International Journal of Control and Automation
Vol. 11, No. 3 (2018), pp.67-78
http//dx.doi.org/10.14257/ijca.2018.11.3.07

Hand-Tracking Framework using Three-dimensional Tracking Ellipsoid

Seok-Han Lee

Department of Information and Communication Engineering,
Jeonju University, Jeollabuk-do, Korea
seokhan@jj.ac.kr

Abstract

In this paper, we present a hand-tracking system. Our system uses a depth-capture device to acquire the 3D geometry of a user's bare hand, such that the user's hand position and gesture information can be estimated from the depth cues. In particular, we build an ellipsoidal tracking object and restrict the hand tracking to within the ellipsoidal area to avoid the diverse problems occurring in conventional object detection/tracking systems. This tracking area is actively controlled according to the statistics of a user's hand gesture and motion history. The proposed system aims to be used in diverse systems that operate based on the natural user interface framework. The experimental results verify the accuracy and efficiency of the proposed system.

Keywords*: Hand Tracking, Hand Shape Representation, Human-Computer Interaction, Hand Gesture Recognition*

1. Introduction

Typically, user-computer interfaces are based on conventional interaction devices, such as a mouse, keyboard, touch screen, tablet, etc., all of which rely on a user's physical contact and operation. However, in many situations, such operations and interactions are impossible; in such a case, a user's natural body motion can be used as very useful information for the human-machine interaction. Therefore, considerable research efforts have been carried out on the techniques known as the natural user interface (NUI), in which the system attempts to track and recognize a user's action and this then provides a very efficient method for the interaction [1-7]. In particular, the NUI technique, which uses a user's bare hand information, is known as one of the most efficient methods that enables users to interact with any device. Although hand information tracking/estimation is steadily gaining importance owing to the drive from various applications, most conventional approaches are restrictive and difficult to use in real applications. In addition, they often require complicated and expensive devices that are generally operated under laboratory conditions by experts, and are not easily adopted in commercial applications. In this paper, we present a hand-tracking and gesture recognition system. Our system uses a depth-capture device (MS Kinect) to acquire 3D geometric information of a user's bare hand, and the user's hand position and gesture information are estimated from the depth clues. In particular, we build a flexible ellipsoidal object and restrict the hand-tracking area within the ellipsoidal object to avoid tracking failure problems, often occurring in conventional object detection/tracking systems. This tracking area is actively adjusted according to the statistical data of a user's hand motion history. The proposed system computes the running average of the hand position during a predefined period, and estimates uncertainty of the hand motion direction based on the covariance of the hand coordinated in the 3D space. This uncertainty is used to determine the shape and size of

Received (October 26, 2017), Review Result (January 22, 2018), Accepted (January 31, 2018)

ISSN: 2005-4297 IJCA

the ellipsoidal tracking object in the next tracking period. Subsequently, the hand position in the next tracking period is assumed to be present in the tracking object, and we detect the user's hand position in the predefined area. This approach ensures very stable tracking even when multiple objects are present in the scene simultaneously, which occurs very frequently in real applications. Once the position of a user's hand is acquired, the system attempts to count the number of fingers to recognize the gesture information of the hand. Our system tracks a user's hand information using the active tracking ellipsoid, such that it can be used in systems, such as a kiosk with a very large screen in which conventional interfaces cannot be used, in a crowded area. The proposed system aims to be adopted in diverse systems that operate based on the NUI framework. The experimental results verify the accuracy and efficiency of the proposed system.

2. Algorithm Description

The proposed system computes the running average of the center coordinate of the hand during a predefined period to remove the noise (jitter) problem. Currently, we accumulate the hand coordinates of 50 frames, and this running average is also used to compute the covariance of the hand motion, which determines the shape of the ellipsoidal tracking object. Once the position and uncertainty of a user's hand are estimated, the system then attempts to count the number of fingers to recognize a user's hand information. This allows the users to perform diverse interactions with the system using only a bare hand. Figure 1 shows the block diagram that illustrates the work flow of the proposed system.

2.1. Three-dimensional Tracking Ellipsoid

In our system, the hand tracking largely relies on the estimation of the tracking ellipsoid. Figure 2 shows the estimation of the ellipsoidal tracking ellipsoid, and the ellipsoid is used to restrict the detection area in the next tracking phase. Initially, the search range is set to a fixed sphere of a predefined radius. For the center position of the hand $\mathbf{X}^{\mathrm{t}} = (x_h,\ y_h,\ z_h)^{\mathrm{T}}$, the initial ellipsoid is given as

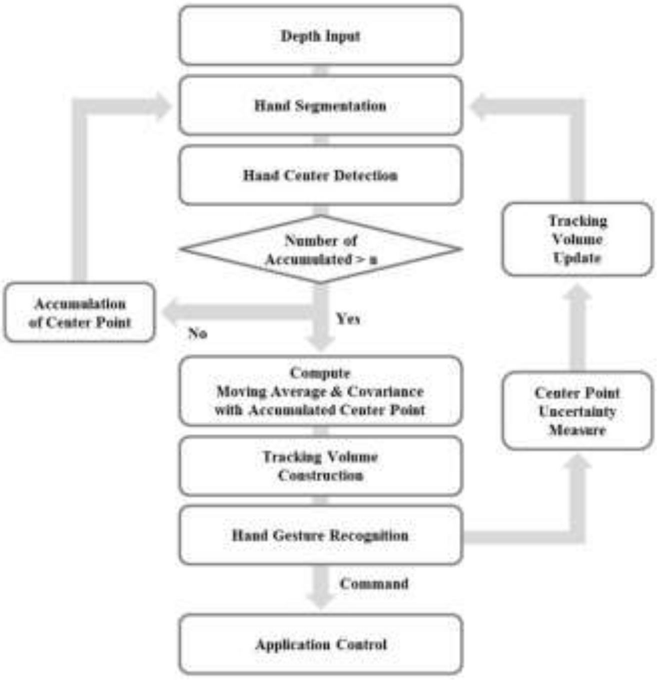

Depth Input
Hand Segmentation
Hand Center Detection
Number of Accumulated > n
Tracking Volume Update
Accumulation of Center Point
No
Yes
Compute Moving Average & Covariance with Accumulated Center Point
Center Point Uncertainty Measure
Tracking Volume Construction
Hand Gesture Recognition
Command
Application Control

Figure 1. Workflow of the Proposed System

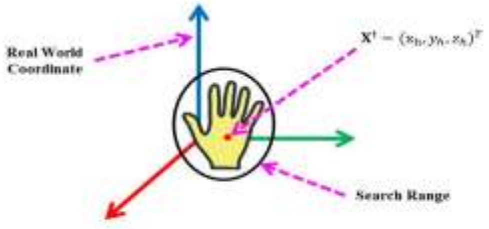


Figure 2. Search Range Restriction using the Ellipsoid

$$(x - x_h)^2 + (y - y_h)^2 + (z - z_h)^2 = r^2. \tag{1}$$

Considering the scale along each axis of the coordinate frame, the ellipsoid is represented as the following.

$$\frac{(x-x_h)^2}{\alpha^2} + \frac{(y-y_h)^2}{\beta^2} + \frac{(z-z_h)^2}{\gamma^2} = r^2, \tag{2}$$

where α, β, and γ are the scale factors determined by the covariance of the hand motion. The covariance for the uncertainty estimation is computed as follows. First, we compute the center of gravity of the hand region [8].

$$m_{00} = \sum_x \sum_y I(x, y)\, , m_{10} = \sum_x \sum_y xI(x, y)\, ,\ m_{01} = \sum_x \sum_y yI(x, y),$$

$$x_c = \frac{m_{10}}{m_{00}},\ y_c = \frac{m_{01}}{m_{00}}. \tag{3}$$

Here, $I(x, y)$ is the hand image, and $\mathbf{x}=(x_c, y_c)$ is the center point of the hand image. From x accumulated during a predefined period of n frames, the average of the hand position is computed as follows.

$$\mathbf{x}_a^t = \frac{1}{n}\sum_{i=0}^{n} \mathbf{x}^i. \tag{4}$$

Further, the running average is used to compute the covariance of the hand motion, acquired by Eq. (5). Here, we empirically set $n = 50$.

$$\mathbf{x}_a^t = \mathbf{x}_a^{t-1} + \frac{\mathbf{x}^t - \mathbf{x}^{t-1}}{n}. \tag{5}$$

From the running average and Eq. (4), we compute a 3 × 3 covariance matrix as shown in Eq. (6). We can assume that the motion vector of the hand motion is independent of each coordinate frame, and we may set all the off-diagonal elements of the covariance matrix to zeros. Once the covariance matrix is acquired, we compute the scale factors of the ellipsoid along each axis, $\alpha = \omega_x \sigma_{xx}$, $\beta = \omega_y \sigma_{yy}$, $\gamma = \omega_z \sigma_{zz}$, and they determine the shape of the ellipsoidal tracking volume. From the center position of hand $\mathbf{X}^{\mathrm{t}} = (x_h,\ y_h,\ z_h)^{\mathrm{T}}$ and the scale factors, we can build the active tracking volume, which is used to restrict the tracking area in the 3D space.

$$\mathbf{cov}_t = \frac{1}{\mathrm{n}}\sum_{i=t-n}^{t} (\mathrm{x}^t - \mathrm{x}_a^t)(\mathrm{x}^t - \mathrm{x}_a^t)^{\mathrm{T}} = \begin{bmatrix} \sigma_{xx} & \sigma_{xy} & \sigma_{xz} \\ \sigma_{xy} & \sigma_{yy} & \sigma_{yz} \\ \sigma_{xz} & \sigma_{yz} & \sigma_{zz} \end{bmatrix} \approx \begin{bmatrix} \sigma_{xx} & 0 & 0 \\ 0 & \sigma_{yy} & 0 \\ 0 & 0 & \sigma_{zz} \end{bmatrix}. \tag{6}$$

2.2. Hand Shape Representation

After the hand position is obtained, the system attempts to recognize a user's hand gesture by counting the number of fingers. Typically, conventional approaches for the finger counting often need specific conditions that ensure stable operations. For example, Ren's approach [5] requires a black belt to be worn on the user's wrist to detect the accurate hand region. However, this type of constraint is not always possible, and may be uncomfortable to users. Instead, we use an alternative approach. First, we obtain the line l_0 that is orthogonal to the direction vector of the hand motion (green line in Figure 3(a)) and passes through the weighted center of the hand region. Subsequently, at the time step t, we compute two points $\boldsymbol{P}^t= \{p_s, p_e\}$ where the line l_0 intersects with the contour of the imaged hand $\boldsymbol{C}^k = \{c^1, c^2,\ldots, c^k\}$, and use these two points (yellow points in Figure 3(b)) as the initial and final points for the outline extraction for the finger counting. Figure 3 shows the geometrical hand shape description and the time series curve that illustrate the extracted hand contour curve. Here, the x-axis is the angle from the initial point to the final intersection, and the y-axis represents the normalized distance from the hand's center point to the imaged contour. From the curves in Figure 3, the topological information of the imaged hand, which is used to recognize a user's hand gesture, can be obtained. In D. Lee's approach [9], the hand contour is extracted and ellipses are fitted to the imaged finger tips. Further, based on fitting the errors, the inliers of the imaged finger contour are counted and the number of fingers is estimated from the number of inliers. However, this approach is not suitable for noisy images, and the counting results might be quite unstable. Instead, we use topological attributes of the time series contour curves of Figure 3. From the extracted hand contour, we estimate the distance between the center point of the hand and the contour curve $\boldsymbol{C}^t = \{c^s, c^{s+1},\ldots, c^e\}$, which exists between the two intersection points $\boldsymbol{P}^t= \{p_s, p_e\}$. Further, the distance is normalized by the minimum distance between the center point and the contour curve. Subsequently, we compute the normalized angles between the vector $\overrightarrow{x^tC^t}$, which points from the center point $\mathbf{x}^t$ to one point on the contour curve, and $\overrightarrow{x^tp_s}$, which is from $\mathbf{x}^t$ to the first point of the intersection. This is represented in Eq. (7) and Eq. (8) as follows.

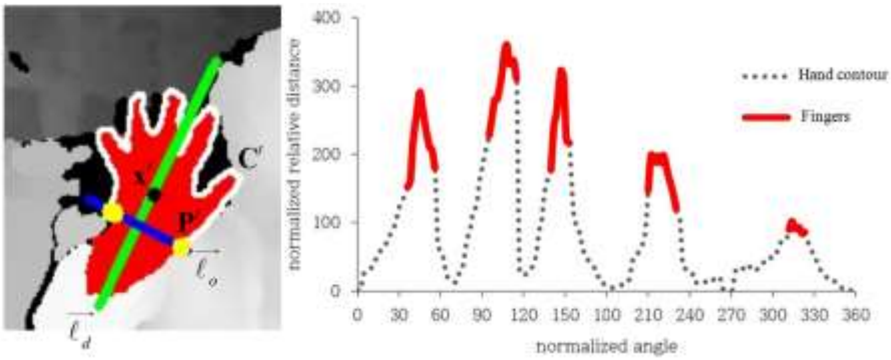
$$ND^k = \frac{\left|\overrightarrow{x^tc^t}\right|}{\min\left(\left|\overrightarrow{x^tc^s}\right|\right)}. \tag{7}$$

$$NA^k = \cos^{-1}\left(\frac{\overrightarrow{x^tp_s}\cdot\overrightarrow{x^tc^t}}{2\pi}\right). \tag{8}$$

Here, ND^k is the k-th normalized distance, and NA^k means the angle between $\overrightarrow{x^tC^t}$ and $\overrightarrow{x^tp_s}$, which is normalized by 2π. In Figure 3(b), the horizontal axis represents NA and the vertical axis ND. From the time series curve, we estimate the curvature data of the finger tips using Eq. (9). This can then be used to count the number of fingers to recognize a user's hand gesture.

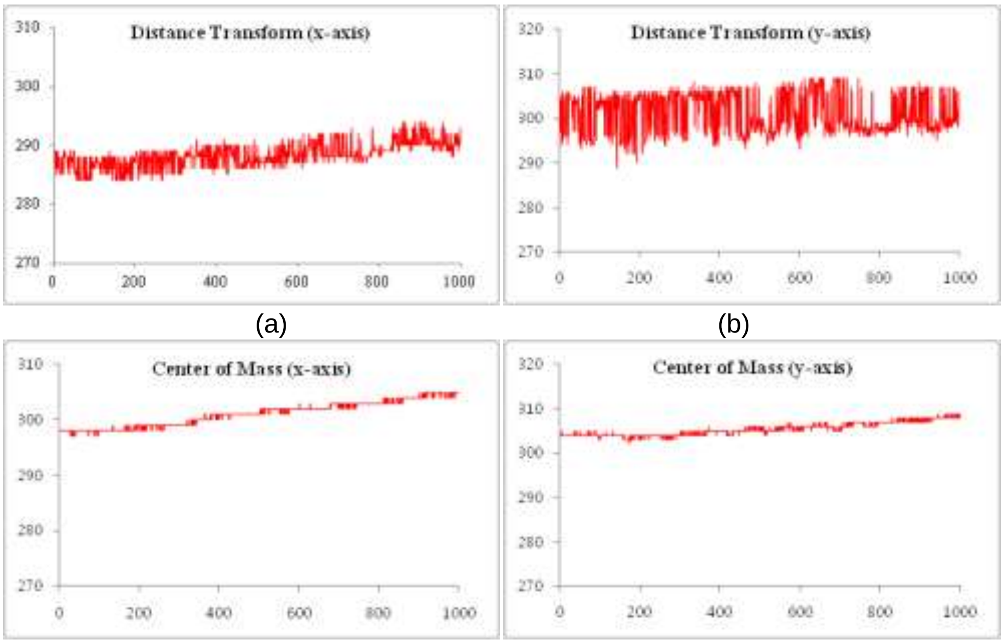
$$c_k(i) = \cos\theta_k = \frac{\overrightarrow{a_k(\iota)}\cdot\overrightarrow{b_k(\iota)}}{|\overrightarrow{a_k(\iota)}|\,|\overrightarrow{b_k(\iota)}|}. \tag{9}$$

In Eq. (9), $\overrightarrow{a_k(\iota)} = \overrightarrow{V_{k+1}} - \overrightarrow{V_k}$, and $\overrightarrow{b_k(\iota)} = \overrightarrow{V_{k-1}} - \overrightarrow{V_k}$, where $\overrightarrow{V_k}$ represents the vector that points to one point on the contour curve, that is $\overrightarrow{V_k} = \left(NA^k, ND^k\right)$. Further, θ is the angle between $\overrightarrow{a_k(\iota)}$ and $\overrightarrow{b_k(\iota)}$.



(a) (b)

Figure 3. Hand Shape Description: (a) Hand Contour and Principal Axis Detection (b) Time-Series Curve of the Hand Contour and the Fingers



(a) (b)

(c) (d)

Figure 4. Estimation of Hand Center: (a) Distance Transform (*x*-axis) (b) Distance Transform (*y*-axis) (c) Center of Mass (*x*-axis) (d) Center of Mass (*y*-axis)

3. Results

To evaluate the efficiency and accuracy of the proposed system, we conduct a set of experiments and compare the results. The experiments are performed using a computer running MS Windows 7, a visual camera, and a Kinect depth-capture device.

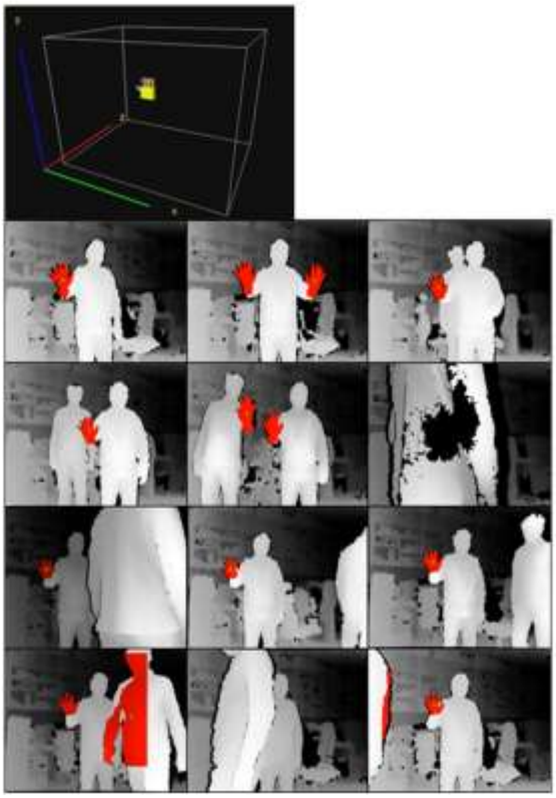


Figure 5. Examples of the Conventional Approach #1 (Soutschek et al.) [6]

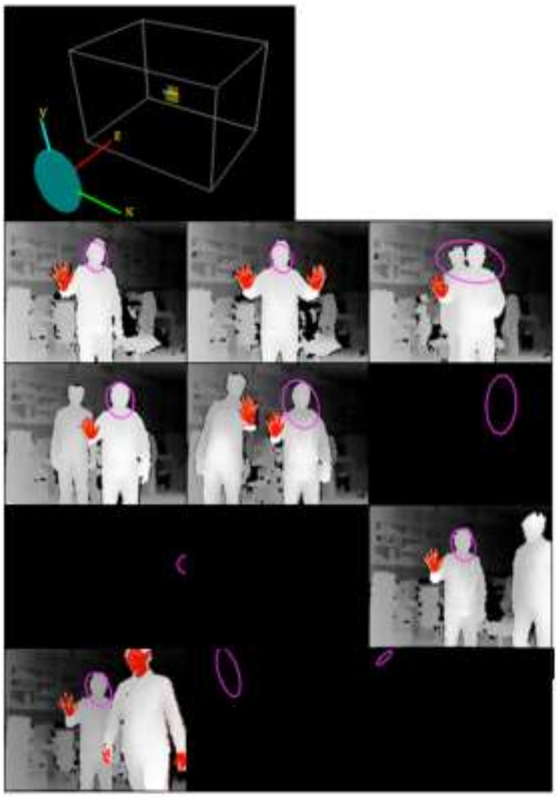


Figure 6. Examples of the Conventional Approach #2 (Bergh et al.) [7]

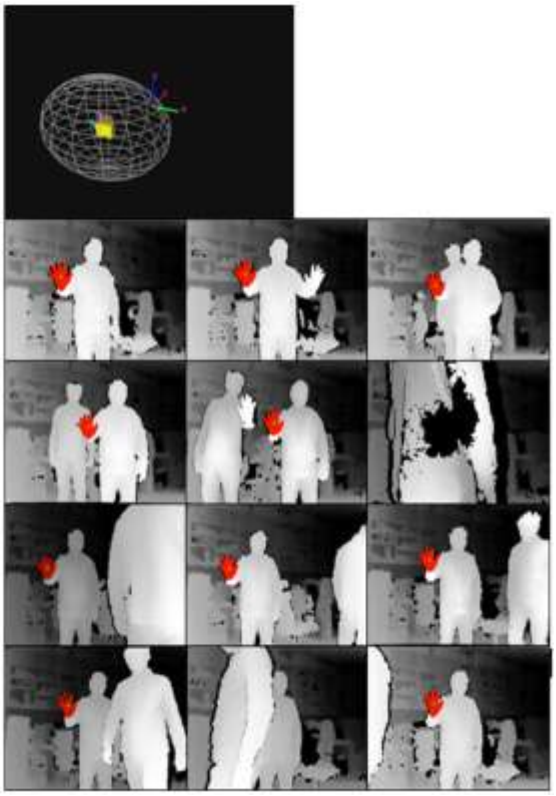


Figure 7. Proposed Method

3.1. Accuracy of Hand Center Estimation

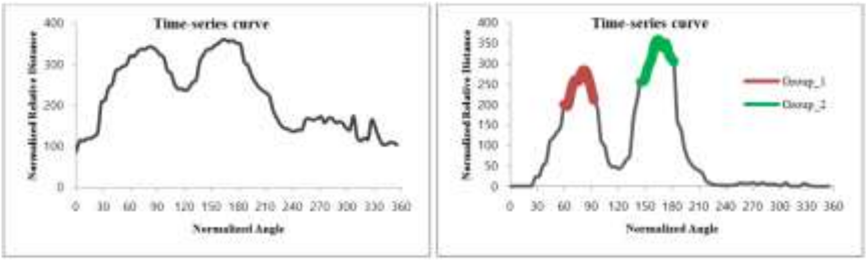
The center point of a region can be computed using several conventional methods such as distance transform [10] and center of mass [11]. Figure 4 presents the results of the center point estimation (a) and (b) the results when using the distance transform, which show that the center point has a lot of noisy jitter. (c) and (d) are the center points obtained by the center of mass method, which verify that the results are very stable. Therefore, we use the center of mass to obtain the center position of the hand.

3.2. Active Tracking Ellipsoid

Figures 5 and 6 illustrate examples of the conventional approaches. In Figure 5, the red regions represent that a user's hand is detected from the depth information obtained by the depth-capture device. For the system to operate successfully, only one user's hand should be detected and tracked. However, the examples show that all objects present in the operation area are detected, resulting in unsuccessful hand tracking. This is because the system assumes the tracking region to be fixed in the 3D scene space [6]. Figure 6 shows the results of Bergh’s approach [7]. In this example, the system first attempts to detect a user’s face, and any object closer to the camera than the face is considered as the user’s hand. If any other face is closer to the camera, the system fails to detect the user’s hand. Nevertheless, as shown in Figure 7, the proposed system restricts the search area using the tracking ellipsoid, and the hand detection is not affected by any other object present in the operation area. Figure 7 also shows that the tracking ellipsoid actively changes according to the hand motion such that it expands toward the direction of the hand motion.

3.3. Evaluation of Hand Gesture Recognition

Figure 8 shows the time-series curves of the hand contours. Here, the x-axis is the angle from the initial point to the final intersection, and the y-axis represents the normalized distance from the hand's center point to the imaged contour. In particular, Figure 8(a) shows a noisy contour curve, which may result in unstable finger counting. Figure 8(b) is a refined curve obtained via the average filter to reduce the noisy components. Figure 9 shows the uncertainty of the hand motion on each coordinate axis. This information is used to determine the shape of the tracking ellipsoid. The 3D tracking ellipsoid is illustrated in Figure 10.



(a) (b)

Figure 8. Time-Series Curve for the Finger Count: (a) before and (b) after Noise Suppression

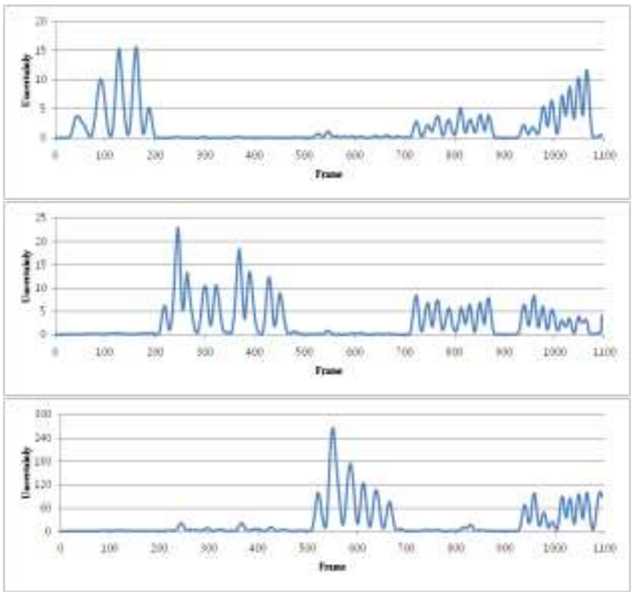


Figure 9. Uncertainties of Hand Motion in *x*, *y*, and *z* axes, Respectively

3.4. Processing Time of the Proposed System

To be used in real applications, the system should run in frame rates, i.e., 24 frames per second. Figure 11 shows the operation time per frame. From the results, we verify that the system operates at a frame rate of approximately 24–30 fps, which ensures the real-time operation of the proposed system. Figure 12 shows the proposed framework in a real application. We build an application in which a user can perform interactions with the system by using his/her bare hand motions. This also verifies that it can be used in the systems, such as a kiosk with a very large screen for which conventional interfaces cannot be used.

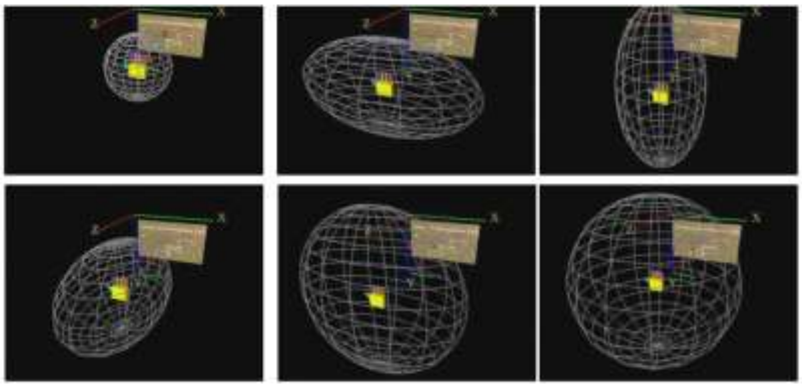


Figure 10. Tracking Ellipsoid that Changes Actively According to the Hand Motion History

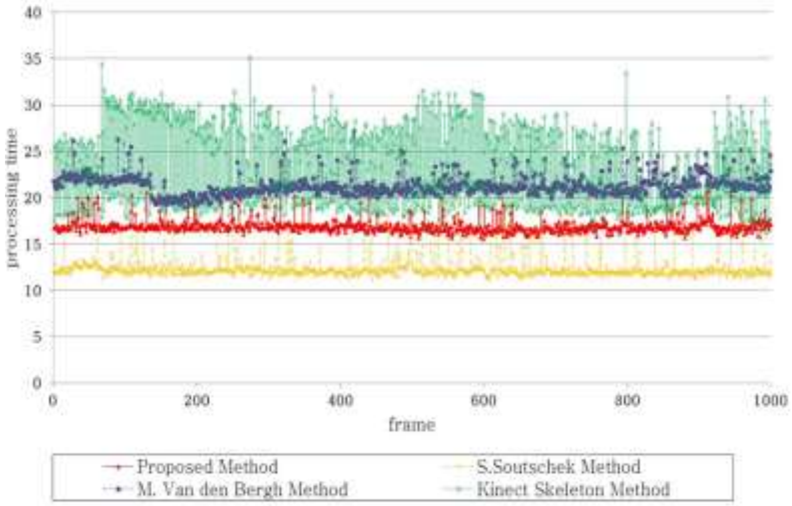


Figure 11. Processing Time of the Proposed System

3. Conclusion and Future Work

We proposed a hand tracking framework that can be used in an NUI-based application. Our system builds a flexible ellipsoid that changes actively according to the statistics of the hand motion history, and the tracking is restricted within the ellipsoidal area. The finger counting technique is used to recognize a user's hand gesture. The experimental results show that the proposed system can be successfully driven in diverse NUI-based systems. Our future work will focus on the development of an application based on the proposed framework.

Acknowledgement

This paper is a revised and expanded version of a paper titled, “Real-time Hand Tracking using the RGB-D Camera,” presented at the “5th International Conference on Interdisciplinary Research Theory and Technology,” Daejeon University, Korea, December 21–23, 2017.

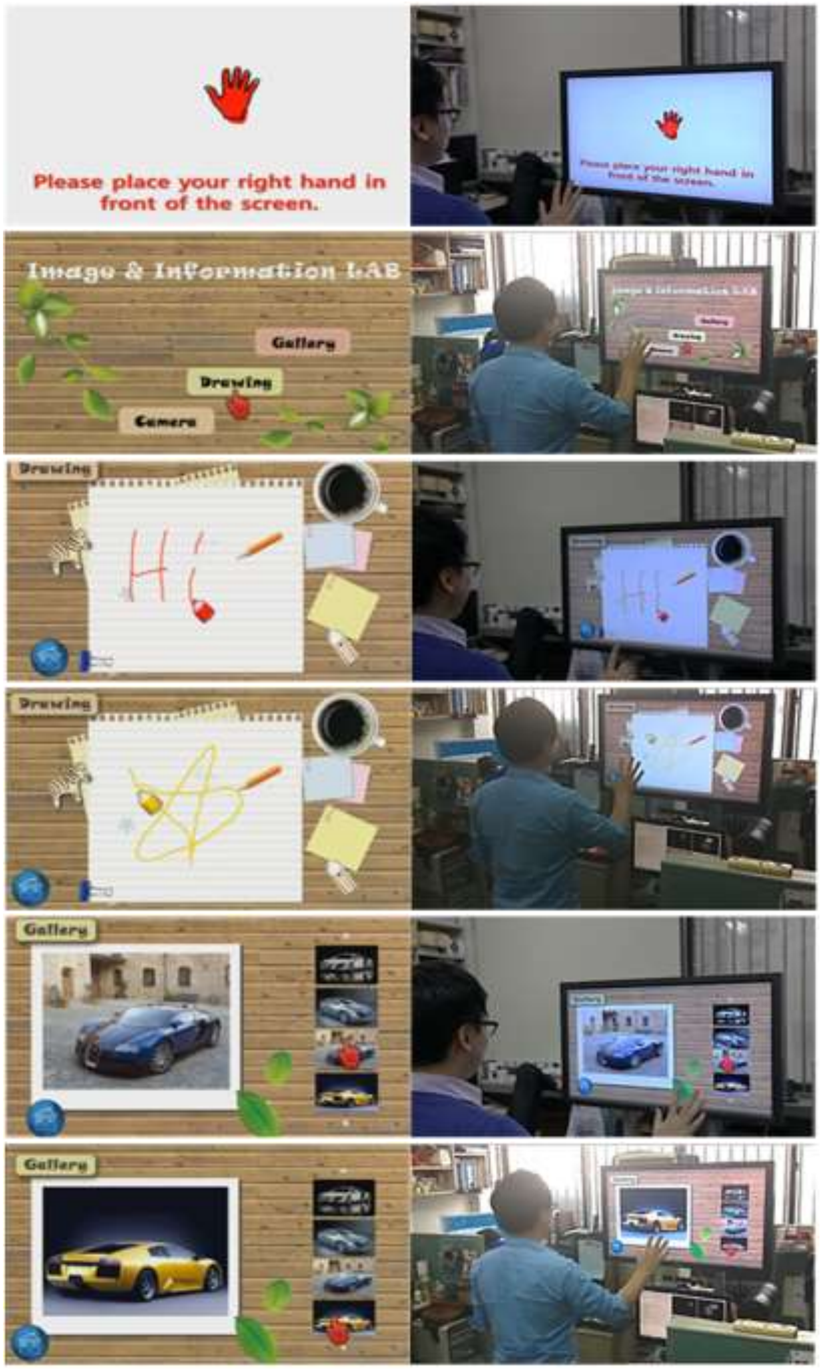


Figure 12. Demonstration of the Proposed System

References

[1] J. Aggarwal and Q. Cai, "Human motion analysis: a review", Computer Vision and Image Understanding, vol. 73, no. 3, **(1999)**, pp. 429-440.

[2] M. M. Hasan and P. K. Mishra, "Hand gesture modeling and recognition using geometric features: a review", Canadian Journal on Image Processing and Computer Vision, vol. 33, no. 1, **(2012)**, pp. 12-26.

[3] S. Rautaray and A. Aggrawal, "Real time multiple hand gesture recognition system for human computer interaction", International Journal of Intelligent Systems and Application, vol. 4, no. 4, **(2012)**, pp. 56-64.

[4] J. Wachs, M Kölsch, H. Stern and Y. Edan, "Vision-based hand-gesture application", Communications of the ACM, vol. 54, no. 2, **(2011)**, pp. 60-71.

[5] Z. Ren, J. Yuan and Z. Zhang, "Robust hand gesture recognition based on finger-earth mover distance with a commodity depth camera", Proceedings of the 19th ACM International Conference on Multimedia, Scottsdale, AZ, **(2011)**.

[6] S. Soutschek, J. Penne, J. Hornegger and J. Kornhuber, "3-D gesture-based scene navigation in medical imaging applications using time-of-flight cameras", IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops (CVPRW'08), Anchorage, AK, **(2008)**.

[7] M. Van den Bergh and L. V. Gool, "Combining rgb and tof cameras for real-time 3d hand gesture interaction", IEEE Workshop on Applications of Computer Vision, Kona, HI, USA, **(2011)**.

[8] P. Viola and M. J. Jones, "Robust real-time face detection", International Journal of Computer Vision, vol. 57, no. 2, **(2004)**, pp. 137-154.

[9] D. Lee and Y. Park, "Vision-based remote control system by motion detection and open finger counting", IEEE Transactions on Consumer Electronics, vol. 55, no. 4, **(2009)**, pp. 2308-2313.

[10] G. Borgefors, "Distance transformations in digital images," Computer Vision, Graphics, and Image Processing, vol. 34, no. 3, **(1986)**, pp. 344-371.

[11] G. Bradski, "Computer Vision Face Tracking as a Component of a Perceptual User Interface", Proceedings of IEEE Workshop Applications of Computer Vision, Princeton, NJ, **(1998)**.

[12] T. Lee and T. Höllerer, "Multithreaded hybrid feature tracking for markerless augmented reality", IEEE Transactions on Visualization and Computer Graphics, vol. 15, no. 3, **(1999)**, pp. 355-368.

[13] Y. Liu, G. Li and Z. Shi, "Covariance tracking via geometric particle filtering", The European Association for Signal Processing Journal on Advances in Signal Processing, vol. 2010, no. 22, **(2010)**, pp. 1-9.

[14] F. Porikli, O. Tuzel and P. Meer, "Covariance tracking using model update based on lie algebra," IEEE Computer Society Conference on Computer Vision and Pattern Recognition, New York, NY, **(2006)**.

[15] O. Tuzel, F. Porikli and P. Meer, "Region covariance: a fast descriptor for detection and classification," European Conference on Computer Vision, Graz, Austria, **(2006)**.

[16] Y. Wu, J. Cheng, J. Wang, H. Lu, J. Wang, H. Ling, E. Blasch and L. Bai, "Real-time probabilistic covariance tracking with efficient model update", IEEE Transactions on Image Processing, vol. 21, no. 5, **(2012)**, pp. 2824-2837.

[17] D. Comaniciu, V. Ramesh and P. Meer, "Real-time tracking of non-rigid objects using mean shift.," Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, Hilton Head, SC, **(2000)**.

[18] N. Bouaynaya, W. Qu and D. Schonfeld, "An online motion-based particle filter for head tracking applications," Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing, Philadelphia, PA, **(2005)**.

[19] M. Yin, J. Zhang, H. Sun and W. Gu, "Multi-cue-based camshift guided particle filter tracking," Journal Expert Systems with Applications: An International Journal, vol. 38, **(2011)**, pp. 6313-6318.

[20] M. Van den Bergh, E. Koller-Meirer and L. Van Gool, "Real-time body pose recognition using 2d or 3d haarlets", International Journal of Computer Vision, vol. 83, no. 1, **(2009)**, pp. 77-84.

Author

Seok-Han Lee, he received his B.S. degree in Electronic Engineering in 1999, and M.S., and Ph.D. degree in Image Engineering, in 2001, and 2009, respectively from Chung-Ang University, Seoul, Korea. From 2001 to 2004, he worked as an engineer in LG Electronics Inc, and from 2010 to 2013, a research professor of Chung-Ang University. He has been working as a professor of Jeonju University since 2013. His research interests include real-time camera tracking, augmented reality, and 3D computer vision.