

The Usability of Hand Pose and Gestures for Spaceflight Training System

Wanhong Lin^{1,2}(✉), Jiangan Chao^{1,2}, Jin Yang², and Ying Xiong²

¹ National Key Laboratory of Human Factors Engineering,
China Astronaut Research and Training Center,
Beijing 100094, People's Republic of China
a.c.c.l.w.h@hotmail.com, 85365060@qq.com

² China Astronaut Research and Training Center,
Beijing 100094, People's Republic of China

Abstract. Virtual hand pose and gestures for manipulation is one of the hardest interactive problem in Spaceflight Training System. This paper researched the usability of hand pose and gestures, proposed a couple of non-contact action determining rules and an action states transition method based on finite state machine (FSM) to control the manipulation process of virtual hand. Meanwhile, a gesture recognition method based on hand pose data was proposed. The experiment result shows that the proposed method has a high success rate and efficiency for manipulation. It can be used for the way of manipulation in Spaceflight Training System.

Keywords: Usability · Hand pose and gestures · Spaceflight training system

1 Introduction

With the rapid development of computer technology in the field of spaceflight training, Augmented and Virtual Reality Environments arise and become part of training system. The role of human computer interaction with these system, or HCI, is becoming more important. In particular, an approach that involves adapting the way humans communicate with each other for HCI is considered to be the most promising. Potential spaceflight training systems which use these new technologies require input devices that can be easily carried along with the user and instantly available when needed (or made available by the environment). Such technologies pose several challenges on the GUI (Graphical User Interface) paradigm designed for desktop interaction, particularly on current input devices such as keyboards and mouse [1–4].

Especially, spaceflight training systems inherently requires controls with a high degree of freedom. For instance, manipulation of a 3D object with using user's hand directly rather than with a mouse can offer an ideal alternative and more comfortable and easier way than using mouse and keyboard for such systems [3–5]. In this way, users can control the position and orientation of a 3D object directly by simply moving their hands.

Although wearable devices might be an option, a computer vision based gesture recognition system seems to be a more natural alternative since it can be considered as part of the environment and operates remotely, it does not require the user to wear or have any physical contact with a device. A gesture based interface has the potential of eliminating the need of pointing devices, thus saving the time and effort in using such devices for the interaction process. As a result, the interaction with AR/VR system based on gestures motivated us to develop a method for tracking a user's hand in 3D and recognizing hand gestures in real-time without using any invasive devices attached to the hand [5]. In this work, referring to the previous method [2] we proposed a new technique for estimating the 3D gesture based on the 3D pose of a user's hand by using depth camera in real-time. In addition, the proposed technique is able to recognize predetermined hand gestures in a fast and robust manner by using Machine Learning which has been properly trained beforehand. Meanwhile, because performing gestures for long periods of time can be physically stressing, the understanding of how gestures can be more effectively and more comfortably used to communicate with training systems becomes more important.

2 Hand Pose and Gestures Lexicon

In terms of hand pose, the user uses a virtual hand based on the data of hand pose to manipulate the virtual object as human do in the reality world. Hand gestures are different. The virtual system captures hand movements and then hand movements will be recognized as corresponding gestures that can generate command to control virtual objects or 3D menu.

In the previous study, we found that the system needs to carry out the physical collision detection between virtual hand and virtual objects when using hand pose to manipulate the virtual object. It is well known that physical collision detection is a very complicated process. In this paper, we will merge these two modes to make an interface with the virtual reality system.

In term of object manipulation, there are several kinds of actions that need to be carried out by the astronaut in the training system:

- Push the button on the menu: The astronaut usually need information which is triggered by buttons on the 3D menu to control the training process.
- Select: Select the object.
- Grab: Grab the object.
- Rotate: Rotate the object.
- Move: Move the object.
- Release: Release the manipulated object.

So the commands generated by hand pose and gestures includes Push, Select, Grab, Rotate, Move and Release. According to our study, we defined the hand pose and gestures as following Table 1.

Table 1. Hand pose and gestures Lexicon

Command	Hand pose and gestures
Push	Palm push forward after selected
Select	Use index finger to point
Grab	Thumb and index or middle fingertips come together
Rotate	Point forward using index and middle fingers, and then cycle the hand.
Move	After grab, movement of object is determined by absolute location of center of hand
Release	Thumb and index-middle fingertips separate

3 Action States Transition Based on FSM

When the astronaut is training, those actions are continuous. So those actions need to be distinguished based on the state of the different movement. According to the special virtual weightless environment of astronaut virtual training, this paper proposed a couple of non-contact action determining rules and an action state transition method based on finite state machine (FSM). Finite state machine (FSM) [6, 7] is a good tool for states control of the virtual hand action. The interactive model of FSM for virtual hand action is shown in Fig. 1, where S represents the state, c represents the states transition conditions.

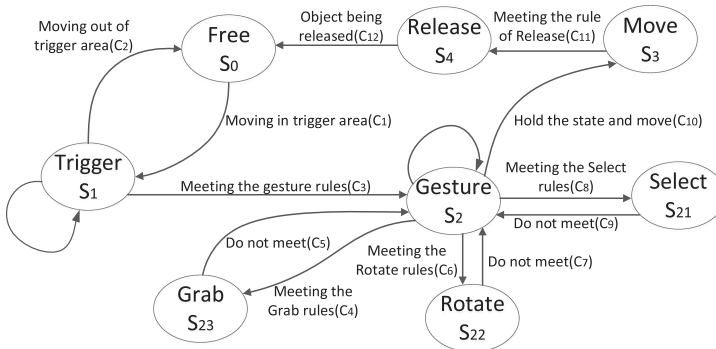


Fig. 1. The finite state machine of action

As our previous research results [8], we proposed some action determining rules which are a kind of rules that used to determine whether a virtual hand manipulates the object in virtual training world.

Because all of the objects in the training system are weightless, a slight touch from virtual hand would lead to the movement of objects, which will result in failure to manipulate. Traditional determining rules based on collision detection are not suitable for weightless virtual environment. We developed two non-contact determining rules based on hand pose and gestures.

Rule One: the distance between the object and the virtual hand must be less than a certain critical value.

Rule Two: the character value of the hand gesture is must be less than a certain critical value, which is called the object action trigger area.

It is determined that action is successful only if the virtual hand and the object meet the condition of Rule One and Rule Two at the same time. After successful action, the object is attached to the virtual hand and moves with it together. The object will be released when the virtual hand does not satisfy Rule Two.

According to the principle of FSM, the constituent elements of FSM are defined as: states (S), input events (X), output (Y), states transition function (f) and output function (g).

States (S): action states of virtual hand. There are five states, Free State, Trigger State, Gesture State, Move State and Release State. Each state corresponds to a kind of action state between the virtual hand and the object.

Input events (X): correspond to the five states shown in Fig. 1. There are six conditions which are the virtual hand moving into the object action trigger area, the virtual hand moving out of the object action trigger area, the virtual hand meeting the rules of gesture, the virtual hand meeting the rules of move, the virtual hand meeting the condition of release and the object being released.

Output (Y): the rendering results displayed to the operator.

States transition function (f): defined as the determination function which determines the states transition from the current state to the next one. Equation (1) shows its relationship with the states and time, where $X(t) \in X, S(t) \in S$.

$$S(t+1) = f(X(t), S(t)) \quad (1)$$

Output function (g): defined as a mapping relationship between the current state and the output. Equation (2) shows its relationship with the states, where $Y(t) \in Y$.

$$Y(t) = g(X(t), S(t)) \quad (2)$$

Free State (S_0): the virtual hand does not touch any object. In this condition, virtual hand can move freely and the finger joints can bend freely.

Trigger State (S_1): the virtual hand moves into the object action trigger area, but does not meet the action determining rules. At this time, the virtual hand does not touch the object, the object does not be manipulated.

Gesture State (S_2): the virtual hand can manipulate the object stably by following the gesture determining rules. The object is adsorbed to virtual hand in the gesture state.

Move State (S_3): the virtual hand can manipulate the object stably by following the two determining rules. The object is adsorbed to virtual hand moving with the virtual hand together as a child node.

Release State (S_4): the virtual hand converts to the release state when it does not satisfy Rule Two of the action determining rules after manipulating the object. In this state, the virtual hand releases the object and then converts to the free state.

4 The Gesture Recognition Method

There are many devices provide the data of hand pose such as Intel RealSense, Leap Motion, Kinect etc. Due to the high accuracy of Leap Motion [9], we choose the pose data provided by the Leap Motion for gesture recognition.

4.1 Gesture Recognition Algorithm

As already reported the Leap Motion device provides only a limited set of relevant points and not a complete description of the hand shape [10]. But the device provides directly some of the most relevant points for gesture recognition and allows to avoid complex computations needed for their extraction from depth and color data. The Leap Motion sensor mainly provides the following data, as depicted in Fig. 2:

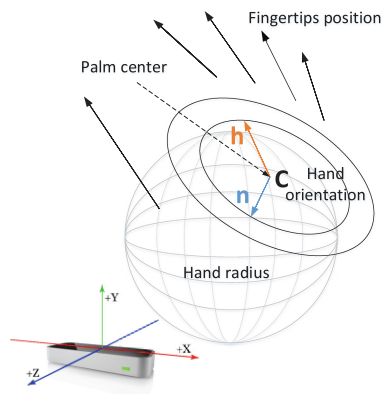


Fig. 2. The pose data from Leap Motion

- Number of detected fingers $N \in [0, 5]$.
- Position of the fingertips $F_i, i = 1 \dots N$.
- Palm center C .
- Hand orientation, h points from the palm center to the direction of the fingers, n is the normal to the plane that corresponds to the palm region pointing downward from the palm center.
- Hand radius r .

Note that the accuracy is not the same for all the reported data vectors. The 3D positions of the fingertips are quite accurate: according to a recent research [9] the error is about 200 μm . While the localization of the detected fingers is accurate, their detection is not too reliable. There are some situations in which the sensor is not able to recognize all the fingers such as fingers folded over the hand, hidden from the sensor viewpoint, fingers touching each other and the hand not perpendicular to the camera, etc. These issues are quite critical and must be taken into account in developing a

reliable gesture recognition approach since in different executions of the same gesture the number of captured fingers could vary. For this reason simple schemes based on the number of detected fingers have poor performance. As previously proposed approach [10] we deal with this issue by sorting the features on the basis of the fingertip angle with respect to the hand direction h . In this paper, we also consider the ratio of circumference and area.

In order to account for the fingers misalignment, we consider the projection of the hand region into the palm plane described by n and passing through C , as depicted in Fig. 3. In this work we analyze 5 different types of features computed from the Leap Motion data that will be described in the rest of this section:

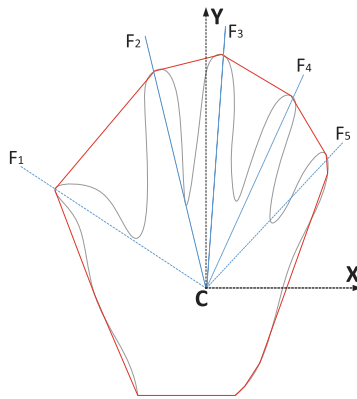


Fig. 3. The features of hand

- Fingertip angles: angles corresponding to the orientation of each fingertip projected onto the palm plane with respect to the hand orientation h .
- Fingertip distances: 3D distance of the fingertips from the hand center.
- Fingertip elevations: distances of the fingertip from the palm region plane.
- Fingertip 3D positions: x , y and z coordinates of the fingertips.
- Circumference and area ratio: Circumference is the sum of the connections between the five fingers and the palm of the hand. Area is the middle part of the five fingers and the center of the palm.

All the features values (except for the angles) are normalized in the interval $[0, 1]$ by dividing the values for the distance between the hand center and the middle fingertip length $S = \|F_{\text{middle}} - C\|$ in order to make the approach robust to people with hands of different size. The scale factor S can be computed during the calibration of the system.

After the completion of the feature extraction, we need a classifier to classify the features. Due to the small number of samples, the SVM (Support Vector Machines) classifier [11] was selected for feature classification. In SVM, the sample space is mapped to a high dimensional space through a nonlinear mapping method, which makes the original nonlinear problem transform into the linearly separable problem in the feature space. Then the optimal hyper plane is found in the high dimension space.

In this way, the features are classified. Because the radial basis kernel function can realize the nonlinear mapping and need fewer parameters, the radial basis kernel function model is selected in this paper.

4.2 Noise Suppression

Although the tracking accuracy of Leap Motion is less than submillimeter in theory, Leap Motion still can produce identification unstable phenomenon. Device resolution, thermal magnetic noise, hand shake [12], visual and numerical block singular value solvers are likely to introduce noise signal. In this paper, the literature [13] proposed an adaptive cut-off frequency of the low pass filtering method, the cutoff frequency of the low pass filter will be changed by detecting the real-time speed of palm. Equation (3) gives adaptive filter parameters and calculation expression.

$$\hat{X}_i = a_i X_i + (1 - a_i) \hat{X}_{i-1} \quad (3)$$

X_i is a high-dimensional vector which is composed of the coordinates and vector values returned from Leap Motion.

\hat{X}_i is a vector after adaptive filtering, a_i , as Eq. (4), can be calculated by the sensor data update interval T_i and time constant τ_i , as Eq. (5), is a smoothing factor between [0,1]:

$$a_i = \frac{1}{1 + \tau_i/T_i} \quad (4)$$

$$\tau_i = \frac{1}{2\pi f_{ci}} \quad (5)$$

The cut-off frequency f_{ci} is determined by the Eq. (6). \hat{V}_i represents the linear velocity and angular velocity of the palm movement is the derivative of \hat{X}_i . In this paper it takes the empirical value $f_{cmin} = 1$ Hz, $\beta = 0.5$.

$$f_{ci} = f_{cmin} + \beta |\hat{V}_i|. \quad (6)$$

5 Experiment and Result

For verification of the method, an action rules sub-experiment and an equipment assembly training sub-experiment were designed. Then subjective evaluation and objective evaluation were carried out [14].

This experiment has a total of 40 subjects, including 30 researchers who develop the space flight simulator and 10 graduate students who major in graphics and virtual reality. They understand the basic principles and application of virtual reality, computer graphics and the interaction with virtual reality system. But they didn't have the experience of the orbit assembly training.

5.1 Action Rules Experiment

In order to test our action rules and simulation FSM model, we designed a grab experiment. We analyzed 358 times grabbing operation from 40 subjects. We took the average number of grabbing operation every successful grab and the average grabbing time as the evaluation. The result is shown in Table 2.

Table 2. Evaluation of virtual hand grab

Evaluation	Average	Standard deviation
Numbers of grabbing operation every successful grab	1.50	0.984
Grabbing time (seconds)	4.41	3.364

From the evaluation above we can conclude that the virtual hand grabbing operation, which following our action determining rules and FSM model, has a high success rate and efficiency. This method can be used as astronaut operation training method in a virtual weightless environment.

5.2 Equipment Assembly Training Experiment

The experiment system is shown in Fig. 4. The subjects complete the gas generator for exercise assembly training according to the predefined process.



Fig. 4. The experiment system main view

To verify the effectiveness of the astronaut virtual assembly training, the assembly training method which guided by the learning manual were compared. The manual of assembly learning is a standard handbook. The specific assembly steps are introduced in detail by words and pictures. The manual of learning as the traditional way of training and virtual assembly training were compared to verify the efficiency and effect of the method of training.

Subjects were divided into control group and experimental group. Each group has 20 subjects. Control group does experiment by learning assembly manual training, and then have a real assemble. Learning training manual required time and the real assembly time were recorded. After the completion of the real assembly, the control group of experimenters transfer to virtual assembly training experience, experience after the completion of the fill in for virtual training methods of subjective evaluation questionnaire. While in the experimental group, in reverse order, assembly training through the virtual assembly training system, then the assembly objects, recorded both in manual training experience and fill in the questionnaire.

Through the calculation and analysis of the recorded data, we can find that, the average time required for the control group to training is 287.83 s, the experimental group is 183.81 s, the average time required for the control group and experimental group to manipulate are 87.13 s and 77.52 s, which indicates that the training efficiency and training effect of the experimental group are significantly better than that of the control group. Other results of subjective evaluation questionnaires are as shown in Table 3.

Table 3. Manipulation subjective evaluation questionnaires results

Evaluation	Means (0–5)	Standard deviation
Degree of alternative manual	3.875	0.68641
Interactivity	3.350	0.73554
Realistic	3.325	0.99711
Gesture recognition effect	4.025	0.76753
Operability	3.400	1.12774
Weightlessness effect	3.625	0.89693
Immersion	4.175	0.74722
Overall evaluation of virtual training	3.950	0.55238

It can be seen from subjective evaluation results and questionnaire that the proposed method is effective and can be used into the virtual training system.

6 Summary

Our experiments show that the advantage is obvious with using hand pose and gestures for HCI. As a new way to interact with VR system, it provides a more natural way to manipulate the virtual objects in the VR environment, reduces the user's learning time, improves operational efficiency and can very harmoniously embedded in training system. But there are also some disadvantages, such as the time to finish a menu operation is slower than using a mouse and as is the case usually occurs as likely to cause fatigue. Further research will be conducted to reduce the discomfort fatigue.

Acknowledgments. This work was supported by China Astronaut Research and Training Center within the following research programs: Advanced Space Medico-Engineering Research Project of China (2013SY54A1303, 9140c770204150c77318, 2013SY54B1301, FTKY201302).

References

1. Sen, F., Diaz, L., Horttana, T.: A novel gesture-based interface for a VR simulation: Re-discovering Vrouw Maria. In: 2012 18th International Conference on Virtual Systems and Multimedia (VSMM), pp. 323–330 (2012)
2. Sato, Y., Saito, M., Koike, H.: Real-time input of 3D pose and gestures of a user's hand and its applications for HCI. In: Virtual Reality, 2001. Proceedings, pp. 79–86. IEEE (2003)
3. Lee, C.-S., Oh, K.-M., Park, C.-J.: Virtual environment interaction based on gesture recognition and hand cursor. *Electron. Resour.* (2008)
4. Nasser, N.H., Alhaj, M.: Hand gesture interaction with a 3D virtual environment. *Res. Bull. Jordan ACM* **II(III)**, 186–193 (2011)
5. Ming, A., Yuqing, L., Bohe, Z., Fuchao, H.: Study on real-time interactive simulation of rotating top in weightlessness. *Manned Spaceflight* (2014)
6. Chen, Y., Lin, F.: Safety control of discrete event systems using finite state machines with parameters. In: Proceedings of the American Control Conference. Arlington: American Automatic Control Council, pp. 975–980 (2001)
7. Feng, Z., Yan, B., Xu, T., et al.: 3D direct human-computer interface paradigm based on free hand tracking. *Chin. J. Comput.* **37**(6), 1309–1323 (2014)
8. Hu, H., Chao, J.G., Liu, J.G.: Grab simulation based on FSM for astronaut virtual training. In: 3rd International Conference on Mechatronics, Robotics and Automation (ICMRA 2015), pp. 1060–1064 (2015)
9. Weichert, F., Bachmann, D., Rudak, B., Fisseler, D.: Analysis of the accuracy and robustness of the leap motion controller. *Sensors* **13**(5), 6380–6393 (2013)
10. Marin, G., Dominio, F., Zanuttigh, P.: Hand gesture recognition with jointly calibrated Leap Motion and depth sensor. *Multimedia Tools Appl.*, 1–25 (2015)
11. Chang, C.-C., Lin, C.-J.: LIBSVM: a library for support vector machines. *ACM Trans. Intell. Syst. Technol.* **2**, 27:1–27:27 (2011)
12. Vaillancourt, D.E., Newell, K.M.: Amplitude changes in the 8-12, 20-25, and 40 Hz oscillations in finger tremor. *Clin. Neurophysiol.* **111**(10), 1792–1801 (2000)
13. Casiez, G., Roussel, N., Vogel, D.: 1€ filter: a simple speed-based low-pass filter for noisy input in interactive systems. In: Proceedings of the 2012 ACM Annual Conference on Human Factors in Computing Systems, pp. 2527–2530 (2012)
14. Ledda, P., Chalmers, A., Troscianko, T.: Evaluation of tone mapping operators using a high dynamic range display. *ACM Trans. Graph.* **24**(3), 640–648 (2005)