Human-computer Interaction using Pointing Gesture based on an Adaptive Virtual Touch Screen

Pan Jing[1] and Guan Ye-peng[1, 2]

1. School of Communication and Information Engineering, Shanghai University
2. Key Laboratory of Advanced Displays and System Application, Ministry of Education
ypguan@shu.edu.cn

Abstract

Among the various hand gestures, pointing gesture is highly intuitive, and does not require any priori assumptions. A major problem for pointing gesture recognition is the difficulty of pointing fingertip tracking and the unreliability of the direction estimation. A novel real-time method is developed for pointing gesture recognition using Kinect based depth image and skeletal points tracking. An adaptive virtual touch screen is constructed instead of estimating pointing direction. When a user stands in a certain distance from a large screen to perform pointing behaviors, he interacts with the virtual touch screen as if it is just right in front of him. The proposed method is suitable for both large and small pointing gestures, and it's not subject to users' characteristics and environmental changes. Experiments have highlighted that the proposed approach is robust and efficient to realize human-computer interaction based on pointing gesture recognition by comparisons.

Keywords*: Pointing gesture; Fingertip tracking; HCI; Kinect; Virtual touch screen*

1. Introduction

Human-computer interaction (HCI) is a hot topic in the information technology age. Operators can provide commands and get results with the interactive computing system involves one or more interfaces. With the development of graphical operator interfaces, operators with varying levels of computer skills have been allowed to use a wide variety of software applications. Recent advancements in HCI provide more intuitive and natural ways to interact with the computing devices.

There are many HCI patterns including facial expression, body posture, hand gesture, speech recognition and so on. Among them hand gesture is an intuitive and easy learning means. Wang *et al.*, [1] proposed an automatic system that executes hand gesture spotting and recognition simultaneously based on Hidden Markov Models (HMM). Suk *et al.*, [2] proposed a new method for recognizing hand gestures in a continuous video stream using a dynamic Bayesian network or DBN model. Van den Bergh *et al.*, [3] introduced a real-time hand gesture interaction system based on a Time-of-Flight. Due to the diversity of hand gesture, there exist some limitations for recognition. Comparatively, pointing gesture – the simplest hand gesture which can be easier to be recognized has attracted much attention. Park and Lee [4] presented a 3D pointing gesture recognition algorithm based on a cascade hidden Markov model (HMM) and a particle filter for mobile robots. Kehl and Gool [5] proposed a multi-view method to estimate 3D directions of one or both arms. Michael [6] has set up two orthogonal cameras from the top view and side view to detect hand regions, track the finger pointing features, and estimate the pointing directions in 3D space. Most previous pointing gesture recognition methods use the results of face and hand tracking [7] to recognize the pointing direction. However, the recognition rate is limited by the unreliable face and hand detection and tracking in 3D space. Another difficult problem is to recognize some small pointing gestures which usually results in the wrong direction.

Body tracking or skeleton tracking techniques using an ordinary camera are not easy and require extensive time in developing. From the body or skeleton tracking survey, to detect the bone joints of human body is still a major problem since the depth of the human cannot be determined through the use of a typical camera. However, some researchers have tried to use more than one video camera to detect and determine the depth of the object, but the consequence is that the cost increases and the processing ability slows down due to the increased data processing. Fortunately, the Kinect sensor makes it possible to acknowledge the depth and skeleton of operators. The Kinect has three autofocus cameras including two infrared cameras optimized for depth detection and one standard visual-spectrum camera used for visual recognition.

Take a recognition system robust and the cost of cameras into consideration, some researchers have used Kinect sensor to provide a higher resolution at a lower cost instead. The Microsoft Kinect sensor combines depth and RGB cameras in one compact sensor [8, 9]. It's robust to all colors of clothing and background noise and provides an easy way for real-time operator interaction [10-14]. There are some freely available libraries that can be used to collect and process data from a Kinect sensor as well as skeletal tracking [15, 16].

Two main contributions are as follows in this paper. Firstly a new method to detect pointing fingertip is proposed which is used to recognize pointing gestures and interact with large screen instead of head–hand line. Secondly a scalable and flexible virtual touch screen is constructed which is adaptively adjusted as the operator moves as well as pointing arm extends or contracts. Experimental results have shown that the developed method is robust and efficient to recognize pointing gesture and realize HCI by comparisons.

The organization of the rest paper is as follows. In Section 2, pointing gesture recognition method is described. How to construct a virtual touch screen is introduced in Section 3. Experimental results are given in Section 4, and some appropriate conclusions are made in Section 5.

2. Pointing Gesture Recognition

The input to the proposed method is a depth image acquired by the Kinect sensor, together with its accompanying skeletal map. When there are some people standing in front of the Kinect, the most important thing is how to determine the real operator. The horizontal view is limited in a certain range, so the one who stands out of the range will be ignored. A user standing closest to Kinect sensor is deemed as the operator, who will be tracked until he is obscured by another one or leaves the scene.

2.1. Pointing Hand Segmentation

Huma hand tracking and locating is carried out by continuously processing RGB images and depth images of an operator who is performing the pointing behavior to interact with large screen. The RGB images and depth images are captured by the Kinect sensor which is fixed in front of the operator. Microsoft Kinect has the ability to track the movements of 24 distinct skeletal joints on the human body, wherein head, hand and elbow points have been used in this experiment.

When operator held out the hand to perform pointing gesture, the hand is closer than the other one. The 3D coordinates of pointing hand joint can be obtained from skeletal map.

$$H_p = \begin{cases} H_r, & z_r < z_l \\ H_l, & else \end{cases} \tag{1}$$

Where H_p is the pointing hand, H_r refers to right hand and H_l refers to left hand, z_r and z_l present the z coordinates of right and left hand respectively.

In order to catch the hand motion used for controlling the large screen, it is need to separate the hand from the depth image. A depth image records the depth of all the pixels of the RGB image. The depth value of hand joint can be figured out through skeleton-to-depth conversion, and it is taken as a segmentation threshold which is used to divide the hand region $H_d(i, j)$ from the raw depth image.

$$H_d(i, j) = \begin{cases} 255, & d(i, j) < T_d + \varepsilon \\ 0, & else \end{cases}, d(i, j) \in D; i = 1,2,\cdots,m; j = 1,2,\cdots n \tag{2}$$

Where $d(i, j)$ is the pixel of depth image D, T_d is the depth of the pointing hand's joint, ε refers to a small range around threshold T_d, m is the width of D, and n is the height of D.

2.2. Pointing Fingertip Detection

Most of existing methods for pointing gesture recognition use the operator hand-arm motion to determine the pointing direction. To further study hand gestures, some approaches for fingertips detection have been proposed. In this paper, pointing fingertip [17-19] with more precise is detected instead of hand to interact with large screen. The hand's minimum bounding rectangle makes it easy to extract the pointing fingertip. The rules for pointing fingertip detection especially index fingertip are as follows:

(1) Extract the tracked hand region using the minimum bounding rectangle by skeletal information as well as the corresponding elbow.

(2) When the operator's hand is pointing to left, his or her pointing hand joint's coordinate in the horizontal direction (x coordinate) is less than the corresponding elbow's.

At this moment, if the hand joint's coordinate in vertical direction (y coordinate) is also less than elbow's and the width of minimum bounding rectangle is less than its height, it's not difficult to find that index fingertip moves along the bottom edge of bounding rectangle;

If the pointing hand joint's x coordinate is larger than the corresponding elbow's and the width of the minimum bounding rectangle is less than its height, the index fingertip moves along the up edge of bounding rectangle; If the minimum's width is larger than its height, the index fingertip moves along the left edge of bounding rectangle.

(3) When the operator's hand is pointing to right, his or her pointing hand joint's x is larger than that of the corresponding elbow's one, the distinguishing rule is the same as step (2).

The distinguishing rules above are to find the pointing fingertip in depth image, and its 3D coordinates can be figured out through depth-to-skeleton conversion.

2.3. Pointing Fingertip Tracking

In some cases, the fingertip will be detected by mistaken due to the surrounding environmental interference. To avoid false detection, a tracking technique is introduced to track the feature points. A Kalman filter [20-22] is used to record the detected fingertip motion trajectory, the features of pointing gestures (including fingertip's position and speed) can be extracted through this tracking method.

Motion of pointing fingertip can be described by a linear dynamic model consisting of a state vector $x(k)$ and a state transition matrix $\Phi(k)$. The state vector containing the position, velocity in all three dimensions is described as (3).

$$x(k) = [x, y, z, v_x, v_y, v_z] \quad (3)$$

where x, y, z presents the image coordinates of the detected fingertip, and v_x, v_y, v_z presents its displacement.

The Kalman filter model assumes the true state at time k is evolved from the state at $(k-1)$ according to (4).

$$x(k) = \Phi(k)x(k\text{-}1) + W(k) \quad (4)$$

where $\Phi(k)$ is the state transition model which is applied to the previous state $x(k-1)$;$W(k)$ is the process noise which is assumed to be drawn from a white Gaussian noise process with covariance $Q(k)$.

At time k an observation (or measurement) $z(k)$ of the true state $x(k)$ is made according to (5).

$$z(k) = H(k)z(k) + V(k) \quad (5)$$

where $H(k)$ is the observation model which maps the true state space into the observed space and $V(k)$ is the observation noise which is assumed to be zero mean Gaussian white noise with covariance $R(k)$.

In our method, the Kalman filter is initialized with six states and three measurements, the measurements correspond directly with x, y, z in the state vector.

2.4. Pointing Gestures Recognition

Human motion is a continuous sequence of actions or gestures and non-gestures without clear-cut boundaries. Gestures recognition refers to detecting and extracting meaningful gestures from an input video. It is crucial how to recognize pointing gesture since the recognition procedure is only performed for the detected pointing gestures. When a person makes a pointing gesture, the whole motion can be separated into three phases including non-gesture (hands and arms drop naturally, it’s unnecessary to find fingertip), move-hand (the pointing hand is moving and its direction is changing), point-to (the hand is approximately stationary). Among the three phases, only the point-to phase or the corresponding pointing gesture is relevant to target selection.

To recognize pointing gestures, the three phases must be distinguished. When operator moves his hand, the velocity at time t is estimated by (6)

$$\mathrm{v}_x = x_t - x_{t-1}, \mathrm{v}_y = y_t - y_{t-1}, \mathrm{v}_z = z_t - z_{t-1} \quad (6)$$

where x, y, z represents the position of pointing fingertip in 3D space, respectively. If v_x, v_y and v_z maintains in a small specific range v_T, it is approximately stationary. Assuming this stationary state lasts for n frames, the initial value of n is 0

$$n = \begin{cases} n+1, & |v_x| < \mathrm{v}_T \ \& \ |v_y| < v_T \ \& \ |v_z| < v_T \\ 0, & \text{else} \end{cases} \quad (7)$$

When n reaches a certain amount, it presents the operator is pointing at the interaction target, & is logical AND operation.

3. Virtual Touch Screen Construction

3.1. Kinect Coordinate System

When an operator stands in front of the Kinect sensor and interacts with a large screen. The Kinect coordinate is defined as following shown in Figure 1. Axis *X* is upturned, axis *Y* is rightward and axis *Z* is vertical. The Kinect can capture the depth of any objects in its workspace.

3.2. Virtual Touch Screen

Many researchers pay much attention to human body skeleton, so hand-arm or hand-eye is always used to determine the pointing direction. A virtual touch screen is constructed for human computer interaction, which does not need estimate pointing direction.

The concept of virtual touch screen was introduced in [23]. The distance between the operator and the virtual touch screen remains unchanged namely the arm's length is assumed to be constant. Apparently, it's not suitable for different operators. A virtual touch screen is constructed in this paper which is scalable and flexible. Define z coordinate of the virtual screen as z one of user's pointing fingertip. It will move forward or backward as the pointing arm extends or contracts the screen which is more natural and suitable for different operators.

The 3D coordinates of head and pointing index fingertip are used to construct a virtual touch screen as Figure 1. One can note that the large screen and Kinect are in the same plane, which means its z coordinate is zero. Assume the large screen is divided into $m \times n$ modules, and the x coordinates of vertical demarcation points are x_1, x_2, … x_m, the y coordinates of the horizontal demarcation points are y_1, y_2, …, y_n. Suppose the corresponding values on virtual screen are x'_i, y'_j, i=1, 2, …, m, j=1, 2, …, n.

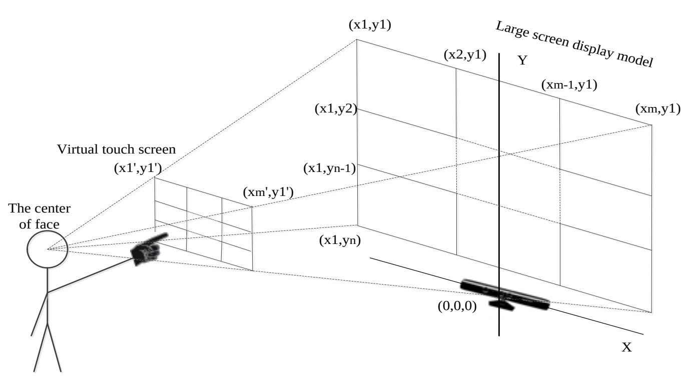


Figure 1. The Virtual Touch Screen

$$\frac{x_h - x_i^{'}}{x_h - x_i} = \frac{y_h - y_{\mathrm{j}}^{'}}{\mathrm{y_h - x_j}} = \frac{z_h - z^{'}}{z_h - z} \tag{8}$$

where z is 0, and z' equals to the pointing fingertip's z coordinate, x_h, y_h and z_h refer to the 3D coordinates of operator's head joint.

The corresponding coordinates on virtual touch screen can be figured out as (9), (10),

$$x_i^{'} = \frac{(x_h - x_i)\times(z_f - z_h)}{z_h} + z_h \tag{9}$$

$$y_j^{'} = \frac{(y_h - y_j) \times (z_f - z_h)}{z_h} + y_h \tag{10}$$

The virtual touch screen will move forward or backward with movement of the pointing fingertip as well as human motion. If operator's pointing fingertip stays in one block of virtual screen for a while, the corresponding block on the large screen will be triggered.

4. Experimental Results and Analysis

The experimental evaluation is based on real-world sequences obtained by a Kinect sensor. Kinect for Xbox is compatible with Microsoft Windows 7. Using the OpenNI the system is developed in Microsoft Visual Studio 2010 with C++ and OpenCV. The proposed method runs on a computer equipped with an Intel core i3- 2120 CPU, 4 GBs RAM. On this system, the average frame rate is 30Hz.

4.1. Experimental Environment

The real experimental environment is show in Figure 2. We draw a piece of area on the wall and divide it into six modules which are taken as a large screen, respectively. In order to make the experimental results intuitive, six lamps are fixed on the center of each module. The pointing lamp turns on or off is used to test the accuracy of pointing gesture recognition.

The distance between the operator and Kinect sensor within 0.8m to 3.5m, the large screen's width is 2.4m and its height is 1.3m. The demarcation points (x_1, y_1), …, (x_6, y_6) can be measured manually. Pointing gestures are recorded from seven volunteers with different heights and postures.

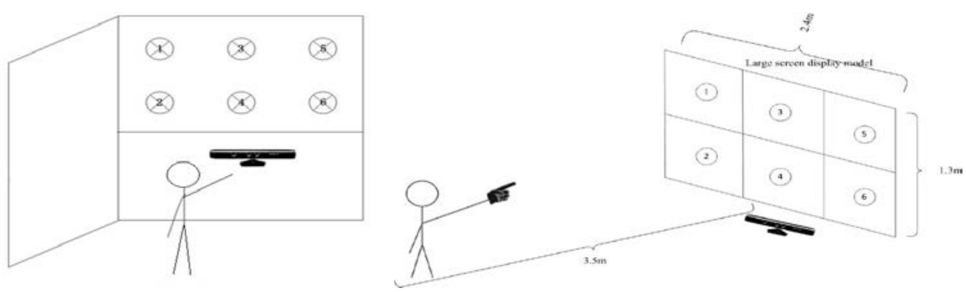


Figure 2. Experimental Environment

4.2. Recognition of Pointing Gestures

Accuracy pointing hand detection is a premise condition for pointing fingertip extraction. The 3D coordinates of hand joint can be obtained from skeletal image, and its position in depth image can be figured out through skeleton-to-depth conversion. The depth value of hand joint is taken as a threshold T_d, and the hand region could be extracted from depth image according to (2). Through some tests, defining ε as 20 is enough to segment the pointing hand completely.

Some non-hand regions detected by mistaken and the false detection blocks which doesn't contains pointing hand joint must be removed as following.

$$H_R = \{R \mid R \supset J_H\} \tag{11}$$

Where R refers to the separable regions by threshold segmentation, J_H is the pointing hand joint.

The first row of Figure 3 shows RGB image and its corresponding depth image, the second row shows detected hand region.

The pointing fingertip is extracted with the method described in Section 2.2 shown in Figure 4. The position of pointing fingertip in 3D space can be figured out through depth-to-skeleton conversion.

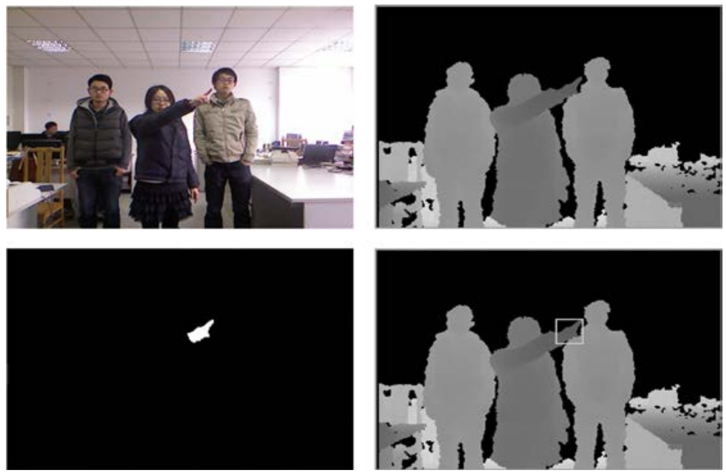


Figure 3. Hand Segmentation

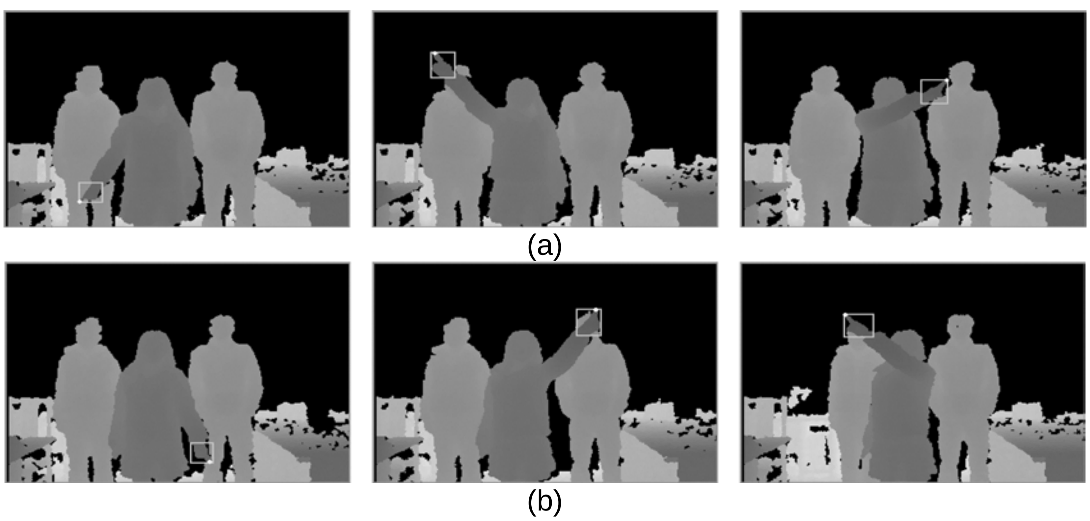
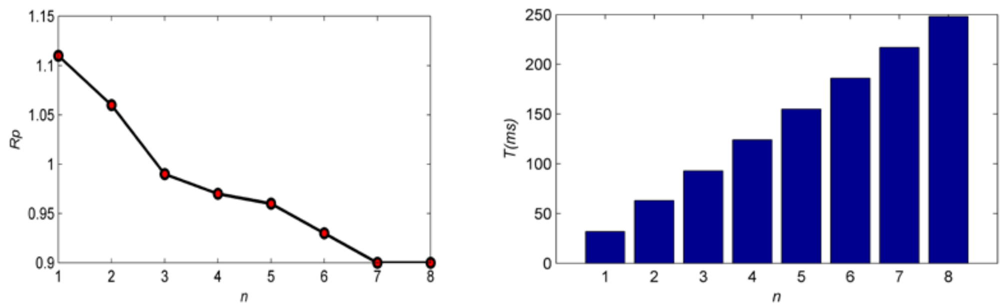


Figure 4. Pointing Fingertip Detection with Different Pointing Gestures. (a) Right Hand is Performing Pointing Gesture. (b) Left Hand is the Pointing Hand

As described in Section 2.4, (6) and (7) are used to determinate if the operator performs pointing gesture. When the operator is pointing at one lamp, his or her fingertip is approximately stationary. According to some statistics, hand trembling range is within 0 to 4mm. In addition, a lot of experiments have been done to choose a favorable *n* for effective pointing gesture recognition.

$$R_p = \frac{M_r}{M_a} \tag{12}$$

Where R_p presents the rate of pointing gesture recognition, M_r refers to the frame number of recognized pointing gesture and M_a refers to the actual pointing frame number.



(a) (b)

Figure 5. Pointing Gesture Recognition Results. (a) Pointing Gesture Recognition Rates with Different *n*. (b) Executive Time with Different *n*

It could be seen from Figure 5(a) that defining V_T as 4mm is applicable. When n is less than three, it's overdone, When n is larger than three, some pointing gestures haven't been recognized. By analysis the n is elected as 3, program running time is about 98ms as shown in Figure 5(b), which satisfies the real-time requirements. If the displacements of pointing fingertip are within 0 to 4mm in continuous 3 frames, which means that the operator is pointing at one target.

4.3. Target Selection

Traditionally, most existing methods for target selection use head-hand line to estimate the position of pointing target. In this paper, when pointing gesture is recognized, a scalable and removable virtual touch screen is constructed. If the pointing fingertip locates at one module of the virtual screen, the corresponding lamp on the wall will turn on. Figure 6 is a simulation image, which presents that when operator's fingertip touches the module 2 on virtual screen, the corresponding number 2 lamp on the large screen turns on.

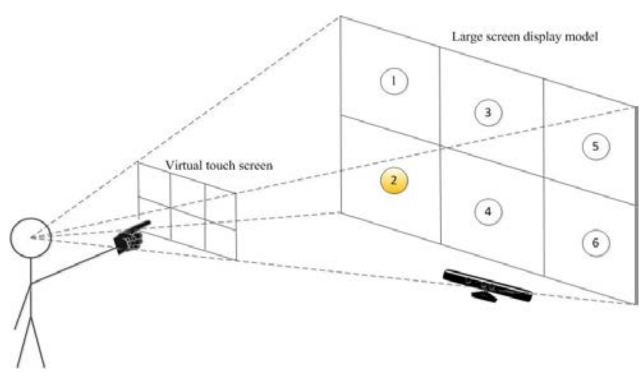


Figure 6. Experimental Results of Target Selection

$$R_c = \frac{N_c}{N_p} \times 100\% \tag{13}$$

Where R_c refers to the correct target selection rate, N_p is the times of pointing to one target and N_c is the times of correct target selection.

In the progress of the experiment, we find that sometimes the pointing fingertip is detected by mistaken, which directly affects the results of target selection. So as described in Section 2.3, a Kalman filter is applied to record the pointing fingertip motion trajectory to decrease the false detection.

Experimental results with Kalman filter or not are given in Table 1, Table 2 respectively. Confusion matrix is used to summarize the results of target selection. Each column of the matrix represents the recognized target, each row represents the actual targets, and the data represents the average rates of correct target selection. L_1, L_2, L_3, L_4, L_5, L_6 refer to the six lamps fixed on the wall as shown in Figure 6, respectively.

Table 1. Results of Target Selection without Kalman Filter

	*L*1	*L*2	*L*3	*L*4	*L*5	*L*6
*L*1	**96.9%**	3.1%	0	0	0	0
*L*2	2.2%	**97.8%**	0	0	0	0
*L*3	0	0	**99.0%**	1.0%	0	0
*L*4	0	0	0.4%	**99.6%**	0	0
*L*5	0	0	0	0	**96.9%**	3.1%
*L*6	0	0	0	0	2.2%	**97.8%**

Table 2. Results of Target Selection with Kalman Filter

	*L*1	*L*2	*L*3	*L*4	*L*5	*L*6
*L*1	**98.4%**	1.6%	0	0	0	0
*L*2	0.5%	**99.5%**	0	0	0	0
*L*3	0	0	**99.6%**	0.4%	0	0
*L*4	0	0	0	**100%**	0	0
*L*5	0	0	0	0	**98.4%**	1.6%
*L*6	0	0	0	0	0.5%	**99.5%**

To further analyze the proposed method, another six volunteers perform HCI based on pointing gesture to control the lamps on the wall. The performance of proposed method is compared with that of methods developed by Yamamoto *et al.*, in [7] and Cheng in [23], respectively. The comparison result is given in Table 3. One can note that the proposed method is best by comparison from Table 3.

Table 3. Comparisons of Correct Target Selection Rate in Different Methods

R_c (%)			
	Method in [7]	Method in [23]	**Proposal**
Average R_c (%)	90.1%	94.8%	**99.2%**

5. Conclusions

A new 3D real-time method is developed for HCI based on pointing gesture recognition. The proposed method involves the use of a Kinect sensor and a flexible virtual touch screen for interacting with a large screen. This method is free from illumination and surrounding environmental changes using the depth and skeletal map generated by Kinect sensor. In addition, the target selection is only relevant to operator's pointing fingertip position, so it's suitable for both large and small pointing gestures as well as different operators.

Due to the widely application of speech recognition in HCI, our further work will focus on the combination of speech and visual-multimodal features to improve the recognition rate.

Acknowledges

This work is supported in part by the Natural Science Foundation of China (Grant No. 11176016, 60872117), and Specialized Research Fund for the Doctoral Program of Higher Education (Grant No. 20123108110014).

Reference

[1] M. Elmezain, A. Al-Hamadi and B. Michaelis, "Hand trajectory-based gesture spotting and recognition using HMM", Proceedings of International Conference on Image Processing, **(2009)**, pp. 3577-3580.

[2] H.-I Suk, B.-K. Sin and S.-W. Lee, "Hand gesture recognition based on dynamic Bayesian network framework", Pattern Recognition, vol. 9, no. 43, **(2010)**, pp. 3059-3072.

[3] M. Van Den Bergh and L. Van Gool, "Combining RGB and ToF cameras for real-time 3D hand gesture interaction", Proceedings of 2011 IEEE Workshop on Applications of Computer Vision, **(2011)**, pp. 66-72.

[4] C.-B. Park and S.-W. Lee, "Real-time 3D pointing gesture recognition for mobile robots with cascade HMM and particle filter", Image and Vision Computing, vol. 1, no. 29, **(2011)**, pp. 51-63.

[5] R. Kehl and L. V. Gool, "Real-time pointing gesture recognition for an immersive environment", Proceedings of 6th IEEE International Conference on Automatic Face and Gesture Recognition, **(2004)**, pp. 577-582.

[6] J. R. Michael, C. Shaun and J. Y. Li, "A multi-gesture interaction system using a 3-D iris disk model for gaze estimation and an active appearance model for 3-D hand pointing", IEEE Transactions on Multimedia, vol. 3, no. 13, **(2011)**, pp. 474-486.

[7] Y. Yamamoto, I. Yoda and K. Sakaue, "Arm-pointing gesture interface using surrounded stereo cameras system", Proceedings of International Conference on Pattern Recognition, vol. 4, **(2004)**, pp. 965-970.

[8] M. Van Den Bergh, D. Carton and R. De Nijs, "Real-time 3D hand gesture interaction with a robot for understanding directions from humans", Proceedings of IEEE International Workshop on Robot and Human Interactive Communication, **(2011)**, pp. 357-362.

[9] Y. Li, "Hand gesture recognition using Kinect", Proceedings of IEEE 3rd International Conference on Software Engineering and Service Science, **(2012)**, pp.196-199.

[10] M. Raj, S. H. Creem-Regehr, J. K. Stefanucci and W. B. Thompson, "Kinect based 3D object manipulation on a desktop display", Proceedings of ACM Symposium on Applied Perception, **(2012)**, pp. 99-102.

[11] A. Sanna, F. Lamberti, G. Paravati, E. A. Henao Ramirez and F. Manuri, "A Kinect-based natural interface for quadrotor control", Lecture Notes of the Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering, 78 LNICST, **(2012)**, pp. 48-56.

[12] N. Villaroman, D. Rowe and B. Swan, "Teaching natural operator interaction using OpenNI and the Microsoft Kinect sensor", Proceedings of ACM Special Interest Group for Information Technology Education Conference, **(2011)**, pp. 227-231.

[13] S. Lang, M. Block and R. Rojas, "Sign language recognition using Kinect", Lecture Notes in Computer Science, **(2012)**, pp. 394-402.

[14] L. Cheng, Q. Sun, H. Su, Y. Cong and S. Zhao, "Design and implementation of human-robot interactive demonstration system based on Kinect", Proceedings of 24th Chinese Control and Decision Conference, **(2012)**, pp. 971-975.

[15] J. Ekelmann and B. Butka, "Kinect controlled electro-mechanical skeleton", Proceedings of IEEE Southeast Conference, **(2012)**, pp. 1-5.

[16] C. Sinthanayothin, N. Wongwaen and W. Bholsithi, "Skeleton tracking using Kinect sensor and displaying in 3D virtual scene", International Journal of Advancements in Computing Technology, vol. 11, no. 4, **(2012)**, pp. 213-223.

[17] G. L. Du, P. Zhang, J. H. Mai and Z .L. Li, "Markerless Kinect-based hand tracking for robot teleoperation", International Journal of Advanced Robotic Systems, vol. 1, no. 9, **(2012)**, pp. 1-10.

[18] J. L. Raheja, A. Chaudhary and K. Singal, "Tracking of fingertips and centers of palm using Kinect", Proceedings of 3rd International Conference on Computational Intelligence, Modeling and Simulation, **(2011)**, pp. 248-252.

[19] N. Miyata, K. Yamaguchi and Y. Maeda, "Measuring and modeling active maximum fingertip forces of a human index finger", Proceedings of IEEE International Conference on Intelligent Robots and Systems, **(2007)**, pp. 2156-2161.
[20] N. Li, L. Liu and D. Xu, "Corner feature based object tracking using adaptive Kalman filter", Proceedings of International Conference on Signal, **(2008)**, pp. 1432-1435.
[21] M. S. Benning, M. McGuire and P. Driessen, "Improved position tracking of a 3D gesture-based musical controller using a Kalman filter", Proceedings of 7th International Conference on New Interfaces for Musical Expression, **(2007)**, pp. 334-337.
[22] X. Li, T. Zhang, X. Shen and J. Sun, "Object tracking using an adaptive Kalman filter combined with mean shift", Optical Engineering, vol. 2, no. 49, **(2010)**, pp. 020503-1-020503-3.
[23] K. Cheng and M. Takatsuka, "Estimating virtual touch screen for fingertip interaction with large displays", Proceedings of ACM International Conference Proceeding Series, **(2006)**, pp. 397-400.