International Journal of Innovative Computing, Information and Control
Volume **18**, Number **2**, April **2022**

ICIC International ©2022 ISSN 1349-4198
pp. **417–432**

NAVIGATING THROUGH GOOGLE MAPS USING AN EYE-GAZE INTERFACE SYSTEM

Hanif Fermanda Putra[1] and Kohichi Ogata[2]

[1]Graduate School of Science and Technology
[2]Faculty of Advanced Science and Technology
Kumamoto University
Kurokami 2-39-1, Chuo-ku, Kumamoto 860-8555, Japan
haniffermanda@st.cs.kumamoto-u.ac.jp; ogata@cs.kumamoto-u.ac.jp

Received September 2021; revised January 2022

Abstract. *Over the course of decades, researchers and companies have developed different types of tools to enable human interaction with computers. One such tool, the eye-tracking device, has the potential to be utilized in all daily life applications. Even though there are numerous eye-tracking units in the market, they are rarely used in day-to-day life. In this study, we present the potential of utilizing an eye-tracking device and its interface as a navigation controller of Google Maps, a popular navigation system. Our goal is to translate eye-gaze movement into joystick-like movement. We compare the performance of the eye-gaze tracking system and the joystick in several short tasks, finding little difference. The study also shows that it does not require extensive training for a beginner user to control the eye-gaze system smoothly.*

Keywords: Eye-gaze, Eye movement, Input-system, Virtual tour, Navigation

1. **Introduction.** Human-computer interaction is a field that encompasses the research and development of technologies that support interaction between humans and computers. A simple example of human-computer interaction is the use of a keyboard to enter text information into a computer. Various devices such as the mouse and the microphone, have been developed to enable humans to interact with computers. Researchers have also been encouraged to develop tools that may help people with limited mobility interact with computers [1-12].

Although many devices such as the mouse, keyboard, and joystick have proven useful, all of them require control by hand, and people with disabilities may find it difficult to use them. Alternative control schemes are required to address this challenge. For example, Microsoft released the *Xbox Adaptive Controller* in September 2018 to help people with disabilities play games on desktop and console platforms [5]. This controller can be connected to different types of keys and analog sticks, enabling users to assemble and customize a comfortable controller. As another example, *Tobii* released a display-mounted eye-tracking device as an add-on to enhance the game experience [13]. The *Tobii* eye-tracking device provides additional control methods in several games. Although both devices potentially aid people with disabilities, healthy users can also utilize them.

In this study, we propose a system that allows us to interact with computers without using hands, i.e., eye-tracking devices. We also discuss the potential applicability of this system to everyday life. Eye-tracking devices have been widely researched and are commercially available in the market. We categorize eye-tracking devices into two groups:

- Screen-based mounted eye-tracking devices;

DOI: 10.24507/ijicic.18.02.417

- Wearable eye-tracking devices.

Although the eye-gaze detection methods and the types of sensors utilized in the devices can be divided into several more categories [14-16], we focused on the device location because it has a direct effect on the device's applicability. More specifically, screen-based devices require the user to sit in front of a screen, whereas wearable devices have a wider range of applications. Because the interaction provided by wearable eye-tracking devices is not limited to one screen, such devices do not restrict the user's position.

A screen-based mounted eye-tracking device is mounted and calibrated with a display screen. There are several types of screen-based eye-tracking devices [16]. Based on the eye-tracking sensor, there are two major types of desktop-display mounted eye-tracking devices. One type of sensor is an infrared camera that obtains the reflection of the user's pupil as an eye-gaze signal [7]. The other type of sensor is a visible light camera that detects the user's face, allowing the system to crop the user's eye image and obtain their pupil positions [12]. Alternatively, this system detects the user's head angle to predict their gaze position [2, 17]. Another implementation applies the eye-gaze tracking interface to mobile devices [8]. Screen-based mounted eye-tracking is useful; however, it has disadvantages. For example, a screen-based mounted eye-tracking device can only be utilized on a targeted display screen. In addition, the user's position and distance to the screen may affect the accuracy of the eye tracker [11, 18].

In wearable eye-tracking devices, an eye-tracking sensor is placed close to the user's eye. One benefit of such a design is that the head movement of the user does not affect the input. Furthermore, this design can be utilized in an open-world manner that is not restricted to a single screen. Several studies have been conducted on wearable eye-tracking devices [3, 4, 6], and ready-to-use products with wearable eye trackers [19, 20] are available commercially. However, these devices present several challenges. For example, wearable eye-tracking devices are currently expensive [21]. Furthermore, for day-to-day applications, wearable eye-gaze interface systems are intended for long-period use. Based on several reports, there is no evidence that infrared LEDs pose any hazard to the eye during a short period of exposure. However, questions remain regarding whether continuous exposure to an infrared LED may cause eye deterioration over time [2, 22, 23]. The lack of applications that support eye-gaze tracking devices is also a challenge. Currently, the eye-gaze tracking system is deemed only as an inessential device that supplements the user experience.

In summary, we have identified several challenges of eye-tracking devices. First, there is a lack of applications for the use of eye-tracking devices as a standalone input. Second, the market price is currently a challenge. However, low-cost development may become an option with the advancement of technology [16, 21]. Third, the screen-mounted tracking device may have a limited controllable area, although this challenge can be addressed with the development of wearable devices. Using a wearable device, a wide range of applications may be developed. However, the effects of long-term use of infrared sensors in a wearable device on the eye require additional expert evaluation. Therefore, different options can be considered to reduce the potential health risks.

Although we have addressed several problems in the development of eye-tracking devices, this study addresses their limited applicability. We propose an eye-gaze interface system to control the popular navigation system, Google Maps. Google Maps is a digital map service that allows users to view the map of any location worldwide. It also allows users to experience a virtual tour from a 360° panoramic view. This function can serve as an entertainment system that allows users to experience a virtual tour. Our contributions can be described as follows.

- We present the design of a wearable eye-tracking device. Our design weighs less than 100 grams without the camera attached. The design is also adjustable to increase user comfort. Our design is comparable in terms of weight to the Tobii Pro Glasses with a core weight of approximately 76.5 g [19] and Pupil Labs with a core weight of approximately 22 g [20]. Furthermore, the wearable camera can be manufactured with a 3D printer, which would extensively lower its cost compared to that of Pupil Labs [24].
- The general communication between the eye-gaze interface and the machine is a pinpoint gaze. The pinpoint gaze is a method to map a user's eye gaze to the screen or gaze object. However, this method is very dependent on a display screen. We propose the use of digital input for our eye-gaze tracking system, as described in Section 3. The digital input is a translation of eye-gaze movement into joystick-like controls. This alternative control scheme may eliminate the dependency on a screen. The proposed system can also be used in screen-less applications such as gesture control.

The rest of the paper is organized as follows. Section 2 describes several related works to this study. The details of our system and the proposed method to use the eye-gaze system as a digital input are described in Section 3. Section 4 details our experiments and discusses the results. At the end of the paper, conclusions from the study and possible future research directions are described.

2. Related Works.

In this section, we describe several studies conducted prior to our proposal. Studies on the interactions between humans and the computer system through devices such as camera, mouse, and keyboard mapping to a display screen are examples that are related to this study.

2.1. Eye-tracking controlled mouse cursor.

The eye-gaze interface system has been used over the years. Most commonly, it is used as a system that can control a mouse or keyboard device. Different methods have been proposed for such systems. Arai and Mardiyanto's work discusses the possibility of using an eye tracker as a pointing device, and tests the proposed system on an on-screen keyboard to assess typing speed [11]. Arai and Mardiyanto utilized a screen-based mounted eye-tracking device with a visible light camera as the sensor, which provides a safe environment for the user. Although the accuracy of their system was not mentioned, their research successfully implemented an eye-gaze interface system.

Arai and Mardiyanto's study points out that the system is rendered unusable once the user leaves the screen. Furthermore, from our point of view, the distance between a user's head and the eye-tracking device should be considered. Inconsistencies in the distance can affect the system's control accuracy. The study also mentions that users may be compelled to leave the screen to recover from head fatigue. Such challenges are common in screen-based mounted eye-tracking devices.

In a recent study by Zdebskyi et al., a user's head pose was recognized in order to control a mouse pointer [2]. Several methods were reviewed in this study to ensure an accurate test result. The main method described in the study employed a neural network with a set of trained data consisting of forty-eight face coordinates and two pupil coordinates aligned with the mouse cursor coordinate. However, the result was not as good as expected, with an average error of approximately 100 pixels. In our view, such a huge margin of error makes accurate mouse control become harder. Another problem is that computing power heavily affects the speed of recognition.

2.2. **Keyboard mapping.** Several keyboard buttons are usually provided to control an application. For example, a game controller utilizes arrow buttons to move the player around, and several additional key buttons allow the character to perform specific actions. To fully control such features, Istance et al. mapped the keyboard button onto the screen [7]. When a user looks at a region on the screen, a button press may be triggered depending on the button map. The designed system allows the game character to be controlled.

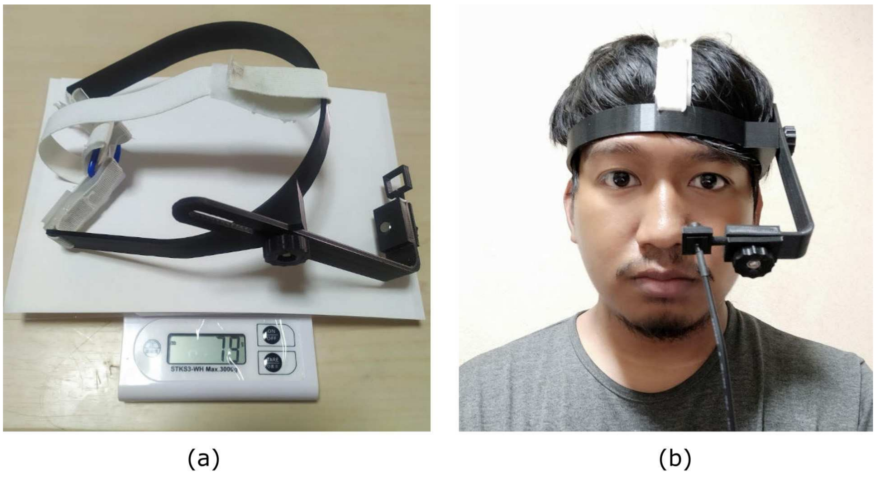
From our point of view, the design indicates promising results, although it may lack consistency. The challenge is located in the key mapped to the screen. The mapped region depends on the size and resolution of the display monitor. Although this may work well with a large-display monitor, the accuracy can decrease with screen size regardless of the type of eye-tracking device.

In a study by Utaminingrum et al., a map of a menu selection was developed in three different gaze directions [1]. A facial landmark was used to detect a user's head and the pupil coordinate. The study answers our earlier concern regarding how the distance between the eye and the eye-tracker may affect consistency. The horizontal length of the eye was measured before the user gaze was recognized. This idea made the system adaptable to any distance. However, the camera's ability to recognize eye movement would still be limited by distance. These reasons motivate us to design a wearable eye-tracker.

3. **System and Software Design.** In this section, we describe our developed system and the proposed method for implementing the eye-tracking device in an application. First, we describe the design of eye-tracking glasses. Second, we describe the application "Google Maps" and its controllable features. Finally, we describe the proposed method to control Google Maps using the developed eye-gaze interface system.

3.1. **Wearable eye-tracker.** In Section 2, we have addressed several problems presented by screen-based mounted eye-tracking devices. Some examples include the limited region in which a user's head can be detected, and the possibility of decreased accuracy as a result of inconsistent distances between the eye-tracking device and the user's head [11, 18]. These reasons became our motivation to develop a wearable eye-tracking device. We designed our wearable-eye tracker frame to be as lightweight as possible. Furthermore, we added support straps to disperse the weight of the system, making it feel more lightweight.

The wearable eye-tracker frame was produced using a 3D printer. Figure 1 shows the weight of the headband, which is approximately 79 g without the camera attached, and a



(a) (b)

Figure 1. (a) Wearable eye-gaze tracking device. Its weight was measured on a digital scale. (b) Snapshot of a user wearing our wearable eye-tracker.

snapshot of a user wearing our eye-tracker. The camera holder installed in the headband is adaptable to any camera shape.

We have also discussed the risk of using an infrared sensor for long-term continuous use. Therefore, we used a visible light camera for this system. In our previous study, we evaluated the performance of several cameras [25]. We concluded that our analog camera is most suitable for our designed system because of the minimal differences in the latency and performance of the camera under low-light conditions. Figure 2 displays the analog camera and the analog-to-USB adapter utilized in this research. Table 1 presents the specifications of the camera and the analog-to-USB adapter.

Figure 2. Analog camera and the analog-to-USB adapter used in this study

Table 1. Camera specification

Camera	YM-3020C
Sensor	1/4" SONY CCD
Resolution	512 × 492 NTSC (29.97 fps)
Analog-to-USB	LANMU USB 2.0 Video Converter
Driver	Not required

Prior to this study, our laboratory developed ring-shaped pattern matching for eye-gaze detection [3,4,6]. We developed a new interface system based on our prior work in the Python language and an open-source image processing library known as OpenCV [26,27]. The position of the iris is found by matching a ring-shaped template to a captured eye image. The colored eye image is converted into HSI color space to make it more robust toward light change. We then process the intensity value and convert it into a black and white color space using a thresholding method. We use the normalized cross-correlation to match the converted image to the ring-shaped template [28].

3.2. **Google Maps.** We chose Google Maps because it is currently one of the most popular map applications. Google Maps is a service that provides digital maps of the world. Furthermore, this service provides several features, such as satellite view as well as both 2D and 3D maps. These features allow the user to locate any place around the world. Google Maps also functions as a GPS that informs the user of their geographical location. Moreover, the digital maps also provide a service called Google Street View, which allows users to experience the scenery surrounding a specific road. Owing to this combination of features, Google Maps has become our choice.

The Google Maps service is available on both mobile and desktop devices. The mobile version can be controlled using finger flicks and button touches, whereas the desktop version can be controlled using a mouse with provided buttons or keyboard shortcuts. Important features for Google Maps users are as follows:

- Scroll function,
- Zoom function,
- Select a place, and
- Show Street View images.

Meanwhile, features that are mainly used in Google Street View include the following:

- Rotate function,
- Zoom function, and
- Move function.

In the desktop version of both services, each feature can be controlled using a mouse or keyboard shortcut. However, controlling these features via eye gaze can potentially be a challenge. As discussed in previous sections, eye-gaze tracking devices may be dependent on the display screen size. The application tends to maintain its control icon size rather than enlarging and maintaining the aspect ratio. Some icons were too small to be identified by the eye-gaze tracking system. To interact with the services using the eye-gaze tracking system, we introduce a digital input.

3.3. **Conversion of eye-gaze input into digital input.** In this section, we describe one of our contributions to address the challenge of a direct mapping system. As we have previously described, a direct mapping of the eye-tracking system is dependent on the display screen size. The mapping can be less accurate on a smaller display screen. As mapping the gaze becomes difficult on a smaller screen, one solution is to translate the eye-gaze position into a mapped control. Thus, we propose a digital input that can generate a joystick-like input. Figure 3 illustrates the proposed system. To move the mouse cursor, the user moves the center of the iris (represented by a white plus mark) into the desired control zone. The zone is divided by the green lines. We adopted the minimum bounding box algorithm to determine the position of the green lines [29]. Figure 4 illustrates our idea to implement the minimum bounding box algorithm. We removed outliers in the data using Equation (1). The value *Edge* represents the edge boundaries, which are denoted by the green lines. P represents the user's eye positions, where $\overline{P}$ and $\sigma(P)$ represent the average and standard deviation of the eye positions, respectively. Most eye-gaze tracking systems are affected by natural head movement, requiring users to hold their heads still or possibly use a chin-rest [30,31]. Because a chin-rest is not used in our current system, we add a constant C to the equation to minimize the error caused by natural head movement. Before the actual experiment, each user was asked to test the

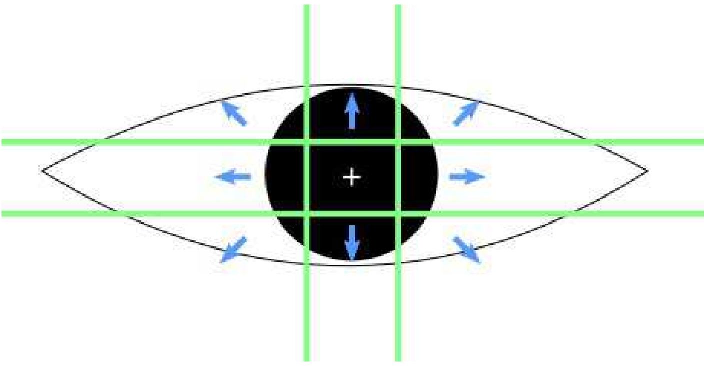


Figure 3. (color online) Illustration of a mapped area in an eye-image

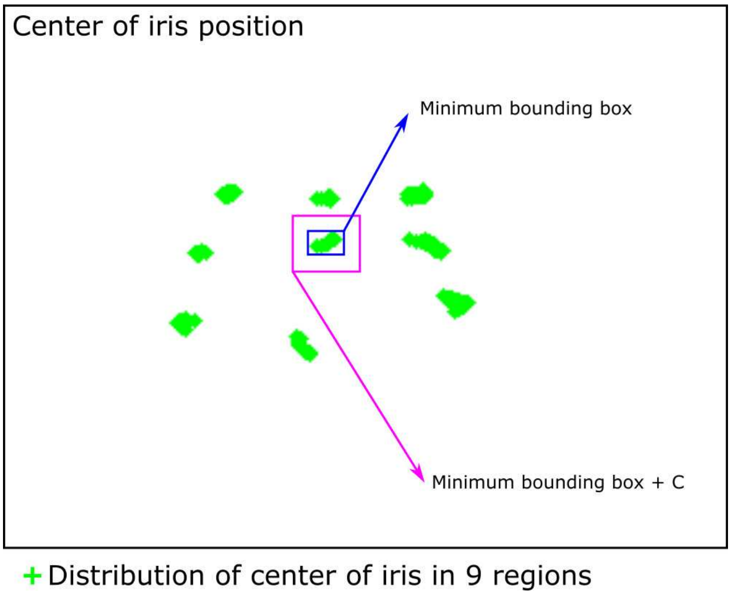


FIGURE 4. (color online) Illustration of the minimum bounding box

system to increase familiarity. During the trial, we determined the value of C that yields the most stable results to be $C = 20$ px. The value refers to the unintentional control of digital input caused by head movement. Hence, we adopted this value throughout the experiment.

$$Edge = \overline{P} \pm (\sigma(P) + C) \tag{1}$$

During calibration, the user was asked to look around the calibration point located at the center of the screen. We collected several gaze positions during the process and determined the minimum bounding box by obtaining the difference between the average and the standard deviation of the gaze positions.

We propose a combination of mouse control and keyboard shortcuts to control Google Maps. In Figure 3, blue arrows represent the direction of the cursor movement during mouse control. When the user gazes at a designated zone, the cursor moves without stopping until the user gazes at the center zone. The center zone itself does nothing. The user's blink is converted into a mouse click, and the user's long blink is converted into a double click. Mouse control is mainly utilized to start the application, select an area or a place on the map, and open the Street View image of the area. In addition, we provide a keyboard mode that allows the user to translate eye-gaze movement into keyboard shortcuts. Keyboard shortcuts can be utilized to control Google Maps and Google Street View.

3.4. **User interface of the controller.** In the last part of the system and software design, we describe the control system and connections. We have explained the mouse control and keyboard modes. To implement these control schemes, we provide a user interface panel with multiple buttons. We set the default control scheme to mouse cursor control. The user can use the cursor to select a button to switch to another control method. Each button was assigned a specific control.

Three buttons are provided. The first is the *Map Scroll* button, which is utilized to pan the 2D map in any direction. In Street View, the *Map Scroll* button can be used to rotate the view of the map. The second button is the *Zoom Mode* button, which is used to enlarge or shrink the image in both the 2D map and Street View. The third button is a *Free-Look* button that addresses the Midas touch problem that commonly appears in an eye-gaze tracking system [32]. To switch back to the mouse control, the user can blink for a longer period.

The control panel was connected to an eye-tracking user interface using socket communication. The eye coordinates and duration of blink data were sent from the eye tracker to the control panel user interface.

4. **Experiment and Discussion.** We describe the experimental setup and the results. The experimental setup describes our plan for validating our system. It comprises a predetermined task, computer specifications, and a prerequisite before the experiment begins. The experimental results are then discussed.

4.1. **Experiment setup.** Two experiments were performed. The first experiment was conducted to validate the proposed system with several short tasks. The basic control of the system was evaluated by comparing completion times. We compared the duration to complete each task between the eye-gaze tracking device and the joystick-keyboard combination. The second experiment was conducted to assess the user experience. Five healthy subjects were asked to test our system, and the required time for each trial was evaluated. Each subject filled out the agreement form required for the experiment. After the experiment, we gathered participants' opinions through a questionnaire.

In the first experiment, we evaluated our system by comparing the results of our eye-gaze tracking controller with those of a keyboard and a joystick. We evaluated the time required to complete each task. Each task was performed five times to verify the results. The tasks performed in the evaluation were as follows:

- **Task 1**: Move mouse cursor from Point A to Point B,
- **Task 2**: Rotate right in 360°,
- **Task 3**: Rotate left in 360°,
- **Task 4**: Move forward 5 steps, and
- **Task 5**: Move backward 5 steps.

For the first task, each user was required to move the mouse cursor from Point A to Point B on a 2D map. We decided that Points A and B were set at train stations, as illustrated in Figure 5. The rotation tasks (Tasks 2 and 3) and forward-backward motion tasks (Tasks 4 and 5) were set to start from the front gate of the Kumamoto University North Campus. The starting point of the rotation task was set with the right side of the gate as the center point. Figure 6 illustrates the starting point of the rotation task. The user was asked to rotate the screen 360° and return to the starting point. Using a similar scene in Google Street View, the forward motion task begins at the front gate facing forward, and involves a movement of five steps. The backward motion task begins from the front gate facing backward and involves a backward movement of five steps.

In the second experiment, each user was given the task of navigating Google Maps, from opening the application to closing Google Street View. Before the experiment began, all users watched a video simulation of the task. Figure 7 illustrates the flow of the task given to each user. The flow of the task is explained as follows.

- **Step 1**: The user moves the cursor and opens the Google Maps Application. We purposely enlarged the cursor and changed its color so that the user could easily locate it.
- **Step 2**: The user zooms in on Kumamoto University South Campus five times. The user can instantly zoom in on an area marked by a red box using a double click.
- **Step 3**: The user moves the cursor and clicks the area around the first intersection from the Kumamoto University South Campus. The clicked area is displayed as a

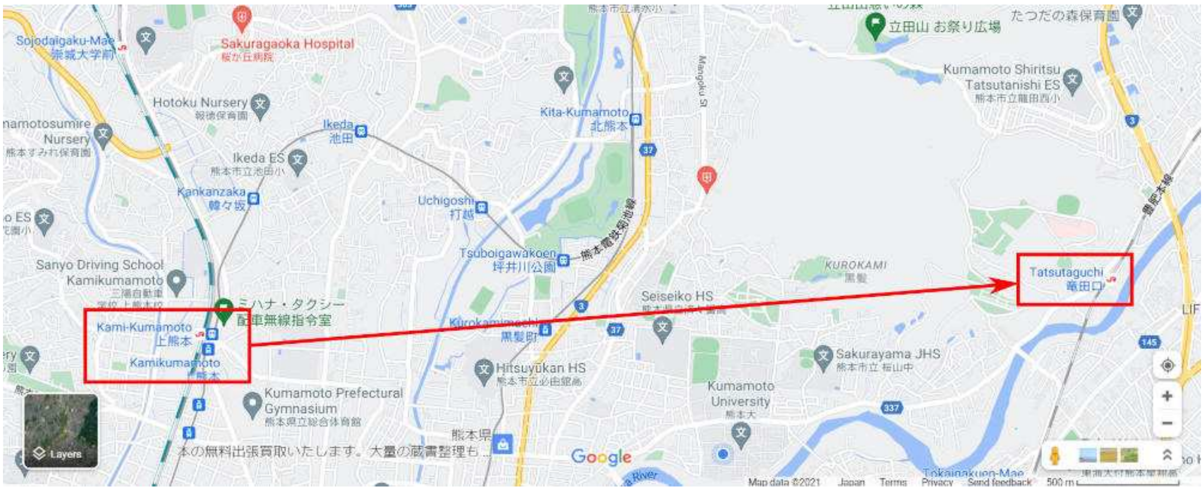


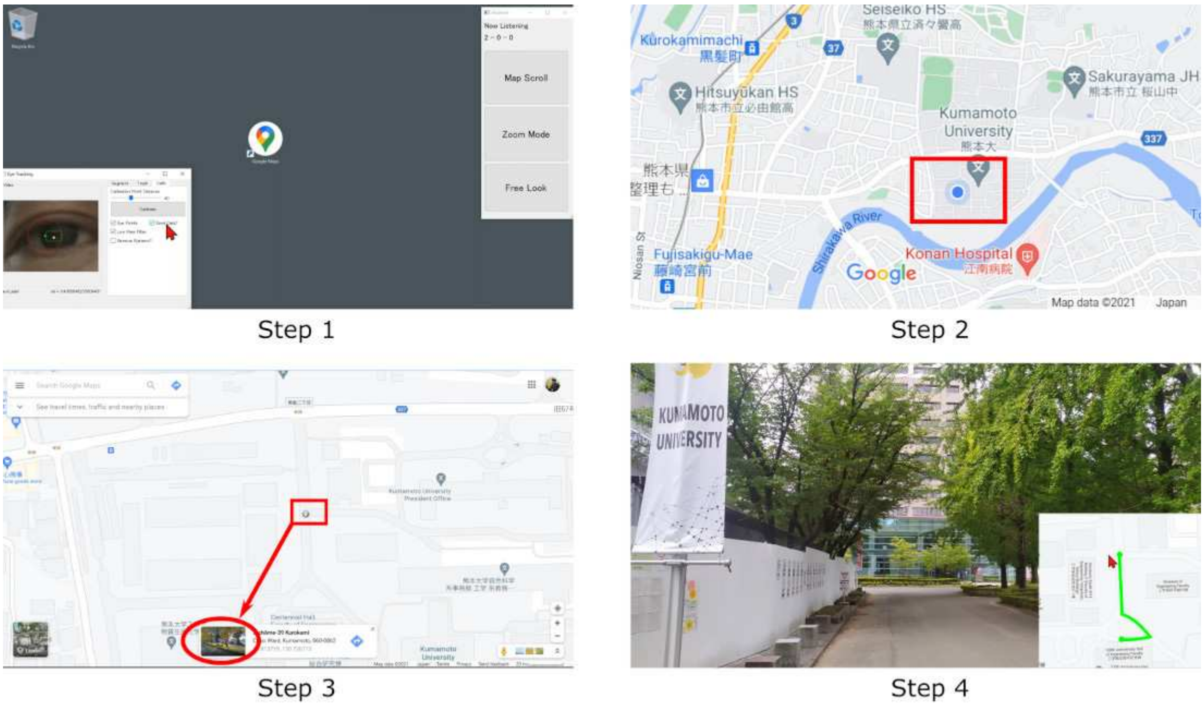
FIGURE 5. Moving cursor task. The box on the left indicates Point A at the Kami-Kumamoto Station. The box on the right indicates Point B at the Tatsudaguchi Station. The user was asked to move the mouse cursor from Point A to Point B. Map data source: Google Maps.

FIGURE 6. Starting point for the rotating task is marked in the red box. The Street View image was taken by the authors.

small icon marked by a red circle. The user then opens the Street View panoramic image of the area by clicking the image marked in the red circle.

- **Step 4**: The user navigates Google Street View by clicking the *Map Scroll* button. The control panel is automatically hidden in Street View. The user is asked to perform a short virtual tour from the intersection area to the front of the 100th anniversary of the engineering faculty building, as illustrated by the green line. Once the virtual tour is completed, the user returns to the 2D map view by closing their eyes for a longer period. The control panel reappears once the user returns to the 2D map.

Each user was asked to sit comfortably with the computer placed approximately 60 cm from the user. Compared to the standard pinpoint eye-gaze tracking system, our system does not require a minimum distance for optimal control. However, we set up a minimum distance due to health and safety concerns. Each user was asked to complete the task five times. We evaluated our developed system by measuring the required time for each user to complete the given task, as well as each user's answers to the questionnaire. All users completed the actual experiment on an HP ProBook 640 G4 (CPU: Intel Core i7 8550U 1.8 GHz, GPU: None, Memory: 16 GB, OS: Windows 10, Display: 15.6").


Step 1

Step 2

Step 3

Step 4

FIGURE 7. Flow of the task in the second experiment. A Street View image was taken by the authors. Map data source: Google Maps.

4.2. **Experiment result.** In the first experiment, we measured the time taken to complete each short task. Each task was conducted five times for validation purposes. Each subject that participated in the experiment had a different background. Subjects 1 and 4 were complete beginners with no knowledge or experience. Subject 2 was familiar with the eye-gaze tracking system, but had never used it. Subject 3 had personal experience with eye-tracking systems. Subject 5 was both knowledgeable and experienced in other eye-gaze tracking systems. Each user was first asked to control the mouse cursor using a joystick, and then using our eye-tracking system. The remaining tasks were performed to compare the keyboard and our eye-tracking system on Google Street View.

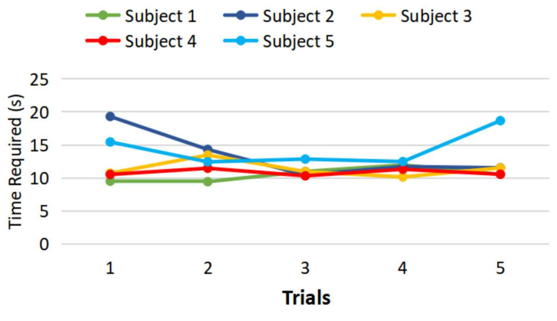
Table 2 compares the completion times between keyboard/joystick (Kb/J) control and our eye-tracking system. Based on the results presented in Table 2, the eye-gaze tracking system is generally slower than Kb/J. Although these results were expected, several ratios between Kb/J and our eye-tracking system were close. In Task 1, the average ratio of completion time between Kb/J and our eye-tracking system was only 1 : 1.15. The result suggests that any user can move the cursor using an eye-tracking system as easily as via Kb/J. In the rotation tasks (Tasks 2 and 3), our eye-tracking system exhibited completion

TABLE 2. Ratio of completion time between keyboard-joystick and our eye-tracking system

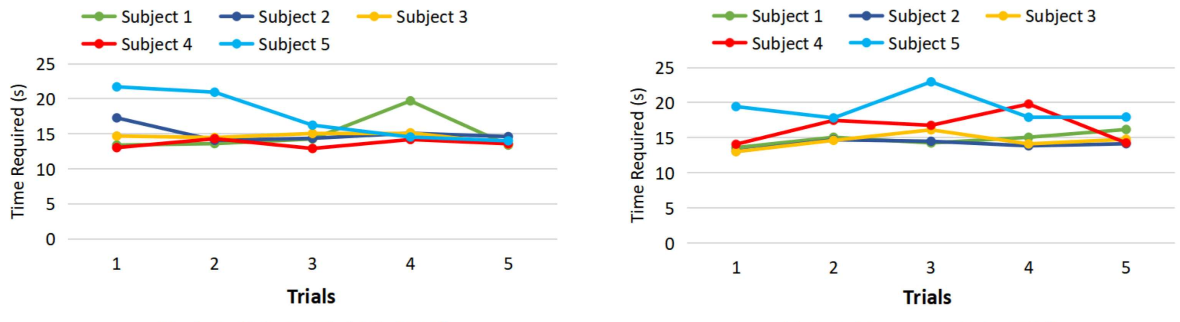
Tasks	Average Time Kb/J : Eye (ratio)					Average ratio
	Subject 1	Subject 2	Subject 3	Subject 4	Subject 5	
Task 1	1 : 1.04	1 : 1.14	1 : 1.13	1 : 1.09	1 : 1.36	1 : 1.15
Task 2	1 : 3.43	1 : 3.30	1 : 3.58	1 : 3.24	1 : 3.87	1 : 3.48
Task 3	1 : 3.37	1 : 3.19	1 : 3.54	1 : 3.86	1 : 4.48	1 : 3.69
Task 4	1 : 1.04	1 : 0.72	1 : 1.43	1 : 0.94	1 : 1.61	1 : 1.15
Task 5	1 : 1.07	1 : 0.87	1 : 1.65	1 : 1.12	1 : 1.32	1 : 1.21

times that are three times Kb/J. Although users were able to skillfully stop rotation with high precision at 360° using a Kb/J, it was not easy to stop at 360° with high precision using eye-gaze control at the same rotation speed. To overcome this problem, we reduced the rotation speed to increase the precision. In the moving tasks (Tasks 4 and 5), the average ratios between our eye-tracking system and Kb/J were close. Subject 2 and Subject 4 completed the task faster using eye-gaze rather than the keyboard. This result can be explained by the fact that there were few cases where the Google Street View page had a slow response that increased the loading time.

Figure 8 shows the corresponding completion time of each user using our eye-gaze tracking system in five trials. The difference in completion time between users is small. In the highlight of Tasks 4 and 5, although Subjects 2 and 4 completed the task faster using our eye-gaze, the graph shows that their completion time is not significantly different from other users. The overall results indicate an improvement from the initial trial followed by a stable completion time. In general, Subject 1 exhibits the best results compared to all other subjects despite being a complete beginner. However, the graph in Figure

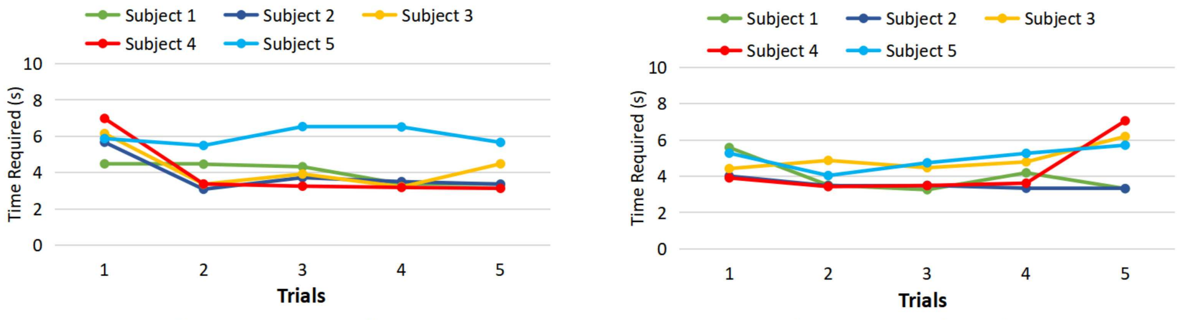


Task 1: Move from point A to B



Task 2: Rotate Right in 360°

Task 3: Rotate Left in 360°



Task 4: Move Forward 5 Steps

Task 5: Move Backward 5 Steps

Figure 8. (color online) Time required to complete each short task in the first experiment using our eye-gaze tracking system

8 indicates that there is minimal difference among the subjects. Based on the research results by Istance et al., it is difficult to expect an eye-gaze tracking system to perform as well as joysticks and keyboards [7]. Nevertheless, our research results show a tolerable difference, which indicates our eye-gaze system's potential.

In the second experiment, we evaluated each user's ability to control the system in longer-duration tasks. The experiment was divided into three timeslots. In each timeslot, each user was asked to complete the task five times. A short break was taken between timeslots. Fifteen completion times were recorded for each user. Figure 9 illustrates the average time of the five trials in three timeslots from each subject. In the first timeslot, three beginner users (Subjects 1, 2, and 4) have average times longer than those of experienced users (Subjects 3 and 5). Subject 4 averaged more than 300 seconds to complete the task. In the second timeslot, the performance of the three beginner users improved from the first trial. Subject 2 and Subject 4 completed the task approximately 50% faster than they did in the first timeslot on average. In the third timeslot, all users were able to complete the task with an average of under 200 s.

Figure 10 illustrates a trend of the time required to complete all fifteen trials for each user. Subjects 1, 2, and 4 had difficulties completing the given task in the first timeslot. Subjects 1 and 2 exhibited rapid improvement by the fifth trial in the first timeslot. By contrast, Subjects 3 and 5 had a better completion time in the first timeslot, which shows the users' experience. Subject 3, who was an experienced user, had the shortest

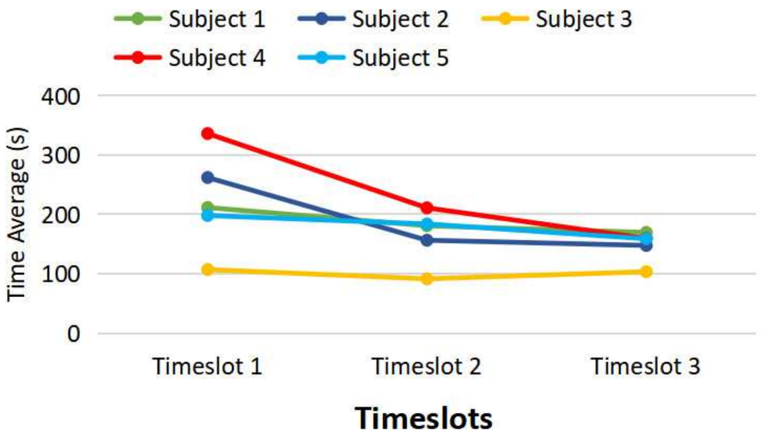


FIGURE 9. (color online) Average time of the second experiment in each timeslot

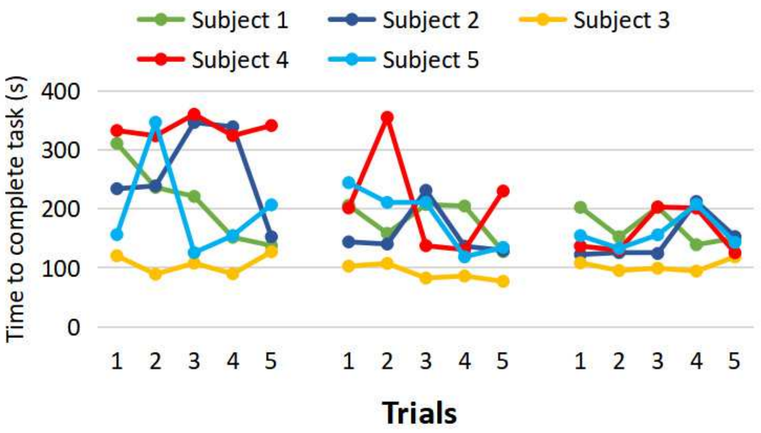


FIGURE 10. (color online) Time required to complete the second experiment

completion time in all timeslots. In the third timeslot, the small deviation indicates a more stable control and sense of familiarity with the system. The decrease in the deviation of the required time can also be seen as the number of trials increases, indicating the users' improving control skills. Considering the trend of the required time, there were occasional spikes in the completion time. The cause of these spikes is the pop-up feature on the 2D maps that indicates the summary of a building when the mouse cursor is hovering around it. Figure 11 illustrates the cause of occasional spikes in the second experiment for Subject 4, in the second trial of the second timeslot. This obstacle requires a quick decision by the user to move the cursor outside of the pop-up area and wait until the pop-up is closed in a relatively short time. From the results, we can conclude that beginner users do not require much time to get accustomed to our system.

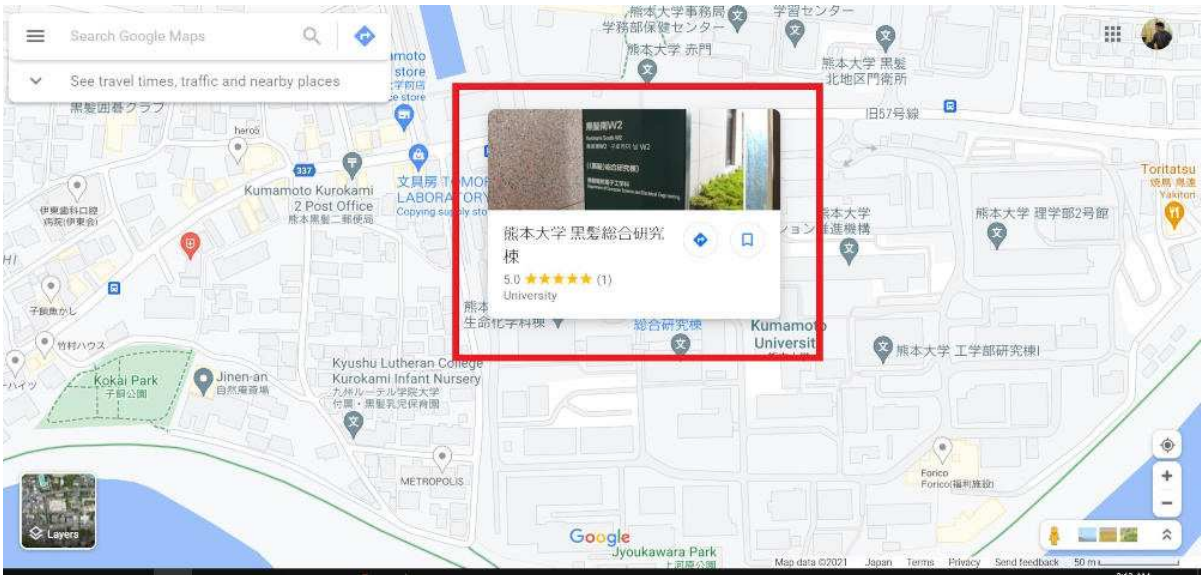


FIGURE 11. Target point is hidden behind the pop-up modal window of a building. The pop-up modal window indicates when the mouse cursor dwells around the building for a certain amount of time. Map data source: Google Maps.

Two recorded samples of Subject 1 from the first and second trials were taken, and the trajectories were analyzed. Figure 12 illustrates the trajectory of the mouse cursor controlled using our eye-tracking system. The image illustrates the trajectory of a mouse during Step 2, the zoom task. To be able to zoom into an area, the user is required to move the cursor to the top of the area and use a long blink to double click. However, in a few cases, the mouse cursor may dwell over a building with a name, causing the building information to pop up in a modal window. This window blocked the view, and the user was unable to zoom into the area. This challenge forces the user to move the cursor outside the window and wait for the window to close. This explanation is supported by a comparison of the trajectory between the blue and green lines. As illustrated in the image, the blue line has more detours before the zoom task.

After the trials were completed, we asked each subject to confirm any additional features in the system, such as the zoom function using the *Zoom Mode* button and free-look mode using the *Free-Look* button. The *Zoom Mode* button was prepared if any user wanted to enlarge or shrink the view of the map. The *Free-Look* button was prepared to solve the Midas touch problem, which is common in eye-camera research [32]. A questionnaire was prepared to collect the subjects' impressions of our developed system. Table 3 lists the

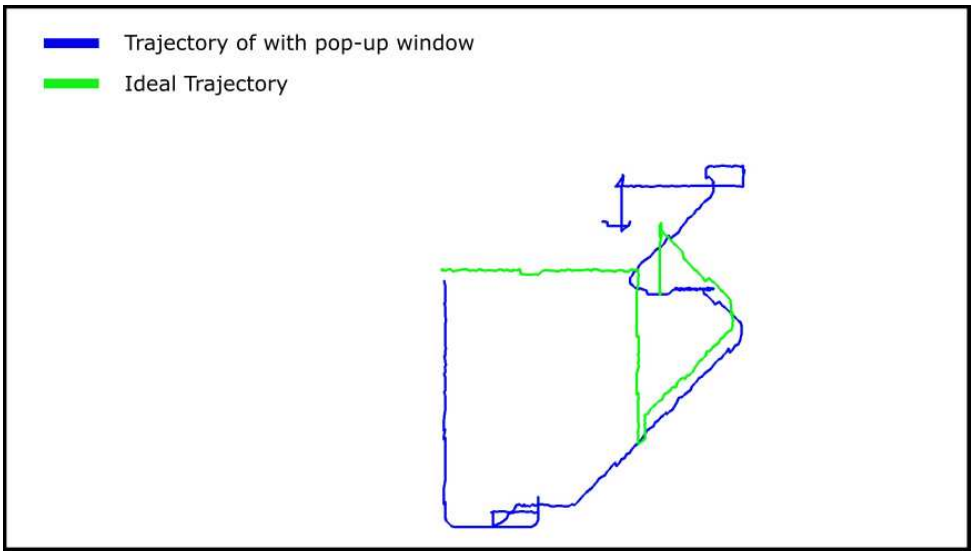


Figure 12. (color online) The trajectory of the mouse cursor. The blue line indicates the trajectory of the mouse cursor when a pop-up window appears. The green line indicates the trajectory of smooth control.

Table 3. Result of the questionnaire

Point to be reviewed	**Score**
Headband feels heavy?	3.8
Camera interferes with visibility?	3.6
Easy to move mouse pointer?	2.8
Easy to switch mode?	3.2
Easy to navigate through maps?	3.0

questions asked of each user, as well as the average score. The scoring for the questionnaire utilized a five-point system. A score of 1 represents the worst result, while a score of 5 represents the best result. Although the results in Table 3 display scores that are only slightly above average, our system was well-received by the test subjects. In particular, the design of the wearable devices was light, and did not strain the users.

We have received several comments regarding our developed system. Two of the subjects mentioned that they could not locate the cursor position while moving their eyes. This challenge may be addressed by using a larger display for future development. In addition, a center marker may be a good idea to help the user keep track of the center of the screen. Subjects 1 and 5 admitted that they blinked involuntarily, which caused unintended mouse clicks. This challenge can be addressed easily by adjusting the blink-time threshold for user preferences.

5. Conclusion.

Google Maps is a popular navigation system that provides maps and panoramic views of areas worldwide. This service can be both informative for navigation, and entertaining as a form of virtual tourism. In this study, we introduced the possibility of controlling Google Maps solely using an eye-gaze interface system. Considering the time required to complete tasks and the stability provided in the discussion, the eye-gaze tracking system has shown potential uses that are yet to be unlocked. A rapid reduction in completion time by the beginner users indicates that our system was easy to use. Based on the questionnaire results, our proposed headband design for wearable eye-tracking devices was relatively well-accepted. Nonetheless, we must improve our software system to improve user experience.

Acknowledgment. Part of this work was supported by Grants-in-Aid for Scientific Research JP17K06464 and JP21K04081 from the Japan Society for the Promotion of Science.

REFERENCES

[1] F. Utaminingrum, A. D. Purwanto, M. R. R. Masruri, K. Ogata and I K. Somawirata, Eye movement and blink detection for selecting menu on-screen display using probability analysis based on facial landmark, *International Journal of Innovative Computing, Information and Control*, vol.17, no.4, pp.1287-1303, 2021.

[2] P. Zdebskyi, V. Vysotska, R. Peleshchak, I. Peleshchak, A. Demchuk and M. Krylyshyn, An application development for recognizing of view in order to control the mouse pointer, *CEUR Workshop Proceedings*, vol.2386, pp.55-74, 2019.

[3] K. Ogata and S. Niino, Automatic threshold-setting method for iris detection for brown eyes in an eye-gaze interface system with a visible light camera, *Optics & Laser Technology*, vol.66, pp.112-121, 2015.

[4] T. Yonezawa, K. Ogata and K. Shiratani, An algorithm for detecting the center of the iris using a color image and improvement of its processing speed, *Electronics and Communications in Japan*, vol.92, no.4, pp.1-8, 2009.

[5] *Plugged in | Microsoft Story Labs*, https://news.microsoft.com/stories/xbox-adaptive-controller/, Accessed on December 24, 2020.

[6] K. Ogata, S. Inoue and T. Ito, A preliminary study of eye-gaze controlled vowel synthesis, *2015 IEEE International Conference on Systems, Man, and Cybernetics*, pp.827-831, 2015.

[7] H. Istance, A. Hyrskykari, S. Vickers and T. Chaves, For your eyes only: Controlling 3D online games by eye-gaze, *IFIP Conference on Human-Computer Interaction*, pp.314-327, 2009.

[8] K. Krafka, A. Khosla, P. Kellnhofer, H. Kannan, S. Bhandarkar, W. Matusik and A. Torralba, Eye tracking for everyone, *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition*, pp.2176-2184, 2016.

[9] H. F. Putra and K. Ogata, Development of eye-gaze interface system and its application to virtual reality controller, *2018 International Conference on Computer Engineering, Network and Intelligent Multimedia (CENIM)*, pp.208-213, 2018.

[10] H. F. Putra and K. Ogata, Prototype development of Google Maps navigation control using eye-gaze interface system, *Record of 2020 Joint Conference of Electrical, Electronics, and Information Engineers in Kyushu*, p.394, 2020.

[11] K. Arai and R. Mardiyanto, Camera as mouse and keyboard for handicap person with troubleshooting ability, recovery, and complete mouse events, *International Journal of Human Computer Interaction (IJHCI)*, vol.1, no.3, pp.46-56, 2010.

[12] M. Miyamoto, Y. Shimada, M. Yasuhiro and K. Shibasato, Development of eye gaze software for children with physical disabilities, *2016 International Conference on Advanced Informatics: Concepts, Theory and Application (ICAICTA)*, pp.1-6, 2016.

[13] *Tobii Gaming | Powerful Eye Tracking for PC Games*, https://gaming.tobii.com/, Accessed on December 24, 2020.

[14] P. Majaranta and A. Bulling, Eye tracking and eye-based human-computer interaction, *Advances in Physiological Computing*, pp.39-65, 2014.

[15] C. H. Morimoto and M. R. Mimica, Eye gaze tracking techniques for interactive applications, *Computer Vision and Image Understanding*, vol.98, no.1, pp.4-24, 2005.

[16] A. Kar and P. Corcoran, A review and analysis of eye-gaze estimation systems, algorithms and performance evaluation methods in consumer platforms, *IEEE Access*, vol.5, pp.16495-16519, 2017.

[17] Y. Matsumoto and A. Zelinsky, An algorithm for real-time stereo vision implementation of head pose and gaze direction measurement, *Proc. of the 4th IEEE International Conference on Automatic Face and Gesture Recognition*, pp.499-504, 2000.

[18] A. Gibaldi, M. Vanegas, P. J. Bex and G. Maiello, Evaluation of the Tobii EyeX eye tracking controller and Matlab toolkit for research, *Behavior Research Methods*, vol.49, no.3, pp.923-946, 2017.

[19] *The Latest in Wearable Eye Tracking | Choose Tobii Pro Glasses 3*, https://www.tobiipro.com/product-listing/tobii-pro-glasses-3/, Accessed on December 24, 2020.

[20] *Eye Tracking Technology – Gain Insight into Human Behavior – Pupil Labs*, https://pupil-labs.com/, Accessed on December 24, 2020.

[21] R. Mantiuk, M. Kowalik, A. Nowosielski and B. Bazyluk, Do-it-yourself eye tracker: Low-cost pupil-based eye tracker for computer graphics applications, *International Conference on Multimedia Modeling*, pp.115-125, 2012.

[22] A. Boucouvalas, IEC 825-1 eye safety classification of some consumer electronic products, *IEE Colloquium on Optical Free Space Communication Links*, pp.13/1-13/6, 1996.

[23] F. Mulvey, A. Villanueva, D. Sliney, R. Lange, S. Cotmore and M. Donegan, *Exploration of Safety Issues in Eyetracking*, COGAIN EU Network of Excellence, 2008.

[24] *Pupil Core – Open Source Eye Tracking Platform – Pupil Labs*, https://pupil-labs.com/products/core/, Accessed on December 24, 2020.

[25] H. F. Putra and K. Ogata, Evaluation of visible light camera towards better quality of eye tracking device, *Record of 2020 Joint Conference of Electrical, Electronics, and Information Engineers in Kyushu*, p.393, 2020.

[26] G. Bradski, The OpenCV library, *Dr. Dobb's Journal of Software Tools*, 2000.

[27] G. Van Rossum and F. L. Drake, Jr, *Python Tutorial*, Centrum voor Wiskunde en Informatica Amsterdam, The Netherlands, 1995.

[28] B. C. Song, A fast normalized cross correlation-based block matching algorithm using multilevel cauchy-schwartz inequality, *ETRI Journal*, vol.33, no.3, pp.401-406, 2011.

[29] G. T. Toussaint, Solving geometric problems with the rotating calipers, *Proc. of IEEE Melecon*, vol.83, p.A10, 1983.

[30] Z. Zhu, Q. Ji and K. P. Bennett, Nonlinear eye gaze mapping function estimation via support vector regression, *18th International Conference on Pattern Recognition (ICPR'06)*, vol.1, pp.1132-1135, 2006.

[31] Z. Wang, J. Zhao, C. Lu, F. Yang, H. Huang and Y. Guo, Learning to detect head movement in unconstrained remote gaze estimation in the wild, *Proc. of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pp.3443-3452, 2020.

[32] P. Majaranta and K.-J. Räihä, Twenty years of eye typing: Systems and design issues, *Proc. of the 2002 Symposium on Eye Tracking Research & Applications*, pp.15-22, 2002.

Author Biography

Hanif Fermanda Putra received his B.E. in Institut Teknologi Sepuluh Nopember, Surabaya in 2016, and the M.Eng. at Kumamoto University in 2019. He is currently a Ph.D. student at Kumamoto University. From 2016 to 2017, he was a research assistant at the Institut Teknologi Sepuluh Nopember. He had worked as a part-time research and development staff at Knowledge Communication Co., Ltd. from 2019 to 2021. His research interests include image processing, virtual reality applications, and embedded systems.

Kohichi Ogata is an associate professor with the Department of Computer Science and Electrical Engineering, Faculty of Advanced Science and Technology, Kumamoto University. He received the B.Eng., M.Eng. and Ph.D. degrees in Engineering from Kumamoto University, Kumamoto, Japan, in 1989, 1991, and 1994, respectively. His research interests include signal processing, speech processing, and image processing, such as the measurement of speech production process and development of its applications, development of eye-gaze interface systems, and augmented reality application systems.