carried out experiments with a total of eight people. To check experience of 3D hand gesture recognizer and 2D touch screen in the past, experienced and non-experienced person using a 1: 1 ratio were recruited to fulfill.
The system environment was developed using MFC (Microsoft Foundation Class) libraries in Windows 7, and 2D sensor using the Windows default event handler, SDK in Leap Motion.
Test method was used as the four surfaces gesture <Table 5>.

Table 4 'Surface Gesture' used in the experiment

	2finger			
Kind gesture	2D to 2D	2D to 3D	3D to 2D	3D to 3D

All four gesture were conducted in a different order each subject. Subjects are repeat that cycle two times. Surface gesture can be divided into free-movement phase and two sub-gesture phase. The result of recognition are <Table 5>, <Table 6> as follows.

Table 5 Surface Gesture recognition rate

	Recognition rate
2D → 2D	90.00%
2D → 3D	82.86%
3D → 2D	78.14%
3D → 3D	77.43%
average	82.11%

Table 6 Detail gesture recognition rate

	Recognition rate
2D Pinch	99.28%
3D Pinch	94.28%
2D Release	98.43%
3D Release	86.43%
Average	94.60%

According to the experimental results, in the case of the " sub-gesture, and showed a high recognition success rate in <Table 6>, surface gesture showed low recognition rate <Table 5>. Because ‘Surface Gesture’ is consist of ‘sub-gestures’, each ‘sub-Gesture’ recognition fails, get together in ‘Surface Gesture’. The cause of the failure, as the cause of the error, and how to access the touch screen can’t depend on user’s posture. So it may lead to input errors. The second cause is that the validation of a leap Motion data is lacking. Leap Motion has problem that difficult to sensing nearby monitor and affected by environment.

5. Conclusion

In this paper, we proposed recognition method for recognizing a ‘Surface gestures’. And algorithms for recognition of sub-gestures. Also we proposed a flag based recognition method for recognizing a ‘Surface gestures’.
Flag-based recognition method provides a way to recognize specific ‘Surface Gesture’, which consists of a detailed step-by-step operation. We design ‘sub-Gesture’ algorithm presenting a number of conditions, even when the error on one of the conditions was designed to be recognized in other conditions. Threshold values for recognition were set by the user experiments. Sub-Gesture’ shows 94.6%," ‘Surface Gesture’ shows 82.1% recognition rate. It shows this method is effective in surface gesture recognition.
Recognition rate of the "Sub-Gesture’ a higher, but ‘Surface Gesture’ are required to improve the recognition rate to be used in a variety of future applications. Because of characteristics of flag-based recognition method, ‘Surface Gesture’ recognition method is no problem. But if recognition rate of ‘Sub-Gesture’ is going to down, the recognition rate of ‘Surface Gesture’ ' decrease a lot. This Leap Motion sensor is providing accurate data when compared with the current commercial sensor, but it has limit, and data correction cannot be achieved sufficiently. To complement for this, we performed to validate the data and modify the threshold value based on the ambient conditions.

References

[1] 3 Gear Systems, ‘3 Gear Devkit’, http://www. threegear.com/about.html

[2] Robert Wang, Sylvain Paris, and Jovan Popović (2011), ‘6D hands: markerless hand-tracking for computer aided design’, ACM User Interface Software and Technology, 549-558

[3] Philip Cohen, et al. (1999), ‘Multimodal interaction for 2D and 3D environments’, IEEE Computer Graphics and Applications1999 vol.19,10-13

[4] Hal Hodson (2013), ‘Leap Motion hacks show potential of new gesture tech’, New Scientist, http://www.newscientist.com/article/mg21829116.100-leap-motion-hacks-show-what-3d-gesture-sensing-can-do.html

[5] Frank Weichert, et al. (2013), ‘Analysis of the Accuracy and Robustness of the Leap Motion Controller’, Sensors Volume 13, Issue 5

[6] Jože Guna et al. , ‘An Analysis of the Precision and Reliability of the Leap Motion Sensor and Its Suitability for Static and Dynamic Tracking’, Sensors Volume 14, Issue 2

[7] Bogdan, N. , Grossman T. , Fitzmaurice G. , (2014) ‘HybridSpace: Integrating 3D freehand input and stereo viewing into traditional desktop applications, 3D User Interfaces (3DUI), 2014 IEEE Symposium