Implementation of 3D Virtual Touch Screen

*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: - This study implements a virtual touch screen that keeps track of the location of hand, inputted from the disparity image being outputted by a time of flight (TOF) camera, using the Kalman filter. Put out from the depth image of the TOF camera, this image is insensitive to light and therefore helps implement a virtual touch screen independent from the surroundings. The biggest problem with conventional virtual touch screens has been that even the slightest change in location led an object to fall out of or enter the virtual touch screen. In other words, the pointing location responded too sensitively to accurately detect the touch point. The Kalman filter, on the other hand, can be a solution to this problem, as it constantly predicts the pointing location and detects this touch point, without interruption, in response to even a slight change in location. This enables a stable and smooth change in the location of pointing point on the virtual touch screen.

Key-Words: - Gesture Recognition, Natural Interaction, Depth Sensor, Virtual Touch Screen

1 Introduction

A technology combining display and input device, touch screen is being widely used in various areas. A touch screen, however, is reliant upon a physical membrane and has a size limit; technically speaking, it is vulnerable to vibrations and its maximum optical transmittance stands only at 86%. Against this backdrop, there has been a growing need for the development of technology that can complement these weaknesses of touch screen [1], and virtual touch screen will be the right option for this purpose.

Kim Hyung-joon [1] proposes the use of two cameras for implementing a virtual touch screen. Using two fixed cameras, a virtual touch screen is built based on mathematical locational calculation, not stereo algorithm; when a certain object (i.e. hand) enters the screen, this is recognized as a touch. Martin Tosas and Bai Li [2] implemented a virtual touch screen using one webcam and a physical grid. When a hand enters the framework of physical grid, as opposed to virtual grid, one webcam tracks the hand, generating an event if a certain motion takes place. Eunjin Koh [3] formed a system structure, using a virtual touch screen—based on the AR marker and two stereo cameras—and HMD; a hand is found and tracked, and then the hand's movements are utilized to implement mouse-like features.

In the technology of Kim Hyung-joon, however, implementation of a touch screen solely based on mathematical locational calculation entails the problem of the screen being fixed mathematically and the touch point being hard to find even when the hand moves only slightly. With only one single webcam being applied, Martin Tosas' technology fails to process three-dimensional depth information; Eunjin Koh's is sensitive to light as stereo cameras are used.

This study, on the other hand, detects the touch point consecutively, using the Kalman filter, despite slight locational change. It also utilizes the TOF camera, where infrared light and infrared camera are used, to extract depth information from an image in a way that is not sensitive to the brightness of light and the surroundings. On this basis, this study suggests a technology for tracking the touch point using the Kalman filter and a technique for configuring a three-dimensional screen, based on depth information, which remains constant against any change in light.

2 Overview of Kalman Filter

Filtering has continuous nature: To estimate the state at the time of t requires the data up to t-1; once the data at t are obtained, the state at the time of t+1 is estimated on this basis. One sort of filtering technique, the Kalman filter has the characteristics of continuous optimality that seeks for the most optimal forecast at each time of estimation; forming a model in the time area, it is easy to deal with mathematically and is therefore being widely used.

The basic premise of Kalman filter is that observation values do have errors. The goal here is to process the observation values—with errors included—using the Kalman filter and renew the estimated values to make an optimal estimation.

The Kalman filter can be broadly divided into three parts: system equation, measurement equation and Kalman filtering. In the system equation part, state vector X(t), which indicates dynamic properties by time, to estimate the state of the system.

$$X(t) = \phi(t-1)X(t-1) + w(t-1) \quad (1)$$

In Equation (1), X(t) is the state vector at the time of t and X(t-1) the state vector at the time of t-1. ϕ(t-1) is a transition matrix that transits the state vector at t-1 into one at t. w(t-1) is a system error vector that obeys normal distribution with the mean of 0 and variance of Q.

As shown in Equation (2), the measurement equation represents the relation for deriving observation vector Z(t) from state vector X(t).

$$Z(t) = H(t)X(t) + v(t) \quad (2)$$

In Equation (2), Z(t) is an observation vector and H(t) an observation transition function matrix that transits state vector X(t) into the observation vector. v(t) is an observation error vector which follows normal distribution with the mean and variance of 0 and R, respectively.

System equation in the Kalman filter is the process of renewing the initially estimated values at the time of t using observation values.

$$\hat{x}(t|t) = K'(t)\hat{x}(t|t-1) + K(t)z(t|t-1) \quad (3)$$

In Equation (3), $\hat{x}(t|t)$ is an optimal estimated value at the time of t and $\hat{x}(t|t-1)$ is a state estimation value at the time of t-1, which is estimated using the data up to t-1. K'(t) and K(t) are time-varying weighted matrices enabling state estimation values at each point of time to have minimum variance. When K(t) is calculated and selected as in the below Equation (4):

$$K(t) = P(t|t-1)H^T(t)\left[H(t)P(t|t-1)H^T(t) + R(t)\right]^{-1} \quad (4)$$

Optimal estimation $\hat{x}(t|t)$ can be obtained, where P(t|t-1) is error covariance. Fig. 1 visualizes this process of renewal [4].

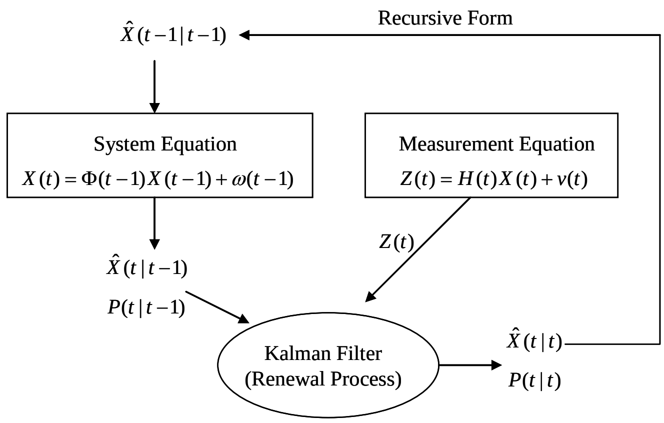

Fig. 1 renewal process of kalman filter [5]

3 TOF Camera

A time of flight (TOF) camera calculates the distance from an object by sending out laser or infrared LED and determining—with its built-in sensor—the time spent for laser/infrared LED particles to hit the object and come back [6]. If the laser or infrared LED that the camera emits forms a sine curve, the equation of the sine curve, s(t), can be written as Equation (5) when modulated frequency is f_m.

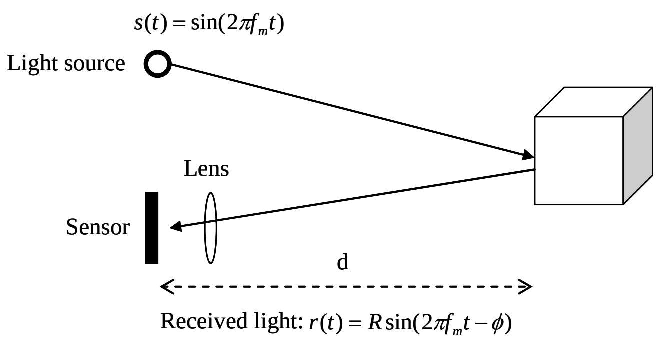

Fig. 2 how a TOF camera works [7]

$$s(t) = \sin(2\pi f_m t) \quad (5)$$

Then, r(t), which represents the light reflected by the camera and the object and transmitted into the built-in sensor, can be expressed as Equation (6).

$$r(t) = R\sin(2\pi f_m t - \phi) = R\sin(2\pi f_m (t - \frac{2d}{c})) \quad (6)$$

Here R is the magnitude of the reflected light and ϕ is the value of the light phase-changed from the object; c is the speed of light, that is, $3\times10^8\ m/s$; and d is the distance between the object and the camera, which can be obtained from the phase-changed value, and this can be written as Equation (7).

$$d = \frac{c\phi}{4\pi f_m} \quad (7)$$

Using the value of distance obtained above, the TOF camera determines the disparity value (i.e. distance). Fig. 3 illustrates a disparity map generated after identifying the distance.

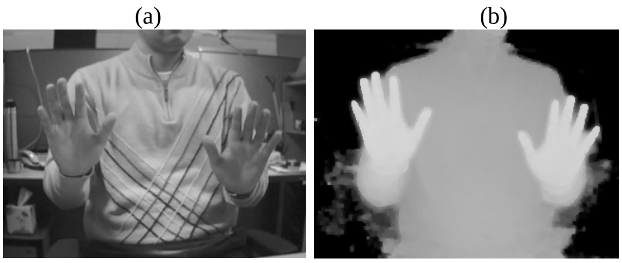

Fig. 3 (a) input image; (b) disparity image

4. Implementation of 3D Virtual Touch Screen

This study implements a virtual touch screen, tracking the touch point with the Kalman filter and using the TOF camera.

4.1 Determination of threshold value

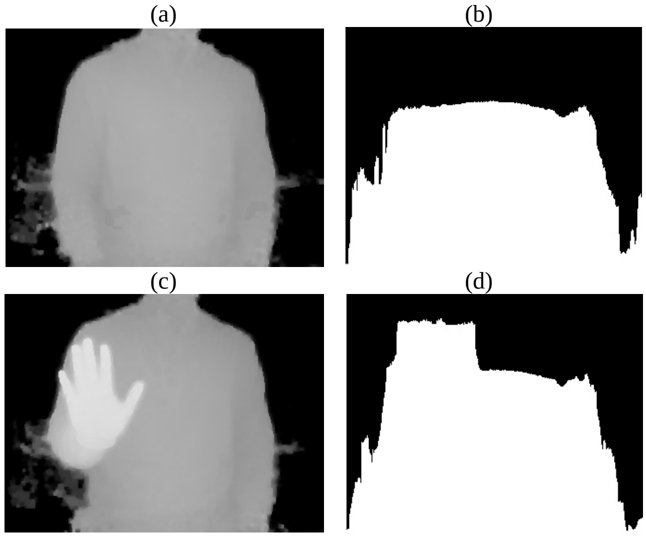

Fig. 4 depth and x-depth transformation images

The very first step to implement a virtual touch screen is to separate the hand's location from the inputted depth image of TOF camera by setting the threshold value on the image. Fig. 4 shows depth and x-depth images. In an x-depth image, the value of the x-axis on the image refers to the x-axis as in a real image and its y-axis stands for the biggest among the depth values of the y value. In other words, this represents the silhouette seen by the camera from above.

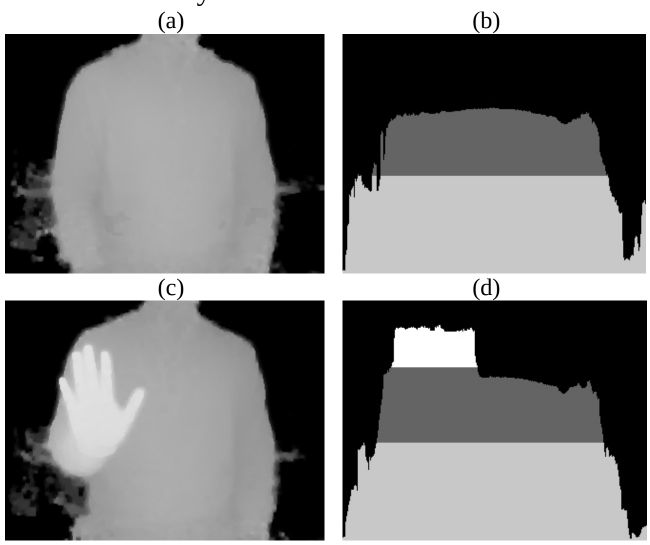

Fig. 5 x-depth image where threshold value is applied

In Fig. 4, (a) is the depth image when the actor does not keep his hand or hands forward, and (b) is the x-depth image for the same situation. What sticks out here represents the whole of the actor's body. Presented in (c) is the depth image of a hand brought forward. In comparison to (b), the hand, when seen from (d), sticks out more than the whole of the actor’s body. By applying a certain threshold value to this x-depth information, x-depth can be separated into hand, body and background. Fig. 5 visualizes this hand-body-background separation based on the x-depth image. In this figure, (a) and (c) are depth images; the upper part of (b) represents the body of the actor and the lower part the background. Shown in (d) are the results of detecting the hand areas other than the body and background of the actor by setting the threshold value.

4.2 Labeling

Once the hand areas are detected using the threshold value, what is needed now is to put together meaningful parts of those areas (i.e. labeling process). If only one hand is completely detected, this process would of course be unnecessary; when both hands are detected, labeling should be done to recognize each of them separately or eliminate the errors of these two hands joined together. For the labeled hand areas, the central moment value should be used to find out the central point again, which will then be used in updating observation vector Z(t).

Fig. 6 shows the central point of the extracted hand areas—expressed in the form of a circle—that are labeled and whose central moment values are calculated.

Fig. 6 central point after calculation of central moment

4.3 Tracking of touch point

After labeling and calculation of central moment, the Kalman filter is used to apply the central point obtained this way to observation vector Z(t) and thereby track the touch point.

State vector X(t) is set using the x- and y-axes and their respective amounts of change in the image labeled from the camera; observation vector Z(t) is determined using the values of the x- and y-axes observed from the image inputted real-time. Also, transition matrices (t-1) and H(t) are set as described in Equation (8).

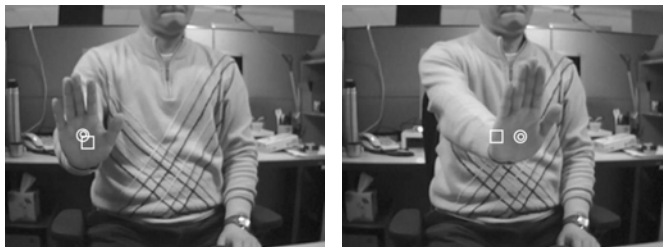

Fig. 7 tracking based on Kalman filter

$$\phi(t-1) = \begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, H(t) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \quad (8)$$

Shown in Fig. 7 is the image of touch points being tracked, using the Kalman filter where state and observation vectors are applied. The large circle represents the central point of a real hand, and the small circle is one measured by calculating the noise of the Kalman filter into the hand. The rectangular is the point forecast by the Kalman filter.

4.4 Dual touch screen

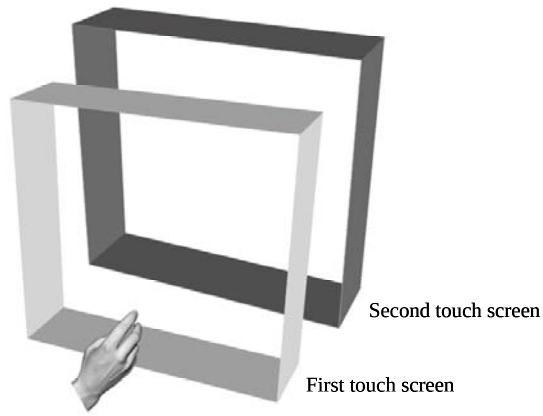

Fig. 8 structure of dual touch screen

A virtual touch screen is implemented using two threshold values, so it has the structure of dual touch screen. The primary virtual touch screen keeps track of the touched x and y values and corresponds them to mouse movement events so that the movements of the mouse can be controlled on the screen. The secondary screen is devised to recognize a touch, which sets off an event equivalent to a mouse click. Fig. 8 illustrates the input structure of a dual-structured virtual touch screen.

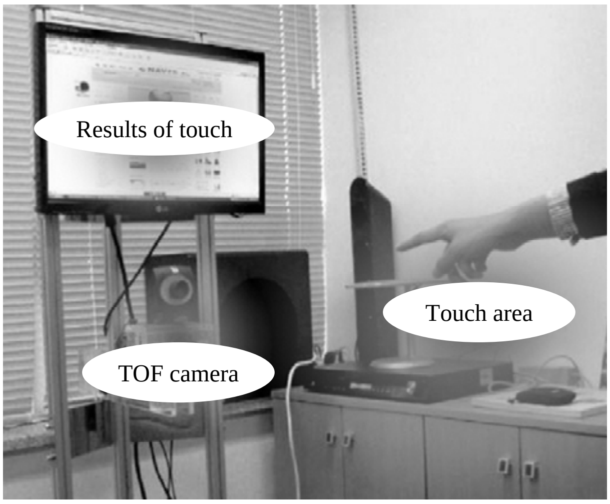

Fig. 9 actually implemented virtual touch screen

Fig. 9 is the picture of an actually implemented virtual touch screen, which has the actor area that receives gesture inputs from the actor; the camera area receiving inputs of what the actor does; and the screen part that runs the touch recognition process and shows the results of touch recognition.

5 Conclusion

This study has built a virtual touch screen system, using the touch point tracking technique based on the TOF camera and the Kalman filter. To address the widespread problem in conventional virtual touch screen systems (i.e. responding sensitively to even the slightest hand movement and failing to recognize the touch point accurately), this study has applied the Kalman filter tracking technique to predict and track the touch point consecutively. Being forecast and recognized, the predicted touch point is bound to exist all the time, but an abrupt movement of touch point is likely to be recognized only insensitively. In future studies, an algorithm that reacts sensitively to a sudden touch point movement needs to be applied to develop a virtual touch screen responding more spontaneously to gestures.

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] Chang Seok-kon, “ Combining Forecasts of Chunf-Ju Dam Daily Inflow Using Kalman Filter”, 2002.

[5] Eric Cuevas, Daniel Zaldivar, and Raul Rojas, "Kalman Filter for vision tracking", 2005.

[6] http://en.wikipedia.org/wiki/Time-of-flight

[7] Gokturk. S. B, Yalcin. H, and Bamji. C., "A Time-Of-Flight Depth Sensor - System Description, Issues and Solutions", *CVPRW '04*, p35, 2004.

[7] Kalman, R. E. (1960), A New Approach to Linear Filtering and Prediction Problems, *Journal of Basic Engineering*, Vol. 82, No. 1, pp. 35-45.