IEEE Access
Multidisciplinary : Rapid Review : Open Access Journal

Received September 15, 2017, accepted October 23, 2017, date of publication October 31, 2017,
date of current version December 5, 2017.

Digital Object Identifier 10.1109/ACCESS.2017.2768405

Hand-Mouse Interface Using Virtual Monitor Concept for Natural Interaction

CHANGHYUN JEON[1], OH-JIN KWON[2], DONGIL SHIN[1], AND DONGKYOO SHIN[1]
[1]Department of Computer Engineering, Sejong University, Seoul 05006, South Korea
[2]Department of Electrical Engineering, Sejong University, Seoul 05006, South Korea

Corresponding author: Dongkyoo Shin (shindk@sejong.ac.kr)

This work was supported by the National Research Foundation of Korea, Ministry of Education, through the Basic Science Research Program, under Grant 2015R1D1A1A01059253.

ABSTRACT The growing interest in human–computer interaction has prompted research in this area. In addition, research has been conducted on a natural user interface/natural user experience (NUI/NUX), which utilizes a user's gestures and voice. In the case of NUI/NUX, a recognition algorithm is needed for the gestures or voice. However, such recognition algorithms have weaknesses because their implementation is complex, and they require a large amount of time for training. Therefore, steps that include pre-processing, normalization, and feature extraction are needed. In this paper, we designed and implemented a hand-mouse interface that introduces a new concept called a "virtual monitor", to extract a user's physical features through Kinect in real time. This virtual monitor allows a virtual space to be controlled by the hand mouse. It is possible to map the coordinates on the virtual monitor to the coordinates on the real monitor accurately. A hand-mouse interface based on the virtual monitor concept maintains the outstanding intuitiveness that is the strength of the previous study and enhances the accuracy of mouse functions. In order to evaluate the intuitiveness and accuracy of the interface, we conducted an experiment with 50 volunteers ranging from teenagers to those in their 50s. The results of this intuitiveness experiment showed that 84% of the subjects learned how to use the mouse within 1 min. In addition, the accuracy experiment showed the high accuracy level of the mouse functions [drag (80.9%), click (80%), double-click (76.7%)]. This is a good example of an interface for controlling a system by hand in the future.

INDEX TERMS NUI/NUX, virtual monitor, hand mouse, gesture recognition, Kinect.

I. INTRODUCTION

Since the computer was developed, it has become an indispensable part of our lives. With the growing importance of computers in day-to-day lives, peoples' thoughts have naturally focused on making their use easy and convenient. For this reason, studies on human-computer interaction (HCI) have been actively conducted. HCI aims at creating or improving computing systems, including their functionality, reliability, usability, and efficiency. From this viewpoint, the user interface (UI) is an important part of HCI, leading to several studies on the same. UI has gradually evolved from a command-line interface (CLI) used to communicate with a computer using simple commands, to a graphical user interface (GUI). Recently, studies on a natural user interface/natural user experience (NUI/NUX) have been carried out [1]–[3]. GUI is currently the most widely-used type for communication between the user and computer, with input devices such as a mouse and keyboard. On the other hand, NUI/NUX is characterized by the use of a user's natural gestures, voice, etc., to communicate with the computer without any special input devices. To accomplish this, machine learning, image processing, and signal processing algorithms are used for gesture and voice recognition. However, because these algorithms are very difficult to implement and require a large amount of time for training to achieve recognition, it is difficult to realize an NUI/NUX. Therefore, as an alternative, it is necessary to develop a UI with a simple implementation and excellent performance.

Recently, Microsoft's Kinect has gained great popularity among the general public and developers as a development tool for NUI/NUX. Kinect can easily obtain depth information that cannot be accessed using a regular camera through a sensor. This can assist in the recognition of a gesture or voice, through the software development kit (SDK) provided, which can handle image processing and speech recognition without additional algorithms. Kinect also implements a skeleton image, which has the advantage of making it easy to obtain the location and depth information for each joint. Examples of studies using Kinect include a system that selects the objects in a virtual world through a Kinect camera and software for mobility-impaired seniors [4], [5]. In addition, a study on a hand mouse, which is the basis of this research, was conducted in 2014 [6].

In this study, the proposed system is a hand-mouse interface that introduces a new concept called a "virtual monitor" to extract a user's physical features in real time. We can easily obtain a user's physical features using Kinect. A virtual monitor is generated based on a user's physical features. It is possible for a user to accurately control a mouse pointer using their hands. Moreover, the virtual monitor is intuitive, because it is used like a touch screen. Through this, we propose an intuitive hand-mouse interface with better accuracy than previous such interfaces.

In section 2, we review the previous work on an NUI/NUX- related hand mouse. Section 3 describes the hand-mouse interface design and implementation. In this section, we explain the total flow of the hand-mouse interface. We then explain how to create the virtual monitor, coordinate the transformation algorithm, etc. Section 4 presents an experimental evaluation of the accuracy and intuitiveness. To verify the accuracy, we compare the results with those of another hand mouse, using OpenCV to demonstrate the superiority of the proposed system.

II. RELATED WORK

A. NATURAL USER INTERFACE

A UI is used to issue commands to control data input and operate a system or software. The purpose of a UI is to ensure smooth communication between the user and the software or computer. Many computer engineers have conducted studies on UIs for convenient communication and easy use. Currently, we are moving into the era of the NUI, which uses a user's natural gestures and voice, through the era of the most widely-used GUI. Examples of NUIs include a user interface based on gestures using fiber, magnetic sensors, and a data glove [7]. A UI using physical devices will accurately recognize a user's gestures, but the specific physical device must be attached to the user. Another example is a UI based on voice, which is the most universal and easiest method of communication between humans. A typical example of a UI using voice is Apple Corporation's Siri. Speech recognition is very intuitive, but may be weak or unrecognizable because of the noise from the surrounding environment [8].

However, appropriate recognition algorithms for each are needed to realize a UI based on gestures or voice. The recognition algorithm needs steps such as feature extraction, normalization, pre-processing, and a machine-learning algorithm like a hidden Markov model (HMM) [9]–[11]. These steps are very complex and cumbersome, and are not easy to implement. Another disadvantage is the time that needs to be invested in training.

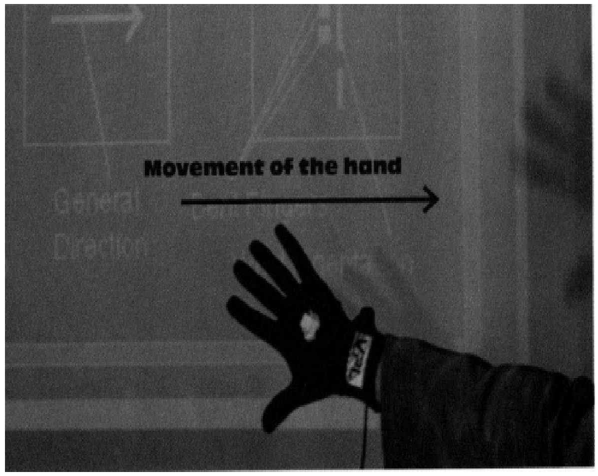

FIGURE 1. Example of natural user interface using data glove.

B. HAND MOUSE

A hand mouse refers to a control method that allows the user to control the mouse pointer using their hands. There have been frequent attempts to control a system or software using the hand. A study on the use of a data glove to make a presentation was conducted in 1993 (See Fig. 1 [12]).

In recent years, an NUI system was developed that recognizes the areas of the hand using a camera and controls the mouse by reconstructing it in a three-dimensional space [13]. However, the existing hand-mouse interfaces have the disadvantage of the user needing to be aware of the instructions for a pre-defined hand.

In our previous work, we designed a hand mouse scheme that determines the "move," "click," and "double click" mouse operations by recognizing the moving distances of the left and right hands in three-dimensional space. The hand-mouse interface in the previous study is shown in Fig 2. However, a user had some difficulty with this scheme, due to the low accuracy of the mouse function [6]. This problem was caused by the need to extract the user's physical features in real time. When a user uses the hand-mouse interface, subtle changes occur, (because the user's body is not fixed), which affect the mouse control area, making it difficult for the user to control the mouse pointer.

III. VIRTUAL MONITOR CONCEPT-BASED HAND-MOUSE INTERFACE

A. VIRTUAL MONITOR CONCEPT

The virtual monitor concept has recently been introduced to eliminate the disadvantages found in the study on an NUI/NUX framework [6]. These disadvantages are the result of subtle changes in the physical features that are extracted, when the user controls the mouse pointer using their hands in real time. A user's physical features are obtained using a

FIGURE 2. Hand-mouse interface in previous study.

Kinect camera, but these are not accurately converted to the coordinates on a real monitor, because the depth information and coordinate data can be easily changed by even a slight movement by the user. In addition, when the user activates the double click function, which measures the moving distance based on the initial position of the hands, if the user's action does not occur within the specific time, it cannot be recognized. For example, if the user is not able to click twice to perform the double click function within 0.3 s when the time delay is 0.3 s, this action amounts not to a double click, but a single click performed twice. The virtual monitor concept is introduced in order to overcome these disadvantages.

In this paper, a virtual monitor is defined as a "virtual space where the mouse pointer can be controlled by the hands between the Kinect and the user." The advantage that results from using the virtual monitor is a reference point that can control the mouse pointer. In addition, the coordinates in the virtual space can express the coordinates on the real monitor accurately, making the movement of the mouse pointer natural. In addition, it is possible to increase the accuracy of the mouse functions and make them more intuitive, because the user can use the virtual monitor as a touch-screen.

A user's physical features are used to create a virtual monitor, because every user's physical features are different, including their height and arm length. Therefore, the scope to deal with the interface for each person individually is limited and different. Thus, the virtual monitor must be created with the appropriate size and location for each user by extracting the user's physical features, to make it user-friendly. The proposed hand-mouse interface extracts a user's physical features using Kinect in real time. The RGB image that is obtained using the Kinect SDK can reconstruct a skeleton image. The coordinates and depth of each joint are obtained using this skeleton image. Fig. 3 shows the skeleton joint information that can be obtained using the Kinect SDK.

As shown in Fig. 3, Kinect can recognize 20 joints of a user. The physical features used in the proposed system include the arm length, shoulder width, and spine and head positions. In addition, the distance between the user and Kinect is regarded as a physical feature and used accordingly. The physical features and functions used are listed in Table 1.

A virtual monitor is created using the physical features listed in Table 1, as shown in Fig. 4. The size and height

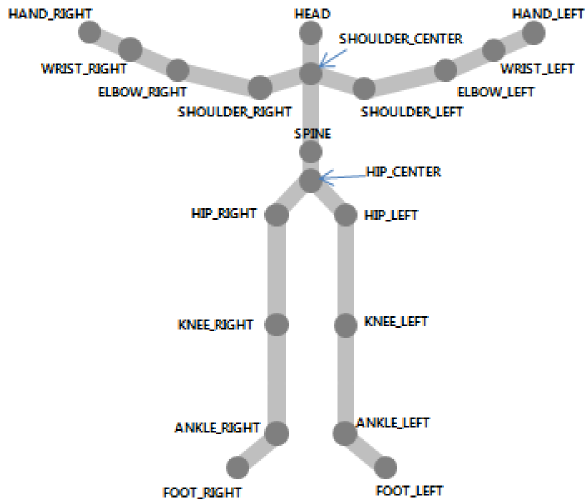

FIGURE 3. Skeleton joint information that can be obtained using Kinect SDK.

TABLE 1. Applications in each class.

Physical features	Function
Arm length	The distance between the user and the virtual monitor
Shoulder width	The virtual monitor's width length
Spine position	The virtual monitor height length and height
Head position	The virtual monitor height length and height
User's location	The virtual monitor's position

position of the virtual monitor is based on the user's shoulder width (α) and the distance (β) between the user's face and spine. When a user uses the hand-mouse interface, the range that can be controlled by their hands is set to two times the width of the user's shoulders, to avoid making the space too wide or too narrow. Usually, if the width of the virtual monitor is set to two times the shoulder width because the length of the user's arm is greater than the shoulder width, the user can control the full range of the virtual monitor without moving. Additionally, when the user lifts an arm, the height of the virtual monitor is set to the distance between the head and spine, to prevent it from being too high or low, in comparison to a user's height, to naturally control the virtual monitor. If the size of the virtual monitor is set using the proposed method, the hand-mouse interface can be used by creating the correct size and height of the virtual monitor for each user, enabling use by those with different physical features 01.

A virtual monitor is created using the physical features listed in Table 1, as shown in Fig. 4. The size and height position of the virtual monitor is based on the user's shoulder width (α) and the distance(β) between the user's face and spine. When a user uses the hand-mouse interface, the range

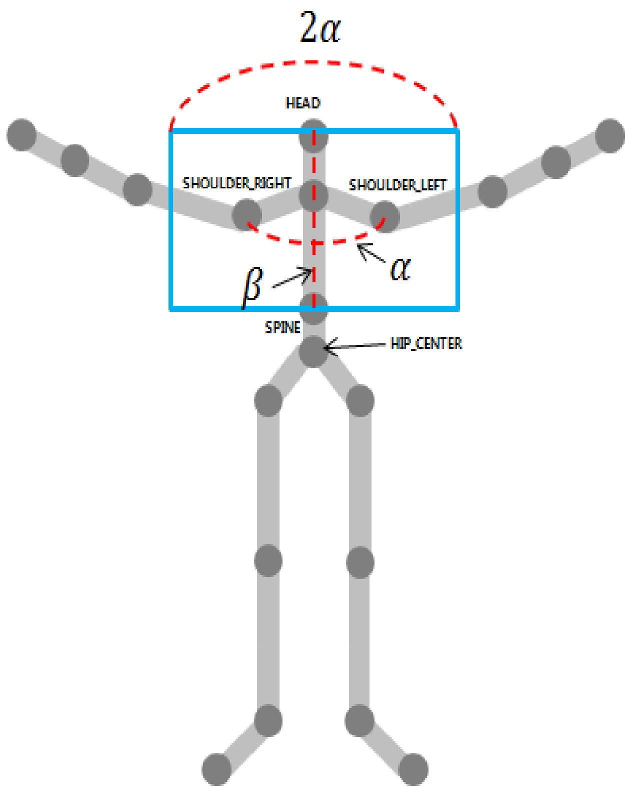

FIGURE 4. Size and position of virtual monitor using the user's physical features.

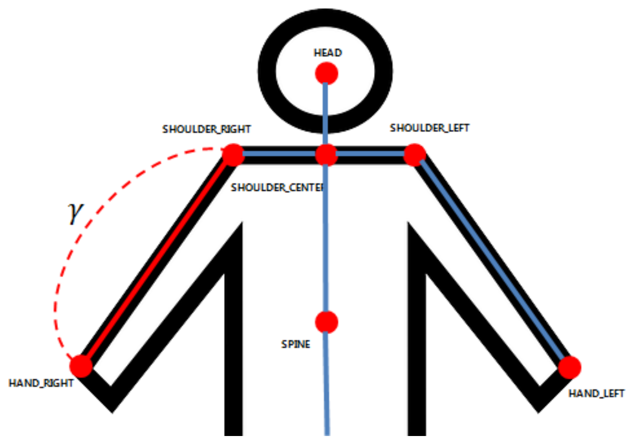

FIGURE 5. User's arm length.

that can be controlled by their hands is set to two times the width of the user's shoulders, to avoid making the space too wide or too narrow. Usually, if the width of the virtual monitor is set to two times the shoulder width because the length of the user's arm is greater than the shoulder width, the user can control the full range of the virtual monitor without moving, as shown in Fig 5. Additionally, when the user lifts an arm, the height of the virtual monitor is set to the distance between the head and spine, to prevent it from being too high or low, in comparison to a user's height, to naturally control the virtual monitor, as shown in Fig. 6. If the size of the virtual monitor is set using the proposed method, the hand-mouse

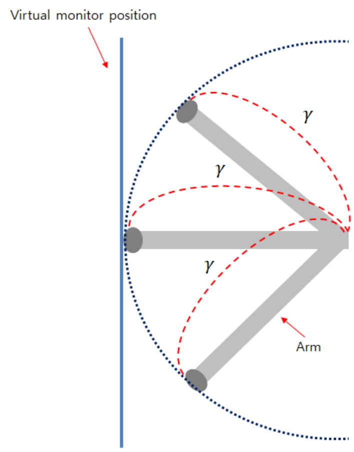

FIGURE 6. Problem with the distance between user and virtual monitor using user's whole arm length.

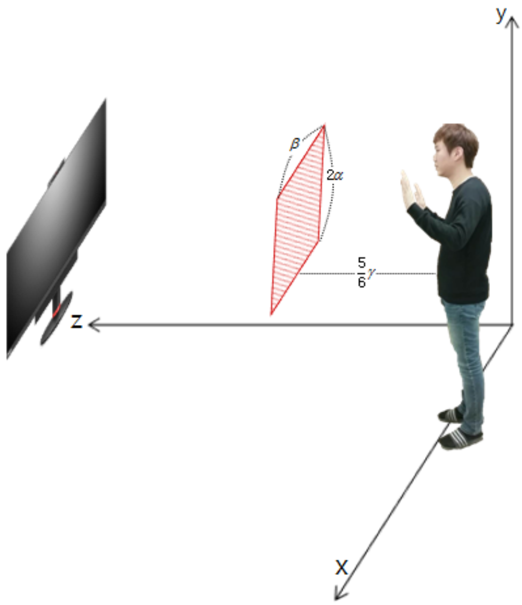

FIGURE 7. Positional relationship between virtual monitor and user, with real monitor.

interface can be used by creating the correct size and height of the virtual monitor for each user, even if the user is different, with different physical features as shown in Fig 7.

B. CONTROL METHOD OF HAND MOUSE FUNCTION

An intuitive UI was introduced in ''Toward the Use of Gesture in Traditional User Interfaces'' by Kjeldsen et al., in 1996. In this, a user's hand was recognized by a camera from the moment the user placed the mouse. Then, the user selected a virtual menu that was created on the real monitor using the hand [14]. This study was followed by several others on UIs and new methods using gestures, but each new interface had the burden of memorizing a gesture for a user command. Thus, these interfaces were not intuitive at all. In addition, only a user who was aware of more than 15 gestures could use the interface. In this study, we designed and implemented a

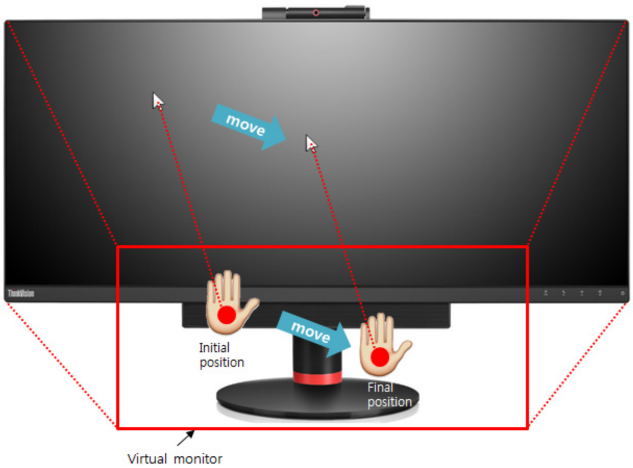

FIGURE 8. Movement of mouse pointer by movement of left hand in virtual monitor.

hand mouse interface reminiscent of the most commonly and intuitively used mouse actions. The actions that a user takes to use a mouse include mouse movements and button clicks. Therefore, actions that may be reminiscent of the mouse can be controlled by simply moving (left hand) and pressing (right hand) the mouse. The use of a hand-mouse interface with a user's hand is very simple and intuitive. The left hand controls the movement of the mouse, and the right hand controls a mouse click. If the left hand moves in the virtual monitor, the left hand coordinate of the virtual monitor can be controlled by converting the coordinates of the mouse pointer on the real monitor. Fig. 8 shows the movement of the mouse pointer by the movement of the left hand in the virtual monitor.

As can be seen from Fig. 8, because the movement of the mouse pointer can be controlled in the same way as the left hand movement on the virtual monitor, the user can move the mouse pointer exactly to the desired position on the real monitor. Simultaneously, it is possible to provide the user with intuitive feedback.

While the left hand can control the movement of the mouse pointer, the right hand can control the mouse click function. As shown in Fig. 9 the movement of the right hand to control the click function on the virtual monitor, which can be recognized by stretching out and bending the user's right hand from position 1 to position 2. The behavior of the right hand can be understood in the sense of touching the touch-screen with the right hand. By merely using the depth information of the right hand and virtual monitor to confirm the movement of the right hand, the click function can be controlled in an intuitive and simple manner.

C. COORDINATE TRANSFORMATION ALGORITHM

In order to express the movement of the mouse pointer on the real monitor based on the movement of the left hand on the virtual monitor, it is necessary to accurately transform the coordinates on the real monitor using the left hand coordinates of the virtual monitor. Accurate coordinate transformation helps the user to accurately express the position of the mouse with the left hand movement, and the mouse pointer can be moved up, down, left, or right to move naturally on the real monitor. Thus, plane V, which contains the virtual monitor, and plane R, which contains the real monitor, are set in order to transform the coordinates for the real monitor using the virtual monitor, as shown in Fig. 10.

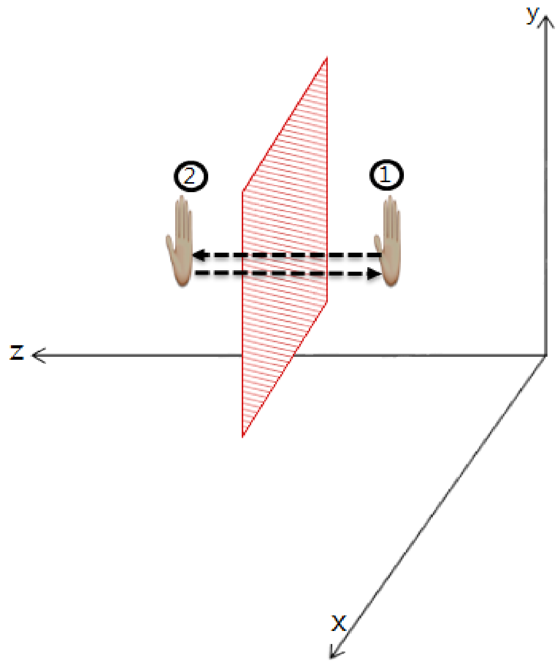

FIGURE 9. Movement of right hand to control click function.

This study used the ratio of the virtual monitor and real monitor for the transformation algorithm. We used the resolution of the monitor because users have different size monitors, and the size of the virtual monitor can be easily obtained because it is generated by physical features. The horizontal and vertical ratio is multiplied by the ratio for the left hand coordinate on the virtual monitor. The coordinate transformation algorithm may be represented by the following formula.

$$\mathrm{X}_{\mathrm{rate}} = R_{size\_x}/V_{size\_x} \tag{1}$$

$$\mathrm{Y}_{\mathrm{rate}} = R_{size\_y}/V_{size\_y} \tag{2}$$

$$R_{mp}(x, y) = v_{lhp}(x \times X_{rate,y} \times Y_{rate}) \tag{3}$$

D. OVERALL FLOW OF HAND-MOUSE INTERFACE

In the previous sections, a method for generating the hand-mouse interface using the virtual monitor was been described. In this section, the proposed hand-mouse interface will be described by integrating these sections. Integrating the previous sections, the proposed hand mouse interface can be divided into three steps, as shown in Fig. 11.

The steps of Fig. 11 are described below

1. Reconstruct the input RGB image into the skeleton image using Kinect SDK.

2. Extract the user's physical features using the joint information of the skeleton image and create a virtual monitor.

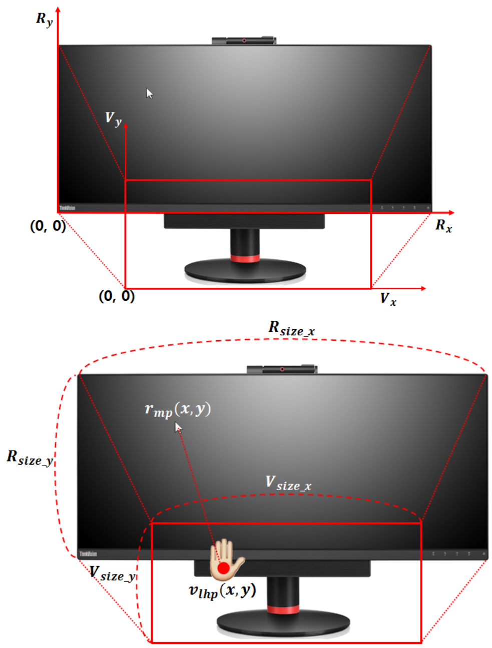

FIGURE 10. Coordinate and coordinate transformation to plane *R* from plane *V*.

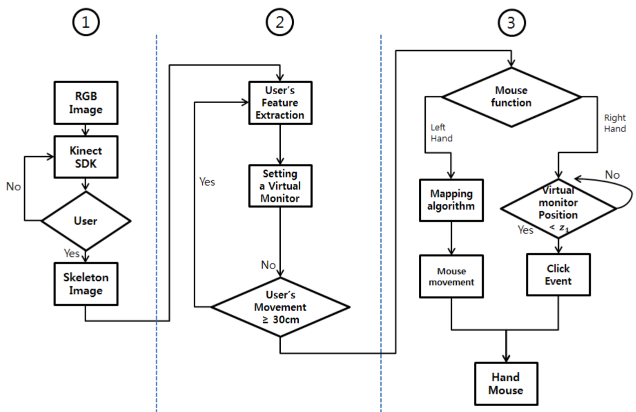

FIGURE 11. Flow chart of hand mouse interface.

3. Perform each function after separating the left and right hands
   - left hand: the movement of the mouse is used to convert the coordinates on the created virtual monitor to coordinates on the real monitor
   - right hand: the mouse click function is confirmed using the depth information of the right hand

The first step is the initialization. To create and use the hand-mouse interface, the Kinect is initialized, and the RGB image input using Kinect is reconstructed into a skeleton image. In the next step, physical features are extracted, using the joint information of the reconstructed skeleton image. Then, the virtual monitor is generated. Although not described in the previous section, a point to be noted is that the user can move when using the hand-mouse interface. One of the features used for creating a virtual monitor is the user's location.

```
while(true)
{
    if(virtual monitor is null)
    {
        width = get_shoulder_width()
        height = get_head_spine_height()
        position = get user position()

        make_virtual_monitor(position, width, height)
    }

    if (move >= 30cm)
    {
        width = get_shoulder_width()
        height = get_head_spine_height()
        position = get_user_position()

        make virtual monitor(position, width, height)
    }
    else
    {
        if( Left hand and Right hand in virtual monitor )
        {
            if (Left hand)
            {
                mouse_point(x,y) = left_hand_point(x,y) * ratio
                if (concentrate_indicator - 0.03 <= EEG <= concentrate_indicator + 0.03)
                {
                    if(VM depth >= Right hand depth)
                        mouse click

                    else if(VM depth < right hand depth)
                        mouse not click
                }
            }
        }
    }
}
```

FIGURE 12. Pseudo code of virtual monitor-based hand-mouse interface.

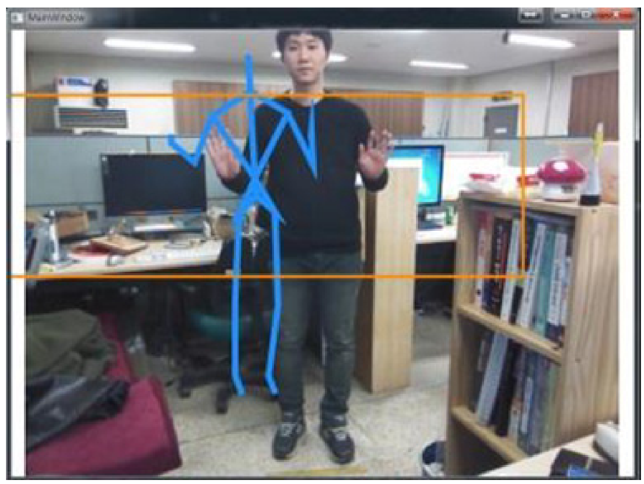

FIGURE 13. Virtual monitor-based hand mouse software.

The location of the user is considered to provide the correct size, allowing the user to stand in any position in front of the camera that they feel is convenient. Accordingly, the user's movement is checked to determine whether or not to regenerate the virtual monitor. Finally, it is possible to control the mouse pointer on the real monitor through an appropriate command by the left and right hands. Fig. 12 shows the pseudo code of the virtual monitor-based hand-mouse interface, and Fig. 13 provides an example of a system applying the hand mouse interface.

IV. EXPERIMENTAL RESULTS AND ANALYSIS

The proposed hand mouse interface is as kind of NUI that utilises the user's intuitive and natural gestures. In order to satisfy the characteristics of a UI, everyone should be able to use the system easily and accurately. For this reason, an experiment was carried out to verify its intuitiveness and accuracy. An intuitiveness experiment was performed to check whether anyone could easily use it, as was an accuracy

TABLE 2. Intuitive experiment results.

(unit : sec)

	10's m	10's f	20's m	20's f	30's m	30's f	40's m	40's f	50's m	50's f
1	42	48	27	21	32	19	55	45	57	66
2	35	45	25	54	38	64	41	38	65	74
3	55	31	34	33	37	58	42	31	52	64
4	14	62	48	25	24	49	30	61	78	48
5	38	55	55	43	32	34	43	46	45	55
Avg.	36.8	48	37.8	35.2	32.6	44.8	42.2	44.2	59.4	61.4

(m: male, f: female)

TABLE 3. Accuracy experiment results.

	10' s m	10' s f	20' s m	20' s f	30' s m	30' s f	40' s m	40' s f	50' s m	50' s f	Avg
Drag	85	78	86	83	84	83	80	70	79	81	80.9
click	79	85	91	88	83	76	71	72	81	74	80
Double click	76	70	72	81	80	86	72	75	76	79	76.7

experiment, to check whether it correctly determined the user's intent. The results of each experiment are shown in the following sections.

A. INTUITIVE EXPERIMENT RESULTS

Intuitiveness is an essential characteristic of an NUI. Intuitiveness is not easy to evaluate objectively, due to its abstractness. In order to evaluate the intuitiveness of the proposed system, we did not provide subjects with instructions on the proposed hand-mouse interface, but measured the time taken to learn how to use it. The experiment was conducted with 50 subjects varying in age from teenagers to those in their fifties, including men and women and theresults are listed in Table 2. As can be seen therefrom, most of the subjects learned how to use the hand- mouse interface within 1 min.

B. ACCURACY EXPERIMENT RESULTS

Accuracy is an indispensable feature of a UI. If a user does not control the system or software according to their intention, it would mean that UI is not functioning properly. Therefore, accuracy is essential to evaluate a UI. In order to test the accuracy of the hand mouse interface, it was ascertained whether it was possible to accurately recognize the functions (click, double click and drag) of a traditional mouse as the user's intention. Each function was performed ten times, and the accuracy was measured by the number of successful attempts.

TABLE 4. Comparison of intuitive experiment results.

	Trial	Count	Accuracy
Click using the OpenCV library [15]	200	180	90%
Double Click using the OpenCV library [15]	200	185	92.5%
Click using the proposed method	200	195	97.5%
DoubleCclick using the proposed method	200	188	94%

The accuracy experiment results are listed in Table 3. The accuracies of the functions (drag, click, double click) were 80.9%, 80%, and 76.7%, respectively.

In addition, by way of comparison, accuracy results for a hand mouse using OpenCV and the proposed hand mouse are listed in Table 4. A comparative experiment measured the number of attempts that were accurately recognized while performing the click and double click functions 200 times. The accuracies of the click and double click functions of the hand mouse using OpenCV were 90% and 92.5%, respectively [15]. The accuracies of the proposed system were better, at 97.5% and 94%, respectively.

V. CONCLUSION

As depicted in many science fiction movies, the time when a user can control a system or software freely, using gestures and voice, is not far off. Already NUI/NUX technology using speech recognition such as Apple's Siri has become commonplace. In this study, an NUI was implemented without machine learning for gesture recognition. The virtual monitor-based hand-mouse interface to extract a user's physical features using Kinect was designed and implemented. This approach was relatively easy to implement, compared to the traditional gesture recognition using machine learning. In addition, the intuitiveness and accuracy of the proposed hand-mouse interface were measured through experiments, and the results were excellent. These results indicate that the method proposed in this paper is a good example that can be applied to systems for controlling a mouse with the hands. Furthermore, the proposed system is expected to emerge as a new interface, in conjunction with other techniques.

REFERENCES

[1] W. Bland, T. Naughton, G. Vallee, and S. L. Scott, "Design and implementation of a menu based OSCAR command line interface," in *Proc. 21st Int. Symp. High Perform. Comput. Syst. Appl.*, Washington, DC, USA, May 2007, p. 25.

[2] M. Park, "A study on the research on an effective graphic interface design of the Web environment for the people with disability-focused on the people with hearing impairment," M.S. thesis, Dept. Design, Sejong Univ., Seoul, South Korea, Feb. 2005.

[3] M. N. K. Boulos, B. J. Blanchard, C. Walker, J. Montero, A. Tripathy, and R. Gutierrez-Osuna, ''Web GIS in practice X: A microsoft kinect natural user interface for Google earth navigation,'' *Int. J. Health Geogr.*, vol. 10, pp. 1–14, Jul. 2011.

[4] M. F. Shiratuddin and K. W. Wong, ''Non-contact multi-hand gestures interaction techniques for architectural design in a virtual environment,'' in *Proc. Int. Conf. Inf. Technol. Multimedia*, Nov. 2011, pp. 1–6.

[5] I.-T. Chiang, J.-C. Tsai, and S.-T. Chen, ''Using Xbox 360 Kinect games on enhancing visual performance skills on institutionalized older adults with wheelchairs,'' in *Proc. IEEE 4th Int. Conf. Digit. Game Intell. Toy Enhanced Learn.*, Mar. 2012, pp. 263–267.

[6] G. Lee, D. Shin, and D. Shin, ''NUI/NUX framework based on intuitive hand motion,'' *J. Internet Comput. Services*, vol. 15, no. 3, pp. 11–19, Jun. 2014.

[7] D. J. Sturman and D. Zeltzer, ''A survey of glove-based input,'' *IEEE Comput. Graph. Appl.*, vol. 14, no. 1, pp. 30–39, Jan. 1994.

[8] J. Choi, ''Speech and noise recognition system by neural network,'' *J. Korea Inst. Electron. Commun. Sci.*, vol. 5, no. 4, pp. 357–362, Aug. 2010.

[9] F.-S. Chen, C.-M. Fu, and C.-L. Huang, ''Hand gesture recognition using a real-time tracking method and hidden Markov models,'' *Image Video Comput.*, vol. 21, no. 8, pp. 745–758, Aug. 2003.

[10] J. O. Kim, ''Bio-mimetic recognition of action sequence using unsupervised learning,'' *J. Internet Comput. Service*, vol. 15, no. 4, pp. 9–20, Aug. 2014.

[11] J. Kim, ''BoF based action recognition using spatio-temporal 2D descriptor,'' *J. Internet Comput. Service*, vol. 16, no. 3, pp. 21–32, Jun. 2015.

[12] T. Baudel and M. Beaudouin-Lafon, ''Charade: Remote control of objects using free-hand gestures,'' *Commun. ACM*, vol. 36, no. 7, pp. 28–35, Jul. 1993.

[13] A. A. Argyros and M. I. A. Lourakis, ''Vision-based interpretation of hand gestures for remote control of a computer mouse,'' in *Proc. ECCV*, 2006, pp. 40–51.

[14] R. Kjeldsen and J. Kender, ''Toward the use of gesture in traditional user interfaces,'' in *Proc. 2nd Int. Conf. Autom. Face Gesture Recognit.*, 1996, pp. 151–156.

[15] S. Kang, C. Kim, and W. Son, ''Developing user-friendly hand mouse interface via gesture recognition,'' in *Proc. Conf. Korean Soc. Broadcast Eng.*, 2009, pp. 129–132.

CHANGHYUN JEON received the M.S. degree from Sejong University, Seoul, South Korea, in 2016. His research interests include NUI/NUX through gesture detection.

OH-JIN KWON received the M.S. degree in electrical engineering from the University of Southern California, Los Angeles, in 1991, and the Ph.D. degree in electrical engineering from the University of Maryland, College Park, in 1994. From 1984 to 1989, he was a Researcher with the Agency for Defense Development of Korea, and from 1995 to 1999, he was the Head of the Media Laboratory with Samsung SDS Co., Ltd., Seoul, South Korea. Since 1999, he has been a Faculty Member with Sejong University, Seoul, where he is currently an Associate Professor. His research interests are image and video fusion, coding, watermarking, analyzing, and processing.

DONGIL SHIN received the B.S. degree in computer science from Yonsei University, Seoul, South Korea, in 1988, the M.S. in computer science from Washington State University, Pullman, WA, USA, in 1993, and the Ph.D. degree from University of North Texas, Denton, TX, USA, in 1997. He was a Senior Researcher with the System Engineering Research Institute, Deajun, South Korea, in 1997. Since 1998, he has been with the Department of Computer Engineering, Sejong University, South Korea, where he is currently a Professor. His research interests include Information security, bio-signal data processing, data mining, and machine learning.

DONGKYOO SHIN received the B.S. degree from Seoul National University, South Korea, in 1986, the M.S. degree from the Illinois Institute of Technology, Chicago, IL, USA, in 1992, and the Ph.D. degree from Texas A&M University, College Station, TX, USA, in 1997, all in computer science. From 1986 to 1991, he was with the Korea Institute of Defense Analyses, where he developed database application software. From 1997 to 1998, he was with the Multimedia Research Institute, Hyundai Electronics Co., South Korea, as a Principal Researcher. He is currently a Professor with the Department of Computer Engineering, Sejong University, South Korea. His research interests include natural user interaction, information security, bio-signal data processing, ubiquitous computing.