International Journal of Electrical, Electronic and Communication Sciences
ISSN: 2517-9438
Vol:3, No:11, 2009

Virtual Gesture Screen System Based on 3D Visual Information and Multi-Layer Perceptron

*Yang-Keun Ahn, **Min-Wook Kim, *Young-Choong Park, *Kwang-Soon Choi,
*Woo-Chool Park, *Hae-Moon Seo and *Kwang-Mo Jung
*Korea Electronics Technology Institute,
#1599, Sangam-dong, Mapo-gu, Seoul
**Chonnam University
77, Yongbong-Dong, Buk-Gu, Gwangju 500-757
Korea
ykahn@keti.re.kr

***Abstract*—** Active research is underway on virtual touch screens that complement the physical limitations of conventional touch screens. This paper discusses a virtual touch screen that uses a multi-layer perceptron to recognize and control three-dimensional (3D) depth information from a time of flight (TOF) camera. This system extracts an object's area from the image input and compares it with the trajectory of the object, which is learned in advance, to recognize gestures. The system enables the maneuvering of content in virtual space by utilizing human actions.

***Keywords*—** Gesture Recognition, Depth Sensor, Virtual Touch Screen

I. Introduction

THIS screens are being widely used at present owing to the ease of intuitive control. The touch screen, however, cannot be controlled if the production of a touch sensor is impossible (e.g. for large-sized screens), or if direct contact cannot be made in cases where the screen is located far away. To address these problems and facilitate control of touch screens, wide-ranging research has recently been carried out on virtual touch screens [1-2].

Without any physical contact surface in place, a virtual touch screen generates a virtual screen at a given distance from the camera and recognizes the virtual touch of a physical object on the screen.

Kim Hyung-joon [1] and other researchers have proposed a dual camera-based virtual touch screen, which derives the three-dimensional (3D) position of a hand from images inputted from two fixed cameras and recognizes the touch point. Martin Tosas and Bai Li [2] have implemented a virtual screen using a single webcam and a physical grid; when a hand is situated within the framework of the physical grid, as opposed to a virtual grid, a single webcam tracks the hand and recognizes a touch.

Kim's research, however, implements a touch screen based on mathematical calculation of positioning, and as such it is subject to changes in the screen's position, as the screen is fixed. Tosas and Li fail to materialize a virtual screen, as 3D information is not included. Also, both studies implement limited touch features from among widely-varying human actions.

A time of flight (TOF) technology-based camera provides 3D depth image information. The results can be expressed in black-and-white images; a virtual touch screen can be implemented in a simple and efficient manner by employing an image processing technique.

A virtual touch screen, however, fails to take into account various human actions, as it simply realizes touch features based on touch points. Against this backdrop, this paper proposes a virtual screen that applies multi-layer perceptron technology to a virtual touch screen to recognize gestures.

The structure of this paper is as follows: Chapter 2 explains a TOF camera-based virtual touch screen, and Chapter 3 discusses a virtual gesture screen based on the multi-layer perceptron. Test results are presented in Chapter 4; a conclusion and suggested directions for future research are given in Chapter 5.

II. TOF Camera-Based Virtual Touch Screen

A virtual screen can be easily implemented by utilizing the 3D depth information of a TOF camera and an image processing technique.

A. Depth image

A TOF camera emits a laser or LED light. Using a built-in sensor, the time taken for the light molecules to return to the camera after touching the object are then recognized, and the distance from that object is calculated [4-5]. The outputs of this camera are 3D depth images, which represent approximate 3D information.

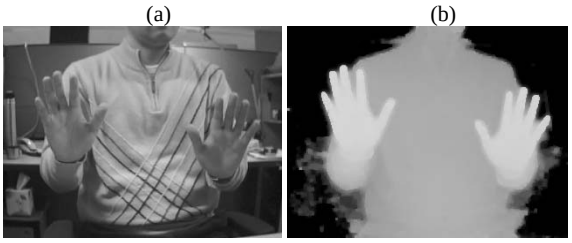

Fig. 1: (a) Image input (b) Depth image

In Fig. 1, (a) refers to the image input of the TOF camera and (b) the corresponding depth map. In (b), the depth map becomes whiter when the object is closer to the camera and blacker when it is farther away.

B. Setting of a threshold value

A virtual screen can be created by setting a given threshold value for the depth map.

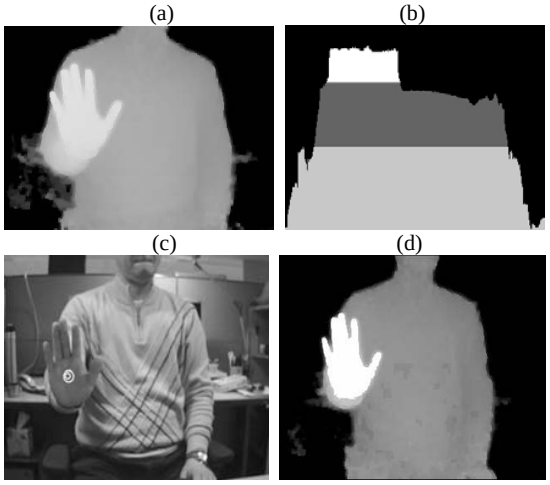

Fig. 2: (a) Depth image (b) X-depth image
(c) Object area tracking (d) Virtual screen image

In Fig. 2, (a) is a 3D depth image and (b) an X-depth image, where the depth image is projected onto the X-axis. This X-depth image represents a bird's eye view on the camera. By setting a certain threshold value for the X-depth image, the area of a hand entering the virtual screen can be extracted, as shown in Fig. 2(d). Fig. 2(c) represents an image used in calculating the central moment of the hand area and tracking the hand area that reaches the virtual touch screen. This technique enables the implementation of a virtual screen.

III. Virtual Gesture Screen

A virtual gesture screen can be realized by incorporating an object trajectory database extraction system and a gesture recognition system into the implemented virtual touch screen.

A. Object trajectory database extraction system

An object trajectory database extraction system consists of gesture input, gesture image correction, and feature extraction units.

1) Gesture input

Using a virtual touch screen, touched points are saved to create a gesture database. Whenever a touch action is performed, calculation is done at each frame to determine if the respective frames are touched. If touches are performed on a continued basis, the touched points are saved in the memory, and the gesture is assumed to have ended if an untouched frame is found. Then, as described in Fig. 3, the points are connected to create an image.

Fig. 3: Gesture input images

2) Gesture image correction

When the gesture input is over, several phases should be carried out to correct the gesture image.

On this image, even the same gestures may take place in different positions and in differing sizes, and thus individual gestures need to be generalized. In other words, gesture recognition data regardless of position and size are required.

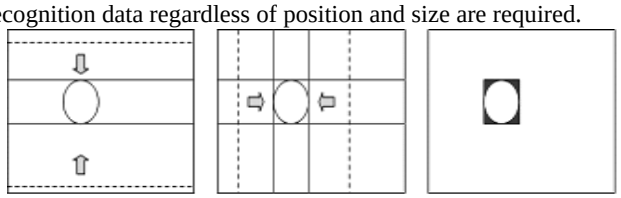

Fig. 4: Positioning of endpoints at the edge of a gesture

For this purpose, the endpoints on the four sides of each gesture image are identified—as shown in Fig. 4—and the image of the area, as described in Fig. 5, is projected onto a 100×100 image. In other words, any image that is smaller than 100 pixels in width or length is enlarged, and one that has width or length greater than 100 pixels is reduced in size.

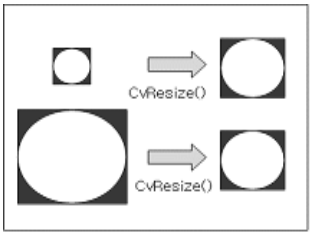

Fig. 5: Readjustment of gesture image size

3) Feature extraction

After obtaining a gesture image with a size of 100×100

pixels, the features to be used as the input unit in the perceptron should be extracted from the image. This paper divides an image of certain size into 25 smaller images with a size of 20×20 pixels and, as illustrated in Fig. 6, the number of pixels in the gesture part of respective areas is taken as their features. Also, the target value of the perceptron is added to the end of the database. If the total number of gestures is 3 and the inputted gesture is Gesture #1, the figure "1 0 0" is entered; if the inputted gesture is Gesture #2, "0 1 0" is added instead. This is because a sigmoid function is used during the learning process as an active function.

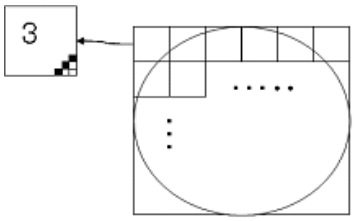

Fig. 6: Number of pixels in the gesture part

B. Gesture recognition system

The gesture recognition system is implemented based on multi-layer perceptron technology.

1) Multi-layer perceptron

Multi-layer perceptron (MLP) [6-7] is the representative algorithm for supervised learning. The algorithm brings the differences between output values obtained from inputs and predetermined targets of supervised learning (i.e. deviation) back to the perceptron structure and redistributes them in order to gradually reduce deviation levels during the learning process.

2) Structure of perceptron

The multi-layer perceptron and data from the database explained in Chapter 2 are utilized to train the perceptron. In this case, 25 feature values, specified in 3.1.3, are used as inputs, and the learning target is set at the target value for the database; the number of hidden layers, learning rate, momentum, and target error rate are also determined in advance. Fig. 7 illustrates the structure of the multi-layer perceptron.

3) Gesture distinction

After training the multi-layer perceptron, we worked on creating a gesture distinguisher. A real-time trajectory database program is run, and the data produced here are placed into the trained perceptron; its outputs are then used to distinguish gestures. The index of the node with the greatest nodal results for the output layer is used to distinguish a gesture from others. In other words, if the first node of the output layer is greater than the values of the other two nodes, the gesture may be concluded to be Gesture #1.

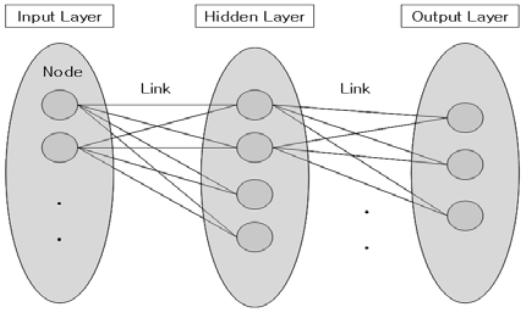

Fig. 7: Structure of multi-layer perceptron

IV. Test Results

To analyze the performance of this system, an experiment is performed to control a flash application with the three gestures described in Fig. 3. The number of hidden layers for the multi-layer perceptron is set at 10, the learning rate at 0.1, and the momentum at 0.8. The target error rate is set to be 0.05 before initiating the learning process; five databases per gesture are used for the experiment.

The experiment shows that the three gestures are different from each other and share no common aspects, and consequently less than 100 iterations of repetitive learning is sufficient to successfully distinguish the gestures.

The flash program provides a menu selection feature when the user makes the first gesture of drawing a circle. The menu is turned to the right when the user draws a right arrow and to the left when the user makes the gesture of drawing a left arrow. Fig. 8 demonstrates how the flash application is controlled using the virtual gesture screen.

Fig. 8: Controlling of flash program on virtual gesture screen

V. Conclusion

This paper has applied an object trajectory database extraction system and a multi-layer perceptron technology-based gesture recognizer to the conventional virtual screen system to enable gesture recognition as well as virtual touching.

This makes it possible to express wide-ranging human actions, which can be expressed only in a limited manner through virtual touches in 3D space, and to develop a wide variety of content on this basis.

The proposed system can recognize gestures on the virtual screen, but it identifies a single object on a real-time image and is applied only to a single gesture made by that object. The applicability of the system will be further enhanced if a multi-gesture recognition program is developed in the future that enables the recognition of gestures from multiple objects.

ACKNOWLEDGEMENTS

This research was supported by Ministry of Knowledge Economy(MKE), Korea as a project, " The next generation core technology for Intelligent Information and electronics".

REFERENCES

[1] Kim Hyung-joon, “ Virtual Touch Screen System for Game Applications”, *Journal of Korea Game Society*, vol. 6, no. 3, pp77-86, 2006.
[2] Martin Tosas and Bai Li, *Lecture Notes in Computer Science*, Heidelberg, Springer Berlin, pp48-59, 2004.
[3] Eunjin Koh, Jongho Won, and Changseok Bae,"Vision-based Virtual Touch Screen Interface", *Proceeding of ICCE 2008*, LasVegas, USA, 2008.
[4] http://en.wikipedia.org/wiki/Time-of-flight
[5] Gokturk. S. B, Yalcin. H, and Bamji. C., "A Time-Of-Flight Depth Sensor - System Description, Issues and Solutions", CVPRW '04, p35, 2004.
[6] P. Hajela, B. Fu and L. Berke, Neural networks in structural analysis and design: an overview. Comput. Syst. Engng 3 1-4 (1992), pp. 525-538.
[7] R.P. Lippmann, An introduction to computing with neural nets. IEEE Acoust. Speech Signal Process 4 2 (1987), pp. 4-22.