A Robust Hand Cursor Interaction Method Using Kinect

Zhiwen Lei[1,2], Xiaoxiao Yang[1], Yanzhou Gong[1], Weixing Huang[1], Jian Wang[1], Guigang Zhang[1]
1. Institute of Automation, Chinese Academy of Science, Beijing, China
2. University of Chinese Academy of Science, China
{leizhiwen2016, xiaoxiao.yang, yanzhou.gong, weixing.huang, jian.wang, guigang.zhang}@ia.ac.cn
*Corresponding author: Jian Wang(jian.wang@ia.ac.cn)

***Abstract*—In this paper, we present a realtime natural interaction system by using Kinect sensor. It can stability and smoothly control hand like mouse by user's holding hands, and implement common mouse operations such as 'clicking', 'dragging' and 'dropping' so on. Our interaction system is made of several novel technique. It can identify the user's interaction with intent by detecting the engaged/disengaged gestures, and distinguish the primary user in the crowd, build a physical interaction zone to map the user's 3D hand position to 2D screen position in a timely and user-friendly manner. Our main contribution is to combine gesture recognition and gesture-tracking, and to implement an interaction application system with the details.**

Keywords-component; Kinect;Human-Computer Interaction; Hand Cursor

I. INTRODUCTION

In the field of human-computer interaction, with the development of mobile phones, touch screens gradually replace the mouse and keyboard as the most using up to interaction device. In recent years, the development of AR and VR technology, put forward new demands for development of human-computer interaction. Currently, the most used VR interactive devices are HTC controller, Oculus Touch. But these interactive services are contact hand-held device, and can only track device position. And these devices are less natural and intuitive in interacting with virtual object, the controllers input action are significantly different from the action output, so these kinds of handhold controller reducing the immersion of virtual reality game world. Users' previous experience and expectations affect how they interact with your application. In the field of VR, the design of the interactive scheme does not have a standard process. The 2 degrees of freedom of the mouse cannot properly emulate the 3 dimensions of space. The use of hand gestures provides an attractive and natural alternative to these VR interface devices for human computer Interaction [1]. To another hands, a gesture can be defined as a physical movement of the hands, arms, face and body with the intent to convey information or meaning [2].

Kinect is released on November 4, 2010, Kinect provides a new type of human-computer interaction, that is, through the sound and gestures, using contactless control. Initially, the main functionality of the Kinect was to be a tool to the user to interact with Xbox 360 using gestures and spoken commands [3]. Since the sale, Kinect was not only applied in the field of games, but also widely used in the robot, medical and human-computer interaction.

Kinect technology is already used in some touch-free medical applications [4], and education [5][6]. Although there are many studies on Kinect hands tracking, but there is not an complete research about Kinect interaction system, has a detail about how to build up a robust and user-friendly hand cursor system based on Kinect.

Therefore, our aim was to find a robust, comfortable, user-friendly interaction system. Our works make the following contributions:

- We focused on robustly hand cursor interaction coordinate, the coordination called physical interaction zone, it keep stable while transform or unstable when user moving or user body size altering.
- We make up with a complete interaction system, not only the hand tracking, but also include interaction engaged/disengaged detecting, user interaction area, physical interaction zone, which makes the interaction system robust and user friendly.

The paper is structured as follows. First, in Section 2, we will present the related works about Kinect interaction research. Second, in Section 3, we will show our method to build the hand cursor interaction system, it contained two parts, the primary user identification and create physical interactive zone. Third, in Section 4, we will experiment our system. Finally, we will discussion the definite about our interaction system in Section 5.

II. RELATED WORK

The hand and fingertips detection and gesture recognition methods have been studied for several years, and research have found vital application to a wide range of real life time scenarios [7].

Especially many open source code and commercial software have include the hand detecting and tracking application interface [8]. SoftKinetic is a software that can recognition the users' 3D gesture with 3D cameras, it also can tracking users' body joint position by machine learning algorithm [7]. OpenNI is an open source Multilanguage, cross-platform framework that defines an API for writing applications utilizing Natural Interaction. There are four middleware components that

978-1-5386-2937-6/17 $31.00 © 2017 IEEE
DOI 10.1109/ISM.2017.106

processers sensory data, one of them is hand point analysis, which can generates the location of hand point[9].which can be used in hand cursor interaction easily, but their robust and user-friendly are not good.

Some researchers also engage in new tracking algorithm, not only using Kinect sensor, but also the simplex color camera. Srinath increased the tracking robustness and speed of pose estimation by using a single depth camera and a detection-guided optimization strategy [10]. Chen modeled a hand simply using a number of spheres and defined a fast cost function which combines a gradient based and stochastic optimization methods, greatly enhanced the robustness of tracking[11].Zhou propose Finger-Earth Mover's Distance to measure the dissimilarity between hand shapes ,which can better distinguish the hand gestures of slight differences[12].

Most of the researchers work on the application of natural interactive. The application in desktop involves manipulating graphic objects, virtual objects [13]. Vito designed a touchless interactions interface which does not require any activation gestures to trigger actions [14]. Albert provided an alternative, more intuitive approach to input-gesture input compared with keyboard and mouse [15].Norman presented examples of how Kinect-assisted instruction can be used to achieve some of the learning outcomes [16]. Natalia developed a 3D depth sensor ontology, modeled different features regarding user movement and object interaction [17].Moyle and Cockburn use hand cursor to simulate the mouse gestures [18]. Szilvia used vision-based gesture to control mouse moving, user can move the cursor in joystick-like way, and the cursor keeps going to the direction the hand is pointing. And he uses five gesture which indicates Start, Stop, Single Click Event, Double Click Event, Hold Button Down event [19]. Toyin set up with a human computer interaction system using Kinect for Windows, the system can move the mouse cursor by left or right hand position and send the command to PowerPoint presentation by predefined gestures [20]. Computer game is also a main application for hand cursor interaction. Freeman control movement of car by tracking player's body position and hand position, orientation [21].

III. OVERVIEW OF THE SYSTEM

Virtual reality has great levels of increase in recent years, hand cursor for virtual reality attracted many researchers. But the virtual object usually moves fast under large viewpoint variations.

There are many limiting factors influencing the uses of Kinect based interaction system for real time system, such as real-time, robustness, user-friendly. In order to achieve the stability and friendliness of the interaction, we divide the interaction process into two stages. The primary user identification, and Create physical interactive zone.

The main requirement for hand cursor interaction system is the tracking technology used for obtaining the input data. The approach used generally fall into commercial SDK: Kinect for Windows SDK. In Kinect for Windows SDK, it can provide us with information such as skeleton tracking and player tracking for six people [22]. It can also detecting the user face and all users position who are front of sensor. The skeleton tracking features of Kinect combined with the face detection allow us to build a interaction system:

A. The Primary User Identification

According to Microsoft, the Kinect for Windows SDK processes the raw data from the sensor, it can recognize up to six users in its field of view. Each user has an unique id, we can identified user and get their body position by their id.

Static as well as dynamic gestures have been developed by different researchers for many years, and they are preferred depending upon the requirement of the application. So the static and dynamic gestures equally important. Though both the hand gesture recognitions are used in varied applications in the literature, they have their own pros and cons.

The hand gesture recognition consists of detection, tracking and recognition phases.

The Kinect physical limits is during 0.8m to 4m, the extended depth (beyond 4m) can also be retrieved but skeleton and player tracking get noisier the further you get away, and therefore may be unreliable. We set the working area in the center of depth ranges. The working area identified the user tracking area, which the user outside of the area will not be tracked.

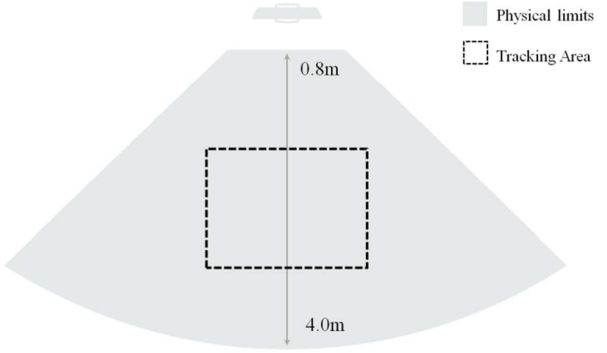

Fig. 1. Physical limits - The actual capabilities of the sensor and what it can see.

When there are multiuser in tracking area, the system must identify which one is going to interact.

While most human-computer interactions, it's easy to know when users mean to interact with the computer, because they deliberately move the mouse, touch the screen. But with Kinect for Windows, it's harder to distinguish between deliberate intent to engage and mere natural movement in front of sensor [22]. We defined status that the alternation from idle to interaction as engaged, on the contrary disengaged. The status alternation is triggered by predefined gestures, such as wave hands, hold hands for a second, or hold hands above the head. The disengaged event is triggered by not only gestures, but also some other behavior, holding down the both hands, moving out of the interaction area, turning back or the face cannot be detected, combining all these information, we can confirm the user interaction status.

The gesture recognition is the most important in status identification. Theoretically the literature classifies hand

gestures into two types, static and dynamic gestures [1]. The static gesture recognition algorithm includes linear classifier, Non-Linear classifier, while dynamic is more complex and has more research works. The recognition algorithm includes Hidden Markov Model, Dynamic Time Warping, Time Delay Neural Network, Finite State Machine, AdaBoost. We use AdaBoost algorithm which also used by official Kinect for Windows SDK to recognize the dynamic gestures.

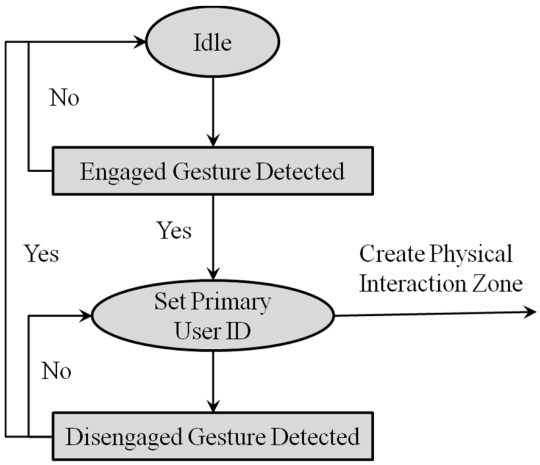

Fig. 2. The primary user identification work flow.

When the primary user is confirmed, the gesture detection will suspend until the primary user free (disengaged) the interaction system.

B. Create Physical Interactive Zone

The next problem is how to transform the 3D hand coordinate in Kinect into 2D cursor coordinate in screen. In this transformation, there are following problems need to be solved:

1) how to structure the base coordinate system in interactive process. During the interactive process, where the user is located ralative to Kinect is always changed, the position of Kinect also always changedunder different circumstances, on the other hand, there are differences ofthe body height betweenusers. In the face of so many complicated cases, the interactive system we designed need to adapt to various situations, the location of hand cursor in screen need to keep steady, and when different users use this interaction system, it can keep robustness and accuracy.

2) how to make interaction system became more comfortable and friendly. Although our arms and hands can move without restriction, there still exist a physical interaction zone(PHIZ) which best fit the movement of arms and hands. Users will not feel uncomfortable in this zone even they operate the device for a long time. We need to find this PHIZ according to left and right hands respectively.

3) how to improve the stability of interaction. We use bones data whichobtained by Kinect to construct interaction system and. As a result of the lack of vision recognition algorithm, bonesdata will become distortion when articulation points were blocked. That will reduce the stability of interaction system. Therefore the slection of articulation is a key to construct a stable interaction system.

For any time t, we define the original bones coordinate which obtained from Kinect to be:

$$C^{(k)}(t) = \left[c_0^{(k)}(t), c_1^{(k)}(t), \cdots, c_n^{(k)}(t)\right]^T \quad (1)$$

where $c_i^{(k)}(t) = \left[x^{(k)}(t), y^{(k)}(t), z^{(k)}(t)\right]^T$ is the coordinate of i-th articulation with respect to t in Kinect coordinate system.

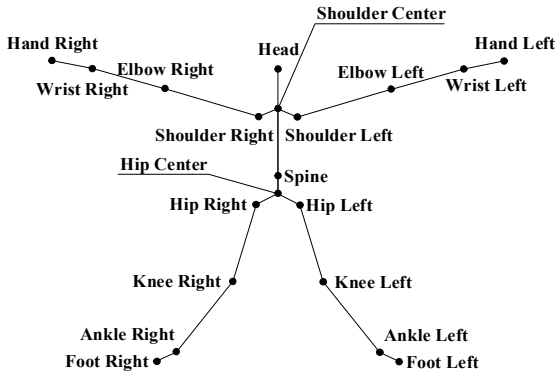

Fig. 3. The original bones data obtained from Kinect.

In order to ensure cursor keep steady in screen when user moved in the process of interaction, we need to build a Self coordinate system which based on user's body, and then we need to map bones coordinate in the Kinect coordinate system on Self coordinate system:

$$C^{(S)}(t) = F\left(C^{(k)}(t)\right) \quad (2)$$

Where $C^{(S)}(t) = \left[c_0^{(S)}(t), c_1^{(S)}(t), \cdots, c_n^{(S)}(t)\right]^T$ is the bones co-ordinate in Self coordinate system. If the hand m relative to the position of the body are unchanged when user moved in the process of interaction, hand coordinate $c_m^{(S)}(t)$ in the Self coordinate system are also unchanged. Therefore the problem is how to construct this Self coordinate system.

At first, we extract some of the most stable point in original bones data(Shoulder Left, Shoulder Right, Shoulder Center, Hip Center, Hip Left, Hip Right), and obtain the Self coordinate system axis direction and origin.

$$S_b = [S_{b1}, S_{b2}] \propto \left[\alpha, \beta, \gamma, x_0^{(s)}, y_0^{(s)}, z_0^{(s)}\right] \quad (3)$$

by regarding these points as basic points.

For axis direction, we describe it as (α, β, γ) in Kinect coordinate system by using Euler angles. The X-Y plane of Self coordinate system is the plane where the user's body placed. This plane is generated by articulation point $c_{sl}^{(k)}(t), c_{sr}^{(k)}(t), c_{hc}^{(k)}(t)$ (Shoulder Left, Shoulder Right, Hip Center). And X direction is from Shoulder Left to Shoulder Right, Y direction and X direction are vertical in the plane where user's body placed, from Hip Center to Shoulder Center. Z direction is vertical with both X and Y direction. As the description above, we can find (α, β, γ).

For origin, origin is in the plane where the user's body placed. Because of the difference between left and right hand when user

operate device, the origin is different. Let us assume the origin when use left hand is:

$$S_{b2_{left}} = \left[x_{l0}^{(k)}, y_{l0}^{(k)}, z_{l0}^{(k)}\right] \quad (4)$$

while the origin when use right hand is:

$$S_{b2_{right}} = \left[x_{r0}^{(k)}, y_{r0}^{(k)}, z_{r0}^{(k)}\right] \quad (5)$$

We have:

$$y_{l0}^{(k)} = y_{r0}^{(k)} = \frac{\frac{y_{sl}^{(k)} + y_{sr}^{(k)}}{2} + y_{hc}^{(k)}}{2} \quad (6)$$

$$x_{l0}^{(k)} = x_{hc}^{(k)} - \frac{\left|x_{hc}^{(k)} - x_{sr}^{(k)}\right|}{2} \quad (7)$$

$$x_{r0}^{(k)} = x_{hc}^{(k)} + \frac{\left|x_{hc}^{(k)} - x_{sl}^{(k)}\right|}{2} \quad (8)$$

$$z_{r0}^{(k)} = z_{l0}^{(k)} = z_{hc}^{(k)} \quad (9)$$

As discussion above, we know the Self coordinate system is always changed, which means it is a function with respect to t:

$$S_b(t) \propto \left[\alpha(t), \beta(t), \gamma(t), x_0^{(s)}(t), y_0^{(s)}(t), z_0^{(s)}(t)\right] \quad (10)$$

After the construction of Self coordinate system, we can transform the coordinate $c_i^{(k)}(t) = \left[x^{(k)}(t), y^{(k)}(t), z^{(k)}(t)\right]^T$ (in the Kinect coordinate system) into the coordinate $c_i^{(S)}(t) = \left[x^{(S)}(t), y^{(S)}(t), z^{(S)}(t)\right]^T$ (in the Self coordinate system) by using the rule:

$$c_i^{(S)}(t) = R\big(\alpha(t), \beta(t), \gamma(t)\big)c_i^{(k)} + = T\left(x_0^{(S)}(t), y_0^{(S)}(t), z_0^{(S)}(t)\right) \quad (11)$$

where R and T represent rotation matrix and translation matrix respectively when Self coordinate system is in the Kinect coordinate system, R is given by:

$$R(t) = R_z\big(\gamma(t)\big)R_y\big(\beta(t)\big)R_x\big(\alpha(t)\big) = \begin{bmatrix} \cos\gamma(t) & \sin\gamma(t) & 0 \\ -\sin\gamma(t) & \cos\gamma(t) & 0 \\ 0 & 0 & 1 \end{bmatrix}\begin{bmatrix} \cos\beta(t) & 0 & \sin\beta(t) \\ 0 & 1 & 0 \\ -\sin\beta(t) & 0 & \cos\beta(t) \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\alpha(t) & \sin\alpha(t) \\ 0 & -\sin\alpha(t) & \cos\alpha(t) \end{bmatrix} \quad (12)$$

while T is given by:

$$T(t) = \begin{bmatrix} x_0^{(S)}(t) & 0 & 0 \\ 0 & y_0^{(S)}(t) & 0 \\ 0 & 0 & z_0^{(S)}(t) \end{bmatrix} \quad (13)$$

Hand coordinate in the Self coordinate system need to be mapped to screen finally. To make sure it can adapt to different size and resolution of screen, we need to normalize the articulation coordinate in the Self coordinate system. We can use:

$$c_i^{(n)}(t) = Nc_i^{(S)}(t) = \begin{bmatrix} \frac{1}{X} & 0 & 0 \\ 0 & \frac{1}{Y} & 0 \\ 0 & 0 & 1 \end{bmatrix} c_i^{(S)}(t) \quad (14)$$

where X and Y are scale factors. User in a state of moving during interacting, so scale factor is a function with respect to time t. We need to find an area which make user feel comfortable and reduce fatigue when they interact with device. This area is so called Physical Interactive Zone (PHIZ) and as shown:

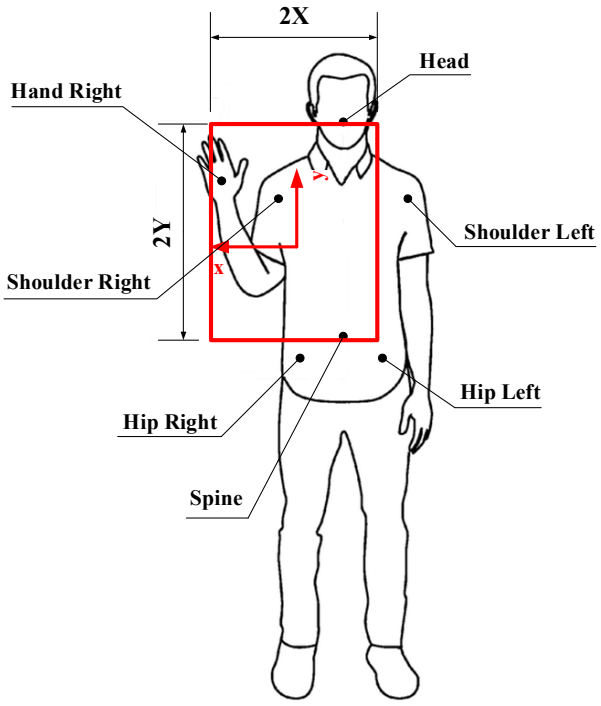

Fig. 4. Physical Interactive Zone.

The width of this area is 2X and the length is 2Y, X and Y are given by:

$$X = \frac{x_a^{(k)} - x_b^{(k)}}{2} \quad (15)$$

$$Y = \frac{y_c^{(k)} - y_d^{(k)}}{2} \quad (16)$$

where a, b, c, d stand for Shoulder Left, Shoulder Right, Shoulder Center, Hip Center respectively.

IV. APPLICATIONS

The interaction application result shows in figure 5.

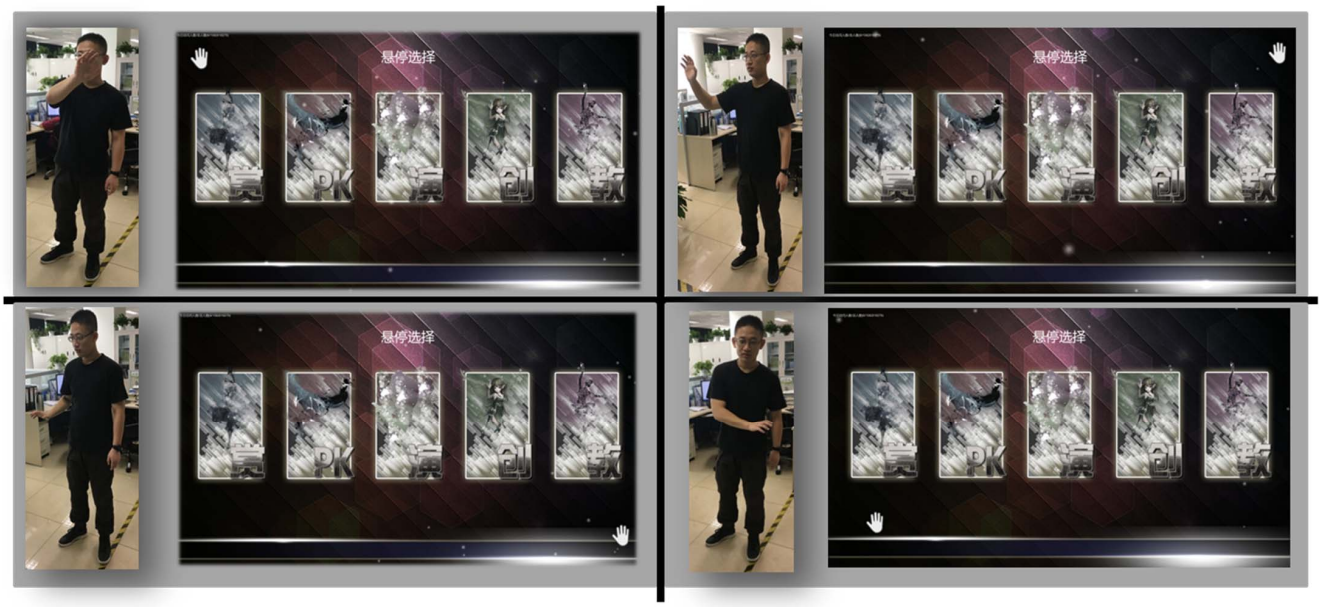

Fig. 5. The application results: four reachable limitation position and corresponding right hand position

Evolution for robustness is a complicated task. As there is no standard baseline algorithms that could accurately define the quantitative robustness of hand cursor interaction. Siddharth defined robustness to three majors: user adaptive, user friendly and gesture independence [2]. Being user adaptive means the system has wide acceptability, independent of the type of user. The gesture implied in system need to be user friendly with high intuitiveness, will not let user feel tired even after long experience. The gesture independent means the system's gesture command mapping the human cognitive behavior.

So we made up an experience test to compare our interactive system with the OpenNI hand cursor application interface, the experiencers do not know which one is our interactive system. We asked experiencers to finish the operation in advance, after the experience, they was required to select a better one. According to the results, most of the experiencers chooses our system to be better.

V. DISCUSSION

At present, our interaction system still has some shortcoming, these shortcomings have restricted the application of interaction system. We used the bones data which obtained from Kinect for Microsoft, so the SDK can only be used in Windows operation system and it is not an open source. In order to improve the application range of the interaction system, we need to use the open source tracking algorithm. Thus we can obtain the spatial data of hand and body from hand recognition results and rebuild the interaction system.

ACKNOWLEDGMENT

We thank the National Key Technology Support Program o f the National '12th Five-Year-Plan of China' under Grant No. 2015BAK25B03 for partially supporting our work. We also thank all colleagues and graduate students who helped us for our visualization system and experiments.

REFERENCES

[1] Rautaray S S, Agrawal A. Vision based hand gesture recognition for human computer interaction: a survey[M]. Kluwer Academic Publishers, 2015.

[2] Mitra S, Acharya T. Gesture Recognition: A Survey[J]. IEEE Transactions on Systems Man & Cybernetics Part C, 2007, 37(3):311-324.

[3] Cruz L, Lucio D, Velho L. Kinect and RGBD Images: Challenges and Applications[C]// Sibgrapi Conference on Graphics, Patterns and Images Tutorials. IEEE Computer Society, 2012:36-49.

[4] Grätzel C, Fong T, Grange S, et al. A non-contact mouse for surgeon-computer interaction[J]. Technology & Health Care Official Journal of the European Society for Engineering & Medicine, 2004, 12(3):245-257.

[5] Meng M, Fallavollita P, Blum T, et al. Kinect for interactive AR anatomy learning[C]// IEEE International Symposium on Mixed and Augmented Reality. IEEE, 2013:277-278.

[6] Villaroman N, Rowe D, Swan B. Teaching natural user interaction using OpenNI and the Microsoft Kinect sensor[C]// Conference on Information Technology Education. ACM, 2011:227-232.

[7] www.softkinetic.com

[8] Li Y. Hand gesture recognition using Kinect[C]// IEEE, International Conference on Software Engineering and Service Science. IEEE, 2012:196-199.

[9] structure.io/openni

[10] Sridhar S, Mueller F, Oulasvirta A, et al. Fast and robust hand tracking using detection-guided optimization[C]// Computer Vision and Pattern Recognition. IEEE, 2015:3213-3221.

[11] Qian C, Sun X, Wei Y, et al. Realtime and Robust Hand Tracking from Depth[C]// IEEE Conference on Computer Vision and Pattern Recognition. IEEE Computer Society, 2014:1106-1113.

[12] Ren Z, Yuan J, Meng J, et al. Robust Part-Based Hand Gesture Recognition Using Kinect Sensor[J]. IEEE Transactions on Multimedia, 2013, 15(5):1110-1120.

[13] Bolt R A, Herranz E. Two-handed gesture in multi-modal natural dialog[C]// ACM Symposium on User Interface Software and Technology. ACM, 1992:7-14.

[14] Gentile V, Sorce S, Malizia A, et al. Touchless Interfaces For Public Displays: Can We Deliver Interface Designers From Introducing Artificial Push Button Gestures?[C]// International Working Conference on Advanced Visual Interfaces. ACM, 2016:40-43.

[15] http://www.albertgural.com/wp-content/uploads/2011/06/project1.pdf

[16] Villaroman N, Rowe D, Swan B. Teaching natural user interaction using OpenNI and the Microsoft Kinect sensor[C]// Conference on Information Technology Education. ACM, 2011:227-232.

[17] Rodríguez N D, Wikström R, Lilius J, et al. Understanding Movement and Interaction: An Ontology for Kinect-Based 3D Depth Sensors[C]// International Conference on Ubiquitous Computing & Ambient Intelligence. 2013:254-261.

[18] Moyle M, Cockburn A. Gesture navigation:an alternative 'back' for the future[C]// CHI '02 Extended Abstracts on Human Factors in Computing Systems. ACM, 2002:822-823.

[19] Szeghalmy S, Zichar M, Fazekas A. Gesture-based computer mouse using Kinect sensor[C]// Cognitive Infocommunications. IEEE, 2015:419-424.

[20] https://www.mendeley.com/research-papers/gesturebased-humancomputerinteraction-using-kinect-windows-mouse-control-powerpoint-presentation/

[21] Freeman W T, Weissman C D. Television Control by Hand Gestures[J]. International Workshop on Automatic Face & Gesture Recognition, 1995:179--183.

[22] www.KinectforWindows.com