Development of a BCI Simulated Application System Based on Unity3D

Banghua Yang, Tao Zhang, Kaiwen Duan
college of mechatronic engineering and automation
Shanghai University
Shanghai, China
e-mail: yangbanghua@shu.edu.cn

***Abstract*—This paper develops a BCI simulated application system based on Unity3D, called going to the cinema. Firstly, a 3D car model and some scene models built in 3ds Max are imported into Unity3D .Then, in Unity3D, the virtual reality scene layout is designed with its own and imported models .The audio effects and collision detection are added to enhance realistic immersion. After that, the behaviors of the 3D car are defined. The communication and timer functions are configured. Finally, the system can receive the commands through the TCP/IP protocol to control the 3D car movement to go to the destination- cinema. The simulation results show that the system is feasible and has good performance. The designed system can be used as not only a simulated application system, but also a feedback training system providing effective feedback of vision and hearing, which can lay a foundation for BCI application and feedback.**

Keywords- brain computer interface; application system; Unity3D; virtual reality; TCP/IP

I. INTRODUCTION

A BCI (Brain Computer Interface) is a kind of interface to establish a direct communication channel between brain and external devices. Due to the high temporal resolution and portability of EEG (Electroencephalogram), the BCI system based on it is easy to put into practice [1]. The BCI system transforms the EEG signals to corresponding commands to control external devices [2]. Nowadays, BCI systems are widely used in the fields of rehabilitation, auxiliary control, entertainments and so on. However, on one hand, before the BCI systems are applied in practice, the hardware equipment needs to be configured to test their performance, which will increase the cost. On the other hand, most subjects can control their EEG independently only after numerous trainings [3]. But conventional feedback training systems are so boring that subjects lose motivation, which make the training effects are unsatisfactory [4][5]. Therefore, it is necessary to develop a BCI simulated application or feedback training system. It can be used to test the performance of the practical system without hardware devices. At the same time, it can provide effective visual and auditory feedback to improve the training effects, which will lay a solid foundation for the BCI application [6].

The VR (Virtual Reality) technology has the capability of creating an immersive and interactive environment [4]. Meanwhile, it can configure the ideal environment of the BCI practical system at a lower cost. So, the VR technology can be used to achieve the high quality simulated system. Unity3D, as an integrated game engine and editor, is one of the VR technologies. It is known that components are used to develop games in Unity3D, which include mesh component, physics component, audio component, rendering component and script component, etc. What's more, Unity3D provides rich resource packages, such as terrain creation tools, common scripts, collision detection, sky boxes and so on. High-level programing languages like c# and JavaScript are used for implementing the script functions [7]. In short, all of these are beneficial to create high-performance multimedia applications or games efficiently.

In this paper, a BCI simulated application system, called going to the cinema, is designed in Unity3D. Receiving commands from BCI signal acquisition and processing system through the TCP/IP protocol, the designed system can control the 3D car to go the cinema in real time. The visual and auditory VR scene in the system is more immersive and fun, which can inspire subjects' participating enthusiasm. In the meantime, the real-time feedback of vision and hearing makes subjects adjust their motor imagery, which improves the adaptability of human brains to computers. The timer function is designed to test subjects' control effects and the classification accuracy of the EEG signals. Above all, the environment configured is consistent with the practical one, which can effectively test the performance of BCI practical application system.

II. BCI SYSTEM OVERVIEW

The whole BCI system consists of two parts, which is shown in Fig. 1. One is the signal acquisition and processing system and the other is the Unity3D simulated application system. In the signal acquisition and processing system, the EEG signal, generated by subjects imaging left hand or right hand movement, is acquired by the electrode cap. Then the signal is processed and transformed to a real number in [-1, 1]. In order to control easily, the real number is further transformed to the command with the predefined format, which is used to control the car to perform turning left, turning right or moving forward. After that, the command is sent to the simulated application system to control the movement of the 3D car through the TCP/IP protocol.

The real number is transformed to the command to control the movement of the 3D car. So, the relationships

978-1-4673-7673-0/15 $31.00 © 2015 IEEE
DOI 10.1109/ICVRV.2015.30

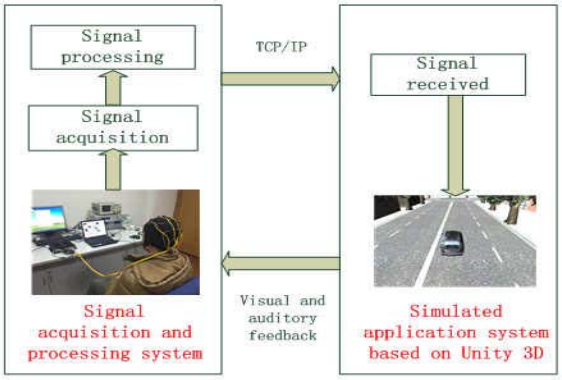

Figure 1. Whole BCI system

between the real number x and the movement of the car are described as

$$\begin{cases} v = 2*(0.5-|x|), x \in [-0.5, 0.5] \\ r = x*0.4, x \in [-1,-0.5) \cup (0.5,1] \end{cases}. \quad (1)$$

where the real number x in [-0.5.0.5] represents 3D car moving forward, and the relationships between the real number x and move speed v (m/s) can be described as a mathematics equation, v=2*(0.5-|x|). Similarly, the real number x in [-1,-0.5) and in (0.5, 1] represent 3D car turning left and turning right, respectively. The relationships between the real number x and rotation speed r (rad/s) is described as a mathematics equation, r=0.4*x. Among the predefined format, the "MD" is the rotating identifier while the "MS" is the moving identifier. For example, if the system receives the command of "MDR0.3", the car will turn right at a rotation speed of 0.3rad/s. If the system receives the command of "MDL0.3", the car will turn left at a rotation speed of 0.3rad/s. If the system receives the command of "MS+1", the car will move forward at a move speed of 1m/s.

III. SIMULATED APPLICATION SYSTEM DESIGN

The simulated application system is built in Unity3D with the models built, the visual and auditory VR scene constructed and the car behavior defined. Then the TCP/IP communication function is configured, through which the application system can receive real-time commands to control the 3D car. The timer function is also added to reflect subjects' control effects. The following steps to design the system are described in detail.

A. Building Models

In order to develop the system rapidly, the models are offered by 3ds Max and Unity3D own resource packages. The models include 2D road models and 3D models, such as a 3D car, buildings, trees, a cube and a sphere. Among them, the cube and sphere are used to replace the 3D car and the cinema in the mini-map respectively. Other models make up the VR scene.

As a tool for 3D modeling, animation making and rendering, 3ds Max is widely used in game development, character animation or other fields. More important, it cooperates with Unity3D smoothly. Therefore, the 3D car and the building models are built in 3ds Max with the method of polygon modeling, resulting in creating 3D mesh models [8]. The mesh models are mapped textures to increase their sense of reality and then are exported as the FBX file format. Finally, the model files are imported into Unity3D, which implements the resource delivery from 3ds Max to Unity3D.

In addition, because the mesh models of trees, roads, a square, a sphere are directly derived from Unity3D resource packages, they just need to be set up their textures and materials in mesh render component [9].

B. Constructing the VR Scene

At first, the scene layout is designed with the models built in Section 3.1. Then the sky, the collision detection and the audio effects are added to the scene. The mini-map is made to help subjects conduct the experiment. The steps of the VR scene construction are as follows.

1) Layout design: It is known that Unity3D uses the left-hand rectangular coordinate system. In other words, all the models are located in x-z plane. In Unity3D, any object appearing in the scene is bound to add the transform component, which contains three attributes: position, rotation and scaling. Hence, with the appropriate position values being set, the models built in Section 3.1 are placed in the reasonable positions, contributing to designing the scene layout conveniently.

2) Skybox: The sky is an important part of VR scene, which is used to make the scene more realistic. Thus, this system employs the skybox technique to render the sky. The skybox consists of six surfaces with sky textures.

3) Collision detection: Collision detection is indispensable in the VR scene. Otherwise, objects can pass into others, which violates physics rules of our real world and reduces the sense of reality [10]. Unity3D has abundant physics engines, which can simulate the real collision effects and draw a more vivid picture. The 3D car, as a special controlled object, is suitable to add the character controller, which can make the car detect collision and achieve the optimal collision effects. Other mesh objects just need to detect collision. So, they are added with the mesh colliders [11].

4) Audio effects: The audio effects can not only make the system more immersive and interesting, but also can serve as auditory feedback to remind subjects or adjust subjects' moods. So, four kinds of audio are added to the system. Among them, the audio of the car starting, the car driving, and the car colliding with trees or buildings can improve the scene of reality. The colliding audio can also reminder subjects to adjust their motor imagery. The audio of the fireworks displaying can encourage subjects to conduct a better motor imagery. The audio effects are controlled by the script, in which the audio is played by calling the function music.Play() and is stopped by calling the function music.Stop().

Figure 2. VR scene

5) Mini-map: Mini-map shows the positions of the 3D car and the cinema clearly. According to the mini-map, subjects adjust their motor imagery in time to control the 3D car to go to the cinema. Actually, the mini-map mainly depends on the vice camera, which always overlooks the VR scene. In the view of the vice camera, a red sphere and a blue cube respectively become the substitutions of the 3D car and the cinema. In order to follow the movements of the 3D car in real time, the sphere is affiliated to the car [12].

Fig. 2 shows the whole VR scene: the 3D car and the cinema are circled in red. In the lower left corner of the VR scene is the mini-map.

C. Defining Behavior of the 3D Car

The character controller component controls the character behavior more flexibly than the rigid-body component. Thus, the 3D car is added character controller component to control its movement. The character controller component of 3D car simulates a third-person perspective. That is to say that the main camera follows the 3D car forever to show the 3D car current movement clearly. Furthermore, a script is used to control the component. The script needs to inherit the MonoBehaviour class and then the function Start() and the function OnGUI() in it are overridden. The procedures of the definition are as follows:

(1) The move and rotation speeds of the car is defined, which are from the received commands.

(2)In the function Start(), the function GetComponent <CharacterController>() is called to obtain the component object ,which is used to make sure that the controlled object is the car.

(3)In the function OnGUI(), the function Rotate() is called to make 3D car turn left or right. When the car moves forward, firstly, the function transformDirection() is called to get the current direction that the car faces. Then the function Move() is called to realize moving forward of the car.

D. Configuring the TCP/IP Communication and the Timer Functions

1) The TCP/IP communication function: In order to avoid the thread blocking [13], the application system communicates with the signal acquisition and processing system through the asynchronous TCP/IP protocol [14]. The application system acts as the server, while the signal acquisition and processing system acts as the client. The specific communication procedures of the server are as follows: (1) Call the function Socket() to create a listening socket. (2) Call the function Bind() to bind the listening socket with an address, which consists of IP address and port number. (3) Call the function Listen() to monitor connection request of the client. (4) Call the function BeginAccept() to receive the connection request and respond to the request, Then the connection is established. (5) Call the function BeginReceive() to receive data from the client or call the function BeginSend() to send data to the client, completing data transmission with the client. (6)Call the function Close() to close the listening socket when the experiment is over.

2) The timer function: The timer function can not only measure subjects' control effects, but also reflect the classification accuracy of the EEG signals, which contributes to the further research and improvement of the signal acquisition and processing system. The realization of the timer function mainly depends on the function WaitForSeconds(), which makes the system main thread reach a wait state. Once the start button is pressed, the timer is triggered to count down. If the car fails to arrive at the cinema within the given time or reaches in advance, the timer is triggered to stop timing. The figure of the timer changes in real time until the experiment is over.

IV. EXPERIMENT RESULTS

During the experiment, the application system received commands through the TCP/IP protocol to control the car to turn left, turn right or move forward in real time. When the car reached the intersection on its way to the cinema, the subject imagined the movement of his right hand. So, the application system received the command of "MDR0.4", controlling the car to turn right at the rotation speed of 0.4rad/s, which is shown in Fig. 3. Moreover, according to feedback of vision and hearing, the subject adjusted motor imagery timely, thereby adjusting the car action. When the 3D car arrived at the cinema ahead of time, the timer stopped timing. Besides, the audio of the fireworks displaying and the caption-“Congratulations!” were used to encourage subjects. When the car failed to reach the cinema within the limited time, the timer stopped timing and the computer interface appeared a caption-“game over and fighting”.

Figure 3. Experiment result

In a word, the system is available. Meanwhile, the TCP/IP communication and the timer functions are normal. The effective multi-sensory feedback, urges subjects to conduct a better motor imagery, which improves the training effects.

V. CONCLUSION

In this paper, the system developed in Unity3D is a VR simulation application system providing effective visual and auditory feedback. Through the TCP/IP protocol, the system receives commands from signal acquisition and processing system to realize the movement of the 3D car. The VR scene built in the system is easy to arouse the subjects' effective EEG signals with strong immersion and interactivity. Besides, the development of the Unity3D simulated application system costs less money and less time. In a word, the designed system furnishes the BCI system with a good testing and training platform, which promotes the BCI application in daily life.

ACKNOWLEDGMENT

This project is supported by National Natural Science Foundation of China (60975079, 31100709), Innovation project of Shanghai Education Commission(11YZ19), Shanghai University, "11th Five-Year Plan" 211 Construction Project.

REFERENCES

[1] G. Pfurtscheller, C. Neuper, "Motor Imagery and Direct Brain-computer Communication," IEEE Journals and Magazines, Proceedings of the IEEE, vol. 89, Jul. 2001, pp. 1123-1134, doi:10.1109/5.939829.

[2] J. Tomas, S. Junst, and P. Willemsen, "Effectiveness of Commodity BCI Devices as Means to Control an Immersive Virtual Environment," SUI Proceeding of the Symposium on Spatial User Interaction, 2013, pp. 97, doi:10.1145/2491367.2491403.

[3] F. Benedetti, N. C. Volpi, L. Parisi, and G. Sartori, "Attention Training with an Easy–to–use Brain Computer Interface," Virtual, Augmented and Mixed Reality, Applications of Virtual and Augmented Reality, Eds.: Springer International Publishing, 2014, pp. 236-247, doi:10.1007/978-3-319-07464-1_22.

[4] R. Ron-Angevin, A. Díaz-Estrella, "Brain–computer Interface: Changes in Performance Using Virtual Reality Techniques," Neuroscience Letters, vol. 449, Jan. 2009, pp. 123-127, doi:10.1016/j.neulet.

[5] X. Bin, Y. Wenlu, D. Xiao, and C. Wang, "The Training Strategy in Brain-computer Interface," 2010 Sixth International Conference on Natural Computation, vol. 4, Aug. 2010, pp. 2190-2193, doi:10.1109/ICNC.2010.5583993.

[6] F. Lotte, J. Faller, C. Guger, and Y. Renard, G. Pfurtscheller, A. Lécuyer, and R. Leeb, "Combining BCI with Virtual Reality: towards New Applications and Improved BCI," Towards Practical Brain-Computer Interfaces, Eds.: Springer Berlin Heidelberg, Jul. 2012, pp. 197-220, doi:10.1007/978-3-642-29746-5_10.

[7] I. Aswin, M. Shinozaki, "The Investigation on Using Unity3D Game Engine in Urban Design Study," ITB Journal of Information and Communication Technology, vol. 3, 2009, pp. 1 – 18, doi:10.5614/itbj.ict.2009.3.1.1.

[8] J. H. Yu, X. M. Bai, and J. W. Lv, "Complex Object Modeling Method Based on 3ds MAX Platform," Applied Mechanics and Materials, Vol. 184-185, Jun. 2012, pp. 724-727, doi:10.4028/www.scientific.net/AMM.184-185.724.

[9] S. Wang, Z.L. Mao, and H.L. Gong, "A New Method of Virtual Reality Based on Unity3D," International Conference on Geoinformatics, Proceedings of the IEEE, Jun. 2010, pp. 1-5, doi:10.1109/GEOINFORMATICS.2010.5567608.

[10] B.H. Yang, Q. Wang, Z.J. Han, H. Wang, and L.F. He, "Development of a BCI Simulated Application System Based on DirectX," Practical Applications of Intelligent, Eds.: Springer Berlin Heidelberg, Jul. 2014, pp. 813-822, doi:10.1007/978-3-642-54927-4_77.

[11] R. Leeb, M. Lancelle, V. Kaiser, D. W. Fellner, and G. Pfurtscheller, "Thinking Penguin: Multimodal Brain-computer Interface Control of a VR Game," IEEE Transactions on Computational Intelligence and AI in Games, vol. 5, Jun. 2013, pp. 117-128, doi:10.1109/TCIAIG.2013.2242072.

[12] F. Sun, D. Hu, H. Liu, H. Zhang, J. Liang, Y. Liu, H. Wang, and L. Zhang, "An Iterative Method for Classifying Stroke Subjects' Motor Imagery EEG Data in the BCI-FES Rehabilitation Training System," Foundations and Practical Applications of Cognitive Systems and Information Processing, Eds.: Springer Berlin Heidelberg, 2014, pp. 363-373, doi:10.1007/978-3-642-37835-5_32.

[13] R. Emardson, P.O. hedekvist, M. Nilsson, and S.C. Ebenhag, " Time and Frequency Transfer in an Asynchornous TCP/IP over SDH-network Utilizing Passive Listening," Frequency Control Symposiun and Exposoyion, Proceedings of the IEEE, 2005, pp. 908-913, doi:10.1109/FREQ.2005.1574054.

[14] G. Carofiglio, L. Muscariello, "On the Impact of TCP and Per-flow Scheduling on Internet Performance," IEEE/ACM Transactions on Networking, vol. 20, Apr. 2012, pp. 620-633, doi:10.1109/TNET.2011.2164553.