The 3D Terrain Interactive Technique Based on Gesture Recognition

Yanyan Li[1, a], Xiaomeng Xu[2, b], Jiayu Sun[3, c], Haimeng Zhao[4, d*]

[1,2]Spatial Information Integration & Its Applications Beijing Key Laboratory, Peking University

[3,4]Beijing Institute of Collaborative Innovation

[a]liyan.@pku.edu.cn, [b]xuxiaomeng@pku.edu.cn, [c]sunjiayu@pku.edu.cn, [d]zhaohaimeng@163.com

Keywords: points cloud; Leap Motion; gesture recognition; Open Scene Graph

Abstract. The sensor Leap Motion can detect the information about locations and speed of hands and fingers with high accuracy in real time. It provides an effective method to the interaction between people and 3D terrain models, and based on the characteristics of the far and near field division of the working area of the sensor, different filter models are established to filter the natural gesture data, and to increase the stability of the anti-jitter ability and the software identification. This paper introduces a 3D controller based on OSG(Open Scene Graph), and designs a terrain model controller from gestures, and evaluates the controller by designing user experiments. The result shows that, comparing with traditional 2D interaction, the user experience and the result of this paper's algorithm is more natural, intuitive and efficient.

1 Introduction

In the process of 3D terrain reconstruction, the operations of the points cloud include zoom in/zoom out, translational motion along axis x, y and z, and rotation by axis x, y and z. Long time before, people mainly operated the reconstructed model by means of mouse and keyboard, but it's not an intuitive way to operate 3D models by 2D tools. Because of the short time of one operation by mouse, the frequency of wrong operations is very high. Some researches focus on how to transform 3D information to 2D inputs, or how to expand 2D inputs in order to control 3D transformation better. [1] designs a small widget named tBox to achieve translation, rotation and zoom of objects. [2] achieves the selection of center point among points cloud by using slice-n-swap. [3] proposes a virtual handle to rotate or translate 3D objects in virtual environment. In real life, people touch and interact with real objects with their hands[4], for example, take up, put down, move and so on. It means a lot to user-based human-machine interaction by operating 3D models with free hands and gestures.

Recognition of gestures is the key technique to interaction of gestures. Whether the devices can get gestures' information correctly is the basis of gesture interaction. In controls to 3D terrain points cloud, the information of the points cloud can get feedback and react in real time. [6] achieves hand gesture recognition with Kinect, but the advantage of Kinect is to capture data from medium or far distance. Leap Motion is a highly accurate sensor, with the accuracy of 0.01mm. It is specialized in gesture motion detection, and it can detect the slight change of movement of hands precisely.

This paper designs a well-understood interaction system based on Leap Motion. The system filters the input data, removes vibration we don't need, and builds an ergonomic 3D roaming device with Open Scene Graph(OSG) to response to real-time gestures and update the state of the terrain. The result shows that, comparing with traditional 2D interaction, the system in this paper makes it more natural, intuitive and efficient for users to deal with 3D models.

2 Related Work

2.1 Noise suppression. The sensor Leap Motion outputs the captured gesture data in the form of frame, including the space tracks of key points on hands, the speed of key points and pointer vectors. Although in theory, the precision of Leap Motion can achieve sub-millimeter level, noise produced by some unconscious actions and gestures lowers the stability and accuracy of data input, as hands are in the air and can move freely. From the view of hardware, Leap Motion uses two high-resolution cameras with the resolution of 640 * 240, in real life, it is common that the sensor can’t distinguish stably. So it is very important to filter out those meaningless noisy input from gestures.

Here are some gestures including rotation, drawing straight line and hovering. When fingers are operating 3D models, there is a certain degree of vibration. And the system filters the movement of fingers.

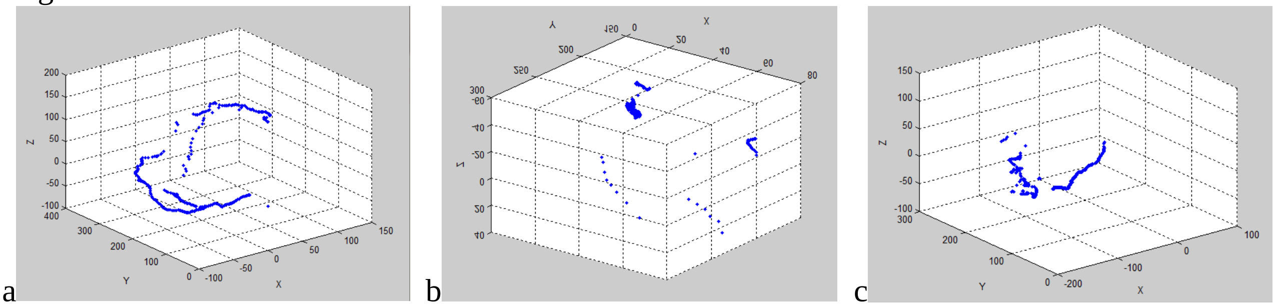


fig.1 the coordinate statistics of fingers when operating models a fingers draw a circle, b fingers hover in the air, c fingers draw straight lines in the air. From the statistics, we can see that there is noises in the data of gestures.

This paper uses a simple speed-based low-pass filter mentioned in [5], change the frequency of low-pass filter in real-time by detecting the speed of users’ palm area of their hands. The formula (1) presents the expression and calculation of parameters of the adaptive filter.

$$\widetilde{D_\iota} = \partial_i D_i + (1 - \partial_i)\widetilde{D_{\iota-1}} \tag{1}$$

In the formula, D_i refers to the high dimensional coordinates from Leap Motion, $\widetilde{D_\iota}$ refers to the vector after using the low-pass filter, and ∂_i is the smooth factor between 0 and 1, which can ben calculated by data update interval T_i and time constant τ_i:

$$\partial_i = \frac{1}{1+{}^{\tau_i}/_{T_i}} \tag{2}$$

$$\tau_i = \frac{1}{2\pi f_{ci}} \tag{3}$$

2.2 The 3D roaming device based on Open Scene Graph(OSG). When designing roaming devices, we need to set the event response based on the event of click, double click, mouse’s wheel, or type by keyboard. The response of the event is mainly through the setting of the translation step size and the rotation step size. Because of the limitation of mouse and scene, a mouse needs to split up an event into several movements, when dealing with large-scale rotation and translation events. Leap Motion uses The Interaction Box, which is defined as a rectilinear area within the Leap Motion fiend of view[7], which allows enough space for an action.

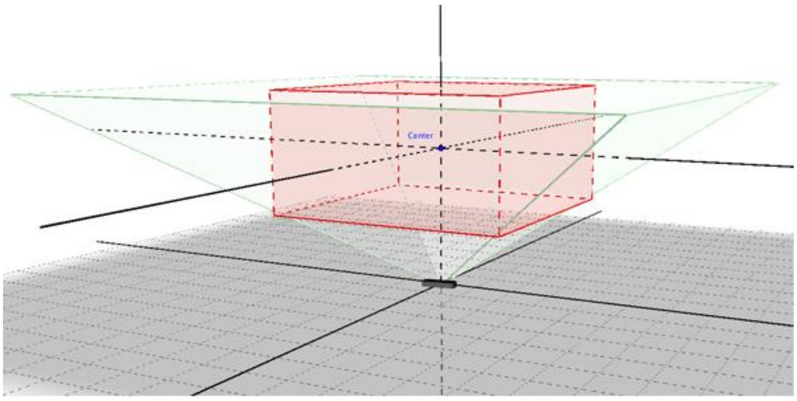


fig.2 The recognition area and center of leap motion sensor.

The roaming device need to store the coordinate (x, y, z) of center point of the points cloud, and the current step length. After capturing the gestures and filtering the data, Leap Motion submit the data to the roaming device, to judge the purpose the gesture. After judging the gestures, Leap Motion deals with the points cloud and store the newly updated points cloud, combined with current step length.

$$S_{i+1}(x,y,z,\partial,\beta,\gamma) = S_i(x,y,z,\partial,\beta,\gamma) + C_i(x,y,z,\partial,\beta,\gamma) \tag{4}$$

$S_i(x,y,z,\partial,\beta,\gamma)$ means the status of points cloud at time i, $C_i(x,y,z,\partial,\beta,\gamma)$ means the information of control at time i. (x, y, z) means the coordinate of point, and(∂,β,γ) means the angle to x, y, z axis. According to the coordinates of OSG, zoom in/out is transformed to the change of axis Z.

$$S_{i+1}(z) = S_i(z) + C_i(z) \tag{5}$$

$S_{i+1}(z)$, $C_i(z)$ means the value of z changes and other value don't change, while the points cloud and control information at time i.

The drag of points cloud is transformed to the translation of vectors, and x, y of the two vectors refer to the click position (x_i, y_i) and the release position (x_j, y_j)on the screen. When executing the gesture of translation, we get the location of points cloud, and we get the new location of points cloud after releasing gestures.

$$\Delta\mathrm{t}(x,y) = t_{i+1}(x,y) - t_i(x,y) \tag{6}$$
$$S_{i+1}(x,y) = S_i(x,y) + \Delta\mathrm{t}(x,y) \tag{7}$$

$\Delta\mathrm{t}(x,y)$ means the distance moved at surface x, y.

The roaming device has an initial angle (a, b, c). In rotation mode, the device calculates the fingertips of length of angle in Leap Motion axis, transform the angle of rotation into length of arc. If the fingertips are near to the coordinate origin, the rotation angle is short, and the arc is small too. Otherwise, it is large.

$$\Delta\mathrm{d}(x,y,z) = d_{i+1}(x,y,z) - d_i(x,y,z) \tag{8}$$
$$C_i(\partial,\beta,\gamma) = DegressToRadian(K * \Delta\mathrm{d}(x), K * \Delta\mathrm{d}(y), K * \Delta\mathrm{d}(z)) \tag{9}$$
$$S_{i+1}(\partial,\beta,\gamma) = S_i(\partial,\beta,\gamma) + C_i(\partial,\beta,\gamma) \tag{10}$$

$\Delta\mathrm{d}(x,y,z)$ means angle change while sliding, $DegressToRadian(\ )$ means changing angle value to arc value.

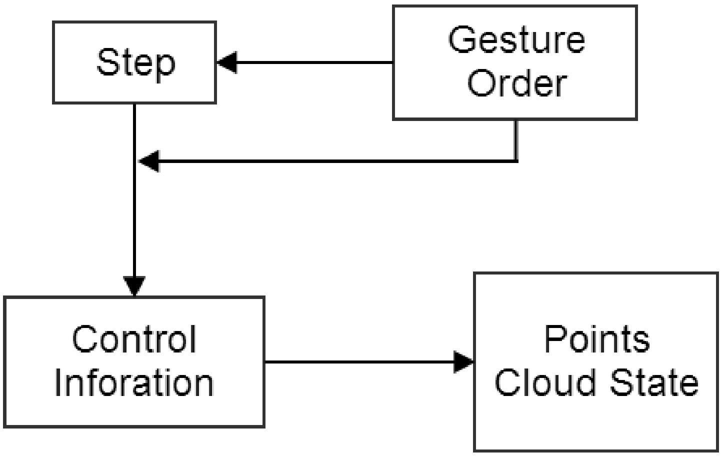
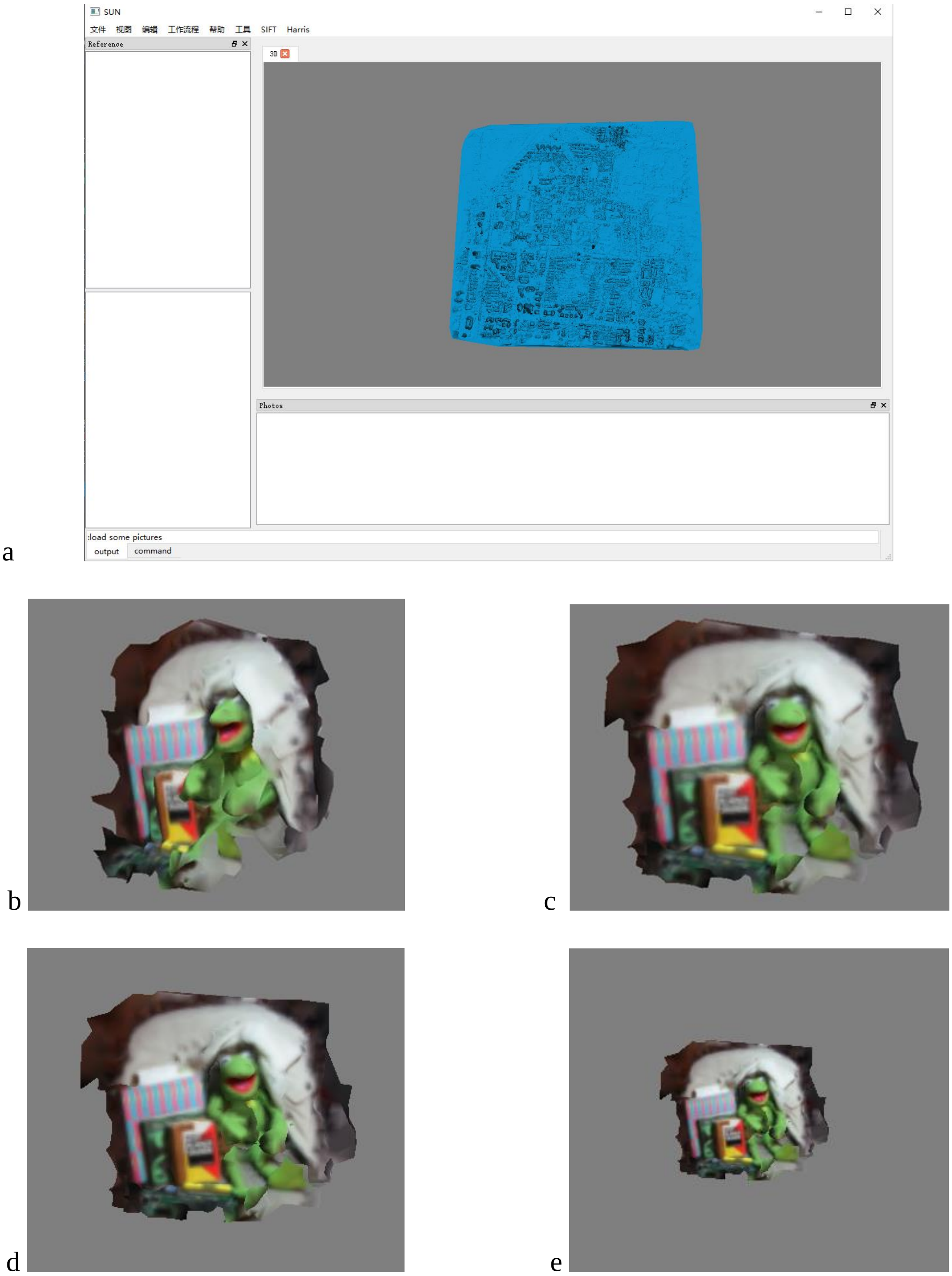


fig.3 The key flow chart of 3D roaming device. By recognizing the gestures, the device updates the control data(step length), and then execute the interact of rotation, translation and zooming in/out.

3 Experimental

The hardware system used in this paper includes a Dell Vostro 3900 computer, with Intel i7-4790 CPU 3.60GHz, 8G memory, a NVIDIA GeForce GTX 745 graphics card, and Windows 10 operating system. The development environment is Visual Studio 2015. A Leap Motion sensor is needed, and use SDK V2.3.1 for Windows provided by Leap Motion cooperation. And this paper uses Qt 5.6 to develop this software’ s interface, and uses Open Scene Graph(OSG) 3.4 as the engine of 3D display.



:load some pictures

fig.4 pic a is the main interface for development, in which the gray background is for 3D display. By rotating to the right, we get pic c from pic b. By zooming we get pic e from pic d.

Analysis: this experiment confirms that the system can rotate, translate and zoom in/out with gestures. When operating 3D terrains with hands, there is a certain noises. The system can lower the noises to a great extent by filtering the operating data. And finally the system completes the implementation of targets according to the wish of users.

4 Result and discussion

In this paper, we proposed a 3D hand gesture operating method based on Leap Motion. Based on the manipulation and exploration of large-screen data control and exploration, this paper proposes a low cost, but efficient and intuitive 3D spatial interaction technology of Leap Motion. The advantages of this technology is that, it makes interaction between people and points cloud more natural by operating the points cloud with free gestures.

When making it easier to interact, Leap Motion has the problem of low accuracy. When dealing with tiny change, Leap Motion is much more difficult than mouse, and Leap Motion has great advantage dealing with large scale interact.

References

[1] CohéA, DècleF, Hachet M. tBox: a 3D transformation widget designed for touch-screens [C]//ACM Conference on Human Factor in Computing System. Vancouver, Canada, 2011:3005-3008.

[2] Felipe Bacim, Mahdi Nabiyouni, Doug A. Bowman.Slice-n-Swipe: A Free-Hand Gesture User Interface for 3D Point Cloud Annotation [C]. IEEE Symposium on 3D User Interfaces, MN, USA, 2014:185 -186.

[3]Song P, Goh W B, Hutama W, et al.. A handle bar metaphor for virtual object manipulation with mid-air interaction[C]. Proceedings of the 2012 ACM Annual Conference on Human Factors in Computing Systems, Austin, TX, USA, 2012: 1297-1306.

[4] Murthy G R S, Jadon R S. A review of vision based hand gestures recognition [J]. International Journal of Information Technology and Knowledge Management, 2009, 2(2): 405-410.

[5] Casiez G, Roussel N, and Vogel D. 1€ filter: a simple speed-based low-pass filter for noisy input in interactive systems[C]. Proceedings of the 2012 ACM Annual Conference on Human Factors in Computing Systems, Austin, TX, USA, 2012: 2527-2530.

[6]Matthew Tang. Recognizing hand gestures with microsoft'skinect[R]. San Francisco:Department of Electrical Engineering of Stanford University, 2011, http://www.stanford.edu/class/ee368/Prpjec-ts /Tang-Hand-GestureRecognition.pdf.

[7] Leap Motion, Inc. https://www.leapmotion.com.