Design and Realization of Gaze Gesture Control System for Flight Simulation

Yuanyuan Qiao¹, Jihui Wang¹, Jing Chen* and Jie Ren¹

¹School of Optics and Photonics, Beijing Institute of Technology, Beijing, 100081, China

*Corresponding author’s e-mail: chen74jing29@bit.edu.cn

Abstract. In airplane cockpit, flight crew, display interface, control device and onboard control device constitute a complex human computer interaction system. As a natural interaction way, eye movement can simplify the interaction process of pilots, avoiding mistakes caused by fatigue or distraction. What's more, since this interactive modality only requires eye movement as input, it can be adapted to some special situations and free the user's hands. In this paper, we design a gaze gesture interaction system for flight simulation and compare it with the existing prominent interaction methods.

1. Introduction
Eye-tracking technology provides a natural way to observe, analyse and utilize visual attention, has a wide range of applications in the field of human computer interaction[1]. Based on the physiological characteristics of the human eye and the basic pattern of eye movement, there are several different ways of eye movement interaction to avoid Midas' Touch issue, such as fixation[2], blinking[3], closure[4] and gaze gesture[5]. Nowadays, there are two mainstream gaze gesture recognition methods. One is gaze gesture recognition method based on single or multiple saccade behaviours [5], the other is based on smooth pursuit, uses the similarity of the eye movement trajectory and the object trajectory to interact[6]. The biggest difference between the two methods is whether there is a target for the eyes to follow, the second method requires the system to always display a variety of eye movement indicating animations, resulting in high interface complexity. In addition, because simple saccade gaze gestures are easily confused with unconscious eye movements, the first method often requires the design of complex gaze gesture[5] or combination with other modalities to distinguish it from unconscious eye movements, for example, Rajanna et al. determined the start of gaze gesture by pressing a hotkey or staring at the upper left corner of the screen [7], Yu et al. restricted the start and end areas of each gaze gesture[3]. Motivated by this, we proposed a real-time gaze gesture recognition algorithm based on the smooth pursuit and screen area for flight simulation.

2. Gaze gesture control system for flight simulation design

2.1. System Overview
This system is carried on a personal computer with Intel Core i7-10700K 3.80 GHz processor, 16GB RAM and Windows 10 64-bit operating system. We use C# language in Unity3D (version: 2018.3.13f1) to complete BP Neural Network construction, interactive scene rendering and UDP communication with x-plane. The gaze gesture control system of flight simulation can be described in Figure 1, which mainly includes three modules: the information collection module for gathering eye
movement data through the desktop eye tracker Tobii 4C, the information processing module for real-time recognition of gaze gesture behaviour and the flight simulation module for displaying flight interface and executing flight control commands. The information processing module includes two parts: user calibration and gaze gesture recognition. The former is used to obtain the corresponding data between eye image feature extraction parameters and gaze points under multiple different eye movement states by Tobii eye tracking software, the latter is used to identify gaze gesture control signals and transmit them to the flight simulation module by UDP transport protocol. In the flight simulation module, x-plane certified by the US Federal Aviation Administration was used in this paper to simulate aircraft scenes and Logitech flight yoke system was used as the yoke and throttle quadrant simulation controller.

![Figure 1. Gaze gesture control system of flight simulation.](image)

### 2.2. Gaze gesture task design

We classify the existing flight interaction tasks and obtain five sub-tasks of flight interaction, namely click, input, rotation, translation and zoom. Since the click and input tasks are often accompanied by the user’s visual attention focused on specific buttons, it is more suitable to use fixation based on dwell time to achieve, so this paper only discusses the use of simple gaze gesture to accomplish rotation, switching and zooming tasks, the corresponding gaze gesture design is shown in the Table 1.

| Flight simulation task type | Gaze gesture | Schematic diagram |
|----------------------------|--------------|------------------|
| Rotation                   | Clockwise and anticlockwise | [Diagram] |
| Switching                  | Swipe left and right | [Diagram] |
| Zooming                    | Diagonally upward and downward | [Diagram] |

*represents the starting area of the gaze gesture.*

### 2.3. Real-Time gaze gesture recognition algorithm

As shown in the Figure 2, each gaze gesture consists of two steps, one is that the eye movement coordinates fall within the gaze gesture start area, and the other is that the user's eyes move along a predetermined gaze trajectory. First, use 2D collision detection to determine whether the user's eye movement coordinates fall within the gaze gesture starting area. If so, play the eye movement indicator animation and record the user's eye movement data during the animation playback process. Next, after the animation is over, normalize the eye movement data and obtain the probability that the eye movement trajectory conforms to a predetermined gaze gesture according to the trained BP neural
network (which will be described in detail in section 2.4), if the probability corresponding to a certain gaze gesture meets the pre-set threshold (0.95 in this paper), the eye movement behaviour is confirmed as that gaze gesture.

![Diagram of gaze gesture real-time recognition algorithm]

Figure 2. The flow of gaze gesture real-time recognition algorithm.

2.4. Gaze gesture trajectory recognition algorithm based on Back Propagation neural network

Back Propagation (BP) Neural Network consists of an input layer, several hidden layers (which can be one or multiple layers) and an output layer, layer to layer is fully connected and there is no interconnection between neurons in the same layer. For the BP neural network described in Figure 3, the offline training of the model includes two processes, namely the forward propagation of the input and the back propagation of the error[8].
The data sets were collected from ten participants and each participant provided 20 repetitions for each of the six types of intentional gaze gestures. We normalize the eye movement coordinates recorded during the playback of the gaze gesture indicating animation into 80-dimensional input data by Min-Max scaling as the input layer and set the number of hidden layer neurons to 11, the output layer is a 6-dimensional array corresponding to the six types of gaze gestures, set the expected output vector according to the gaze gesture identification, for example, the expected output \([1, 0, 0, 0, 0, 0]\) corresponding to the gaze gesture with the identifier 1. Select sigmoid as the activation function, set the initial bias to -1, the learning rate to 0.1 and the error tolerance to 0.003, the loss curve during training is shown in the Figure 3. Due to the limited number of gaze gesture samples selected in this paper, we will discuss the overall accuracy of the algorithm by success rate in the next chapter.

### 3. User experiment

#### 3.1. Experiment design

1) Interactive task based on flight simulation: as described in section 2.2, this paper designed six gaze gestures for rotating, switching, and zooming. As shown in the Figure 5, scene c is the rotation scene, which controls the degree of aircraft flaps in x-plane with the clockwise and anticlockwise gaze gestures. Scene d and e are switching scenes, respectively representing the scene of enemy aircraft detection and basic display scene, where the swipe left gaze gesture will switch d to e, and the right will switch e to d. Scene e is also used as zooming scene, with the diagonally upward gaze gesture to zoom in and the diagonally downward gaze gesture to zoom out.

![Figure 4. Experimental scene.](image-url)
2) Gaze gesture indicator animation: we invited the subjects to follow the gaze gesture indicating animations at different speeds and adjust the speed through subjective evaluation. The final animation speeds are set as follows: the clockwise and anticlockwise rotate 120 degrees per second and the others translate 700 pixels per second.

3) Feedback: In the pre-experiment, the subjects showed that the use of gaze position indication can help them better complete the interaction task, so we use the gaze trace function of the Tobii eye tracking software to provide water drop-like interactive instruction.

4) Other interactive modalities for comparative evaluation: we designed touch screen interaction by Easytouch Unity plugin and gesture interaction by LeapMotion as shown in the Table 2.

Table 2. Touch screen and gesture interaction in flight simulation control system.

| Flight simulation task                  | Touch screen                           | Gesture                           |
|-----------------------------------------|----------------------------------------|-----------------------------------|
| Flaps rotation                          | Tap and swipe with one finger on the screen | Tap and swipe with one finger in the air |
| Situational information switching       | Swipe with one finger on the screen    | Swipe with two fingers in the air  |
| Situational information zooming         | Two fingers open and close on the screen | Palm open and hold in the air      |

3.2. Evaluation Indicators

1) Success Rate: the success probability of each interactive model for each interactive task, which can be calculated by dividing the number of successful interactions and the number of total interactions.

2) Task Completion Time: the completion time of interactive task, which can be obtained by subtracting the time when the system instructs the user to start the interaction and the system responds to the completion of the interactive task. The above two times can be automatically recorded by the system.

3) Cognitive load: used to measure the fatigue degree of the user in completing the task by this interactive modality. In this paper, participants filled up the NASA Task Load Index (which has six indicators of mental demand, physical demand, temporal demand, performance, effect and frustration) after completion of each modality. The total score of this index is 120. The higher the score is, the greater the cognitive load of users will be.

4) Satisfaction: participants were asked to rate their satisfaction with each of the interactive modalities on a scale of 0 to 5, with a higher score indicating greater satisfaction.
3.3. Experiment results

Ten participants were invited to perform six tasks in flight simulation, including adjusting the flaps degree (increasing or decreasing), switching scene, and viewing the situation information (zooming in and zooming out). Each participant completed each task 10 times using three interactive modalities, and the experimental results are shown in Figure 6. It can be seen that:

1) No matter from the objective indicators or subjective evaluation, touch interaction has achieved the best results.

2) The success rates of gaze gesture control commands are all higher than 92% (98% for anticlockwise, 96% for clockwise, 94% for swipe left, 93% for swipe right, 95% for diagonally upward and 92% for diagonally downward), so the gaze gesture recognition algorithm proposed by this paper can effectively reflect the user's intentions. In addition, the success rate of clockwise and anticlockwise gaze gestures (average 97%) is higher than that of straight-lines (average 93.5%), which may be due to the fact that curved gaze gestures are more tolerant of the user's blinking.

3) In terms of completion time, the touch screen has the lowest interaction time (around 2s), which may due to that the user is more familiar with the touch screen and the pre-set sliding distance is short. The pre-set eye movement trajectory is longer to distinguish it from the unconscious eye movement behaviour. Due to the similar length of each eye movement trajectory, the completion time of each interactive task is similar (around 6s). In addition, the completion time of the gesture interaction is related to the selected gesture (fluctuating from 1s to 10s). If it has nothing to do with the sliding distance, the time-consuming is shorter.

4) The success rate of gaze gesture (average 94.6%) is better than gesture recognition (average 92.3%), which may be because Leap Motion does not provide calibration for each user like the eye tracker, making it difficult to set parameters that match each participant.

5) Through the communication with users during the experiment, we found that gaze gesture interaction required mental attention to complete the gaze gesture trajectory, which may be the reason for the higher cognitive load (average score 44) and lower satisfaction (average score 2.5) of eye movement interaction than touch screen(scored 13 and 4.6). But the performance of gaze gesture interaction is similar to gesture interaction (scored 41 and 2.6). This shows that it is feasible to choose eye movements for interaction when the touch screen is not available.
Figure 6. Experiment results.

4. Conclusion
In this paper, we proposed and implemented a flight simulation system based on eye movement interaction, which uses the screen area to trigger gaze gesture interaction, and calculates the similarity between the eye movement trajectory and the pre-set trajectory through the BP neural network. The usability of this interactive modality has been verified through user experiments. However, due to the limitations of conscious eye movement, it brings a higher cognitive load to users. In the future, we will explore the integration of eye movement with other kinds of interactive modalities and make use of the advantages of eye movement that can better reflect users’ visual attention to complete pointing and selecting tasks.

References
[1] Heikkilä, H., & Räihä, K. J. (2012). Simple gaze gestures and the closure of the eyes as an interaction technique. Eye Tracking Research and Applications Symposium (ETRA), 1(212), 147–154.
[2] Istance, H., Hyrskykari, A., Immonen, L., Mansikkamaa, S., & Vickers, S. (2010). Designing gaze gestures for gaming: An investigation of performance. Eye Tracking Research and Applications Symposium (ETRA), May 2014, 323–330.
[3] Mahmud, S., Lin, X., & Kim, J. H. (2020). Interface for Human Machine Interaction for assistant devices: A Review. 2020 10th Annual Computing and Communication Workshop and Conference, CCWC 2020, March, 768–773.
[4] Møllenbach, E., Hansen, J. P., & Lillholm, M. (2013). Eye movements in gaze interaction. Journal of Eye Movement Research, 6(2).
[5] Fujii, K., Gras, G., Salerno, A., & Yang, G. Z. (2018). Gaze gesture based human robot interaction for laparoscopic surgery. Medical Image Analysis, 44, 196–214.
[6] Vidal, M., Bulling, A., & Gellersen, H. (2015). Pursuits: Spontaneous Eye-Based Interaction for Dynamic Interfaces. GetMobile: Mobile Computing and Communications, 18(4), 8–10.
[7] Wang, J., & Zhai, Y. (2020). Application of BP neural network based on genetic algorithm optimization in thermal conductivity of nanofluids. ICOPE 2019 - 7th International Conference on Power Engineering, Proceedings, 811–816.
[8] Yu, J., Liu, X., Zhang, S., Zheng, W., & Zhou, X. (2019). Design and Realization of Eye Control System for Small Ground Unmanned Platform. Proceedings - 2019 Chinese Automation Congress, CAC 2019, 4303–4308.