IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 18 April 2023, accepted 24 May 2023, date of publication 2 June 2023, date of current version 16 August 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3282185

RESEARCH ARTICLE

Improving Gaze Tracking in Large Screens With Symmetric Gaze Angle Amplification and Optimization Technique

JOSEPH KIHOON KIM[1], JUNHO PARK[1], (Graduate Student Member, IEEE), YEON-KUG MOON[2], AND SUK-JU KANG[1], (Member, IEEE)

[1]Department of Electronic Engineering, Sogang University, Seoul 04017, Republic of Korea
[2]Korea Electronics Technology Institute, Seongnam-si, Gyeonggi-do 13509, Republic of Korea

Corresponding authors: Yeon-Kug Moon (ykmoon@keti.re.kr) and Suk-Ju Kang (sjkang@sogang.ac.kr)

This work was supported in part by the National Research Foundation of Korea (NRF) grant funded by the Korea Government (MSIT) under Grant 2021R1A2C1004208, in part by the Ministry of Culture, Sports and Tourism and Korea Creative Content Agency under Grant R2020040058, and in part by the National Research and Department Program through the National Research Foundation of Korea (NRF) funded by Ministry of Science and ICT under Grant 2021M3H2A1038042.

This work involved human subjects or animals in its research. The authors confirm that all human/animal subject research procedures and protocols are exempt from review board approval.

ABSTRACT Many gaze tracking applications focus on use in personal devices such as mobile phones and PCs. However, gaze tracking in large screens poses challenges because with an increase in screen size, gaze tracking in the edge region decreases owing to the restricted range of human eye movement. In addition, as large screens are often exposed to the public, anyone can use the gaze tracking module. This makes it difficult to apply personalized calibration as in personal devices. To acquire accurate gaze in the edge region, we propose a novel approach–symmetric angle amplifying function–for the gaze angle, which amplifies angles when a user is looking at the edge area of the large screen. Our function is designed particularly for the case where the screen is divided into grid-shaped regions. Furthermore, for the better user experience, we optimize neural networks using the network-optimization framework and also propose a center gravity function that pulls gaze coordinates presented on the screen to the predefined center of the region to compensate for the person-wise difference in movement of the human eyes. Experimental results revealed the superiority of the proposed methods over the baseline and different types of fitting functions. The gaze tracking module serves as a part of an aggregated system and is implemented for use in autonomous vehicles.

INDEX TERMS Gaze estimation, gaze tracking system, network optimization, user interface.

I. INTRODUCTION

Gaze tracking is a task of measuring where we look, what is referred to as point of gaze. It is one of the most relevant tasks in human–computer interaction applications. Most of the interaction with a device at a remote distance can be realized using gaze. Gaze tracking has been applied to games [1], visual attention analysis [2], and virtual reality simulators [3].

Previously, achieving accurate gaze tracking required a cumbersome calibration procedure using expensive hardware. However, as almost every device has built-in cameras, it has become easy for consumers to experience gaze tracking applications in their everyday life owing to the development of gaze estimation methods. Gaze estimation is closely related to gaze tracking, and it is the underlying task of gaze tracking, which detects and obtains gaze direction vectors from the input image. Gaze estimation methods can be categorized into two types: model- and appearance-based methods. A model-based method creates a geometric eye model

The associate editor coordinating the review of this manuscript and approving it for publication was Giuseppe Desolda.

and determines gaze from the features of the constructed geometric model. However, model-based methods have the limitation that the gaze estimation accuracy is low owing to its dependency on accurate eye feature detection. To detect accurate eye features, high-resolution images obtained from various sensors such as near infrared (NIR) cameras are required. In contrast to the model-based method, an appearance-based method depends only on the appearance, taking images of the eyes as input and learning the mapping between the appearance and the image. Therefore, it can handle low-resolution images, which makes it robust to the distance between the camera and the user. With the recent advent of deep learning architectures for computer vision tasks, gaze estimation methods have adopted a deep learning-based approach treated as other computer vision tasks. In this context, appearance-based methods become prevalent in the domain of gaze estimation. Deep learning-based methods benefit from the large amount of real and synthetic training datasets, facilitating unconstrained and person-independent gaze estimation. This means that we can estimate gaze in everyday environments without any assumptions regarding personal facial features and properties of the environment such as illumination conditions and camera angles. Several studies and commercial products that utilize gaze tracking have focused on personal devices such as mobile phones or PCs [4], [5], [6], which typically have small screens, designed for individual use. Thus, most gaze tracking applications in personalized devices require calibration to achieve accurate gaze tracking performances. However, in this work, we implement a gaze tracking system for large screens in a public environment. Furthermore, the gaze tracking system implemented in this work is not independent from two perspectives. First, to achieve an accurate point of regard, other measures such as face landmark and the global coordinates of the user are required. Second, the gaze tracking system is a part of an integrated system with various functionalities such as pose estimation, voice assistance, and hand gesture recognition.

The characteristics of the proposed gaze estimation system give rise to two main challenges in its implementation. First, in contrast to devices that have small screens, where the entire screen can be covered with relatively small eye movement, the eyes must be moved as far as possible to stare at the edge of large screens. The appearance of the eyes reflected on the camera is similar when staring at the point within the farmost edge region of the large screen and the point more inward but close to the point within the farmost edge region. This makes it difficult for the appearance-based model to distinguish the difference between the two points in each region. To resolve this issue, the small difference of the eye movement when a user is gazing at the edge of the screen needs to be amplified. Our observation indicates that the absolute value of the angle, which is the output of the gaze estimation network, enlarges as the gaze moves from the center to the edge of the screen. Therefore, we propose a novel function called symmetric angle amplifying function (SAAF), which amplifies the angle as the value of the angle enlarges, enabling accurate gaze tracking in the edge region.

The second problem we address is the necessity of a short inference time and the lightweight deep learning network. As discussed earlier, in many cases, the gaze tracking module is not a standalone component. In other words, it is often used with other modules and functions. For example, in our case, the gaze tracking module is used with multimodality modules, which we explain in the implementation section. In addition, gaze tracking is often implemented and embedded in edge devices rather than high-performance computing devices. Such considerations again impose constraints of light weight and real-time performance on the designed system. The innate aspect of gaze tracking also coincides with the constraints because the delayed inference would be noninformative for the gaze tracking task. Hence, we address this issue with a simple ResNet-based [7] gaze estimation model and utilize the network with a network-optimization framework. In summary, the contributions of our paper can be summarized as follows:

1) We propose a novel function, SAAF, for accurate gaze tracking in the edge region of a large screen. Using our SAAF, the limitation of the appearance-based gaze estimation method, which relies solely on inputs from the monocular RGB cameras, can be overcome.

2) We present a framework suitable for a public environment. In this work, we aim to utilize the proposed framework in the public environment aggregated with other application modules such as pose estimation, voice assistance, and hand gesture recognition. In addition, owing to the potential use under the multi-person condition, we design a person-independent system that does not require personalized calibration.

3) We analyze and verify the proposed method and framework with extensive experiments. Additionally, we compare the proposed method with the baseline model and other functions such as the polynomial function, piecewise function, and bezier function. Experiments revealed the effectiveness of the proposed method.

4) We show the implementation of the framework for use in the real world. Finally, we implemented our module on two different vehicles which show the practical use of the aggregated system with various user interface contents.

II. RELATED WORK

A. SYSTEM TYPE AND GAZE TRACKING ACCURACY

Generally, gaze tracking systems are categorized into head-mounted system and remote system. Head-mounted tracking system depends on the head movement on the user. The pupil and the glints on the corneal surface can be attained from the high-resolution near-eye camera of the system, thus giving the accurate gaze. For example, See Glasses uses one camera and 8 infrared sources for gaze tracking, achieving accuracy of less than 0.5° based on corneal reflection. Pupil

Invisible is the first deep learning based gaze tracking eye glasses. Besides scene camera, it is equipped with two IR near-eye camera for each side. In addition, it also utilizes IR LED to illuminate the respective eye region. On the other hand, remote systems generally can be operated at a certain distance. There are several studies on remote systems. Li et al. [8] proposed gaze estimation algorithm for long-distance camera based on deep learning using convolutional neural networks (CNNs). For the case of commercial products, Tobii Pro Fusion operates distance within 50-80cm. In the optimal condition, using pupil corneal reflection, the accuracy of the device is 0.3°. In addition, the number of cameras installed in the Smart Eye Pro is flexibly adjusted for different situations to determine the operating distance and tracking range reaching accuracy of 0.5°.

In summary, the proposed method was conducted as a remote system since several measuring equipment is required to be implemented as head-mounted system, which is expensive and not suitable for a public environment.

B. DEEP NETWORK ALGORITHMS

Deep learning-based gaze tracking methods use multi-layer neural networks to learn a model that maps between appearance and gaze. In general, the input image is expected to be the image of the entire face or eye.

Most recent solutions have adopted CNN-based architectures [9], [10], [11], [12], [13], [14], which aim to learn end-to-end spatial representations. Some studies [9], [10] have proposed datasets and corresponding architectures. Generally, most gaze estimation networks use modified versions of popular CNN architectures in computer vision downstream tasks (e.g. AlexNet [15], VGG [16], ResNet [7], and Stacked Hourglasses [17]). Usually, the difference between the networks comes from the input, intermediate features, or output. For the case of the input, it is divided into a single RGB image stream (e.g., face and left or right eye patch) [9], [13], or multiple RGB image streams (e.g., face and eye patch) [10], [18], and prior knowledge [11] based on eye anatomy or geometric constraints. From the perspective of intermediate features, GazeNet [9] concatenates the head pose to the output of the CNN encoder, and Pictorial Gaze [11] first regresses the gaze map and uses it as an intermediate feature to obtain the final gaze direction output. Finally, the output is different because of the dataset. Most networks output a gaze direction vector that is a 2D vector representing the yaw and pitch of the gaze. In [10], horizontal and vertical distances from the camera were directly regressed in centimeters.

Usually, the gaze moves continuously and dynamically when a person looks around the surrounding environment. In this context, a continuous sequence is generated, and a specific time image frame has a high correlation with a previous time image frame. On the basis of this logic, several studies [19], [20], [21], [22], [23], [24] have utilized time information and ocular kinematics to improve gaze estimation performance over single image-based methods. Given a series of frames, the goal of the task is to estimate the gaze direction to match the ground-truth direction as much as possible. To model this task, a popular recurrent neural network (RNN) structure has been explored.

In our system, we adopted a CNN-based model instead of an RNN-based model for the following reasons. First, CNN-based models are easy to implement because these have already been implemented in many frameworks, including optimization frameworks such as TensorRT. Second, CNN-based models generally have faster inference speed than RNN-based models. Third, CNN layers are more suitable for acquiring spatial features of images.

C. GAZE ANALYSIS DATASETS

With the rapid progress of gaze estimation, several datasets on gaze estimation have been suggested. Datasets can be divided according to the environment in which they were collected: constrained indoor [25]; unconstrained indoor [12], [13], [26], [27]; and outdoor settings [22]. Compared with early environments [25], [28], recently released datasets [22], [29] are less biased and have improved complexity with a large scale, which makes them suitable for training and evaluation. In this section, we introduce some important datasets.

MPIIGaze [12] is the first in-the-wild dataset comprising 213,659 images, which were collected from 15 subjects in natural daily events. This dataset was generated by showing random points to subjects. It provides not only binocular images but also landmark of the eyes, 2D and 3D gaze, 3D head pose and annotation about the 3D center of the eye. Zhang et al. [13] suggested MPIIFaceGaze derived from the motivation that considering the entire face made gaze estimation more accurate and appended additional landmark annotation of faces. However, it has a limitation in that most of the head pose covered by MPIIFaceGaze is the front view and it has a small camera–subject distance, which makes it inappropriate for remote gaze estimation. Gaze360 [22] is a large-scale dataset collected from 238 subjects with a wide range of head poses and gaze directions. It was collected in unconstrained environments, both indoor and outdoor, covering the entire horizontal range of 360°. ETH-XGaze [30] is another large-scale high-resolution dataset collected from constrained indoor environments. It was collected from 110 subjects using 18 DSLR cameras and adjustable illuminations.

In this work, we conducted various experiments using ETH-XGaze, which includes more diverse head poses and gaze range compared to other datasets. Inspired by the robustness of the model trained with this dataset, we acquired various gaze direction vectors for multiple subjects.

D. FUNCTION FITTING METHODS

Function fitting or curve fitting which fits a function (or curve) and interpolates and extrapolates a set of data without exact equation was used in many previous studies. Especially, when there is a desirable shape for the fitted curve, curve

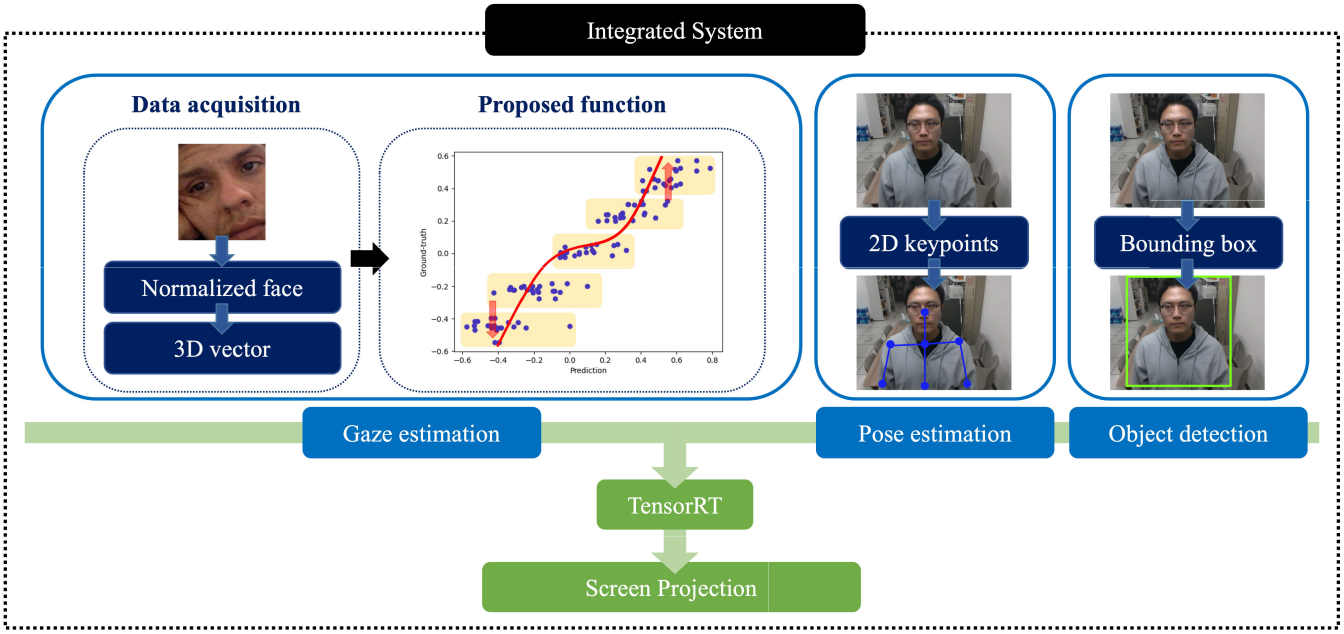

FIGURE 1. Overall framework of the proposed integrated gaze tracking system. For gaze estimation, we propose the symmetric angle amplifying function (SAAF) through the data-acquisition step. For pose estimation and object detection, we employed widely used off-the-shelf models. All neural networks included in the integrated system were optimized through the TensorRT framework. The processed gaze vector output is converted to the screen coordinate and projected to the screen.

fitting methods often becomes a solution. One of the most representative methods for curve fitting is the Bezier curve fitting method. When data points are given without the equation, the Bezier curve fitting method finds the fitted curve and allows interpolation and extrapolation by specifying the degree and optimizing control points of a curve. Recent works on trajectory planning [31], [32] adopted Bezier curve fitting method to generate and plan trajectory and velocity, and Ueda et al. [33] used Bezier curve segments to approximate the boundaries of a point cloud. Another method which is based on basic polynomial method, divided the data into different intervals and fit polynomial function for each divided intervals, or segments. The key idea was to segment intervals. In [34], it used optimization based on l_0 penalized least square regression. In this paper, we compared our proposed fitting function (SAAF) to the forementioned two methods to prove the effectiveness of our proposed function.

III. PROPOSED METHOD

In this section, we explain the proposed method in the top–down approach, starting from the integrated system outline, gaze estimation, and present our SAAF, which makes accurate gaze tracking possible in given conditions. In addition, for a smooth interaction between a user and the edge device, we apply the inference optimizer to deep neural networks included in our systems. This guarantees the inference speed to some extent, enabling undisturbed interaction between the gaze module and the user. To enhance interaction experience, we propose the center gravity function, which pulls gaze coordinates to the center of the predefined regions.

A. INTEGRATED SYSTEM

Our integrated gaze tracking system comprises three modules: a pose estimation module, an object detection module, and a gaze estimation module, which have been optimized for performance as shown in Fig. 1. For the object detection module, we adopted the popular model, YOLOv3 [35], to track multiple users, allowing us to give priority to particular users. To ensure precise pose estimation, we employed HRNet [36], the widely used top–down pose estimation network. Then, the outputs of the pose estimation network were used directly for gaze estimation, generating a 3D face model of the subject. In particular, we used two keypoints corresponding to the ears of a person to find the bounding box containing the face of the user. For the gaze estimation itself, we employed ResNet [7], a simple deep learning network.

B. GAZE ESTIMATION

In practice, the pre-processing and post-processing modules have a considerable impact on gaze estimation performance, which leads to a substantial difference in user experience. Pre-processing in gaze estimation refers to face normalization [37], which is a process of making the appearance of the human face pose independent. To define rotation of the head, we need to model a 3D head pose coordinate system.

The coordinate system is depicted in Fig. 2. It is defined using three parts of the human face: two eyes and the mouth. The x-axis is aligned with the line connecting the midpoints of the eyes, which is depicted by a blue circle on the left side in Fig. 2. The y-axis is an axis contained in the plane stretched perpendicular to the x-axis, pointing in a direction

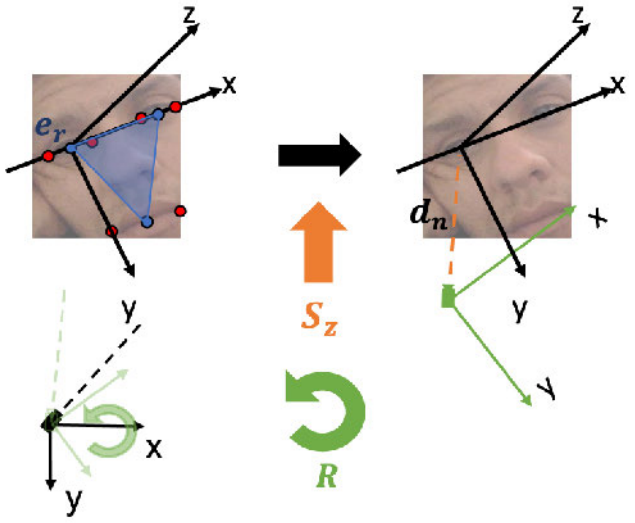

FIGURE 2. 3D face model and normalization procedure. (Left) Definition of human 3D face model and finding rotation matrix (R) for normalization (in green). (Right) Scaling matrix (S_z) in the direction of the z-axis and the final normalized camera coordinate system.

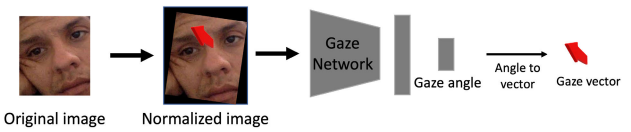

FIGURE 3. Overall pipeline of gaze estimation. Given an input image, we normalize the image using the 3D face and camera model. The gaze network provides the output gaze vector depicted in red.

from the eye to the mouth. The z-axis is perpendicular to the face plane (blue triangle in Fig. 2), pointing backward from the face. After defining the model, the face can be normalized according to two coordinate systems: head pose coordinate system and camera coordinate system. First, we find the rotation matrix, $\boldsymbol{R}$ which rotates camera to look at the eye (e_r) along its z-axis. The scaling matrix S_z locates the camera at a distance d_n from the eye.

Through pre-processing, the head pose-independent image is obtained. After normalization, the gaze estimation network provides the gaze output in 3D vector format. The overall procedure is depicted in Fig. 3. When using the output gaze vector without post-processing, gaze tracking can become unstable because human eyes move rapidly and restively. To mitigate the instability of human eyes, we collected gaze vectors from N frames and used the mean value of the gaze vector for gaze tracking.

C. DATA ACQUISITION

Before designing a fitting function, obtaining the screen coordinate vector and gaze data (i.e. pitch, yaw) of each subject is necessary. First, by using an RGB camera, the screen coordinate system is depicted in Fig. 4. The head position vector of the k-th subject, $(x_k^{sc}, y_k^{sc}, z_k^{sc})$, is the coordinate of the middle of the forehead, i.e., between two pupils, relative

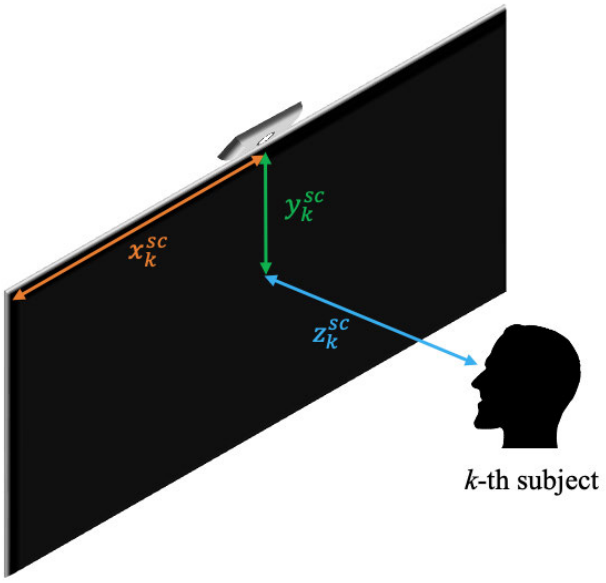

FIGURE 4. Definition of the screen coordinate system. We define the top-left corner of the screen as the origin of the screen coordinates.

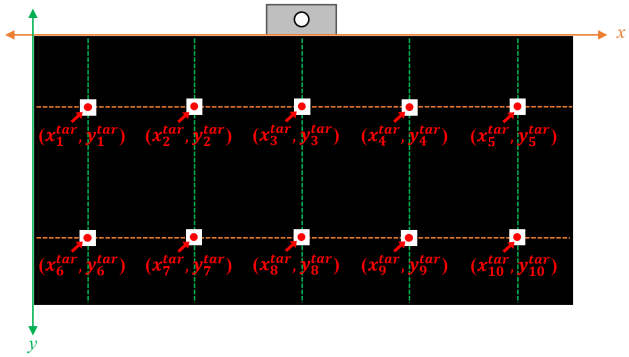

FIGURE 5. Ten targets on the screen with the coordinates. Ten targets were devised to cover the entire screen.

to the upper left corner of the screen. To obtain the gaze data of each subject, various coordinates on the screen must be obtained. Therefore, as shown in Fig. 5, we evenly distributed the position of 10 targets on the screen and measured the x- and y- axis coordinates of the center point of these targets, $(x_1^{tar}, y_1^{tar}), (x_2^{tar}, y_2^{tar}), \ldots, (x_{10}^{tar}, y_{10}^{tar})$, which are marked with red letters. Then, using the screen coordinate vectors $(x_k^{sc}, y_k^{sc}, z_k^{sc})$ obtained above, we can obtain the ground-truths for the pitch and yaw of 10 targets of each subject. The equations for calculating the ground-truths are as follows:

$$\alpha_k^{GT} = arctan(\frac{y_i^{tar} - y_k^{sc}}{z_k^{sc}}) + \rho, \quad (1)$$

$$\gamma_k^{GT} = arctan(\frac{x_i^{tar} - x_k^{sc}}{z_k^{sc}}), \quad (2)$$

where $arctan$ returns a value in the range of $-\pi/2$ to $\pi/2$, $i = 1, \ldots, 10$ is the index of each target, and ρ is the angle between the camera and the screen in radians. The

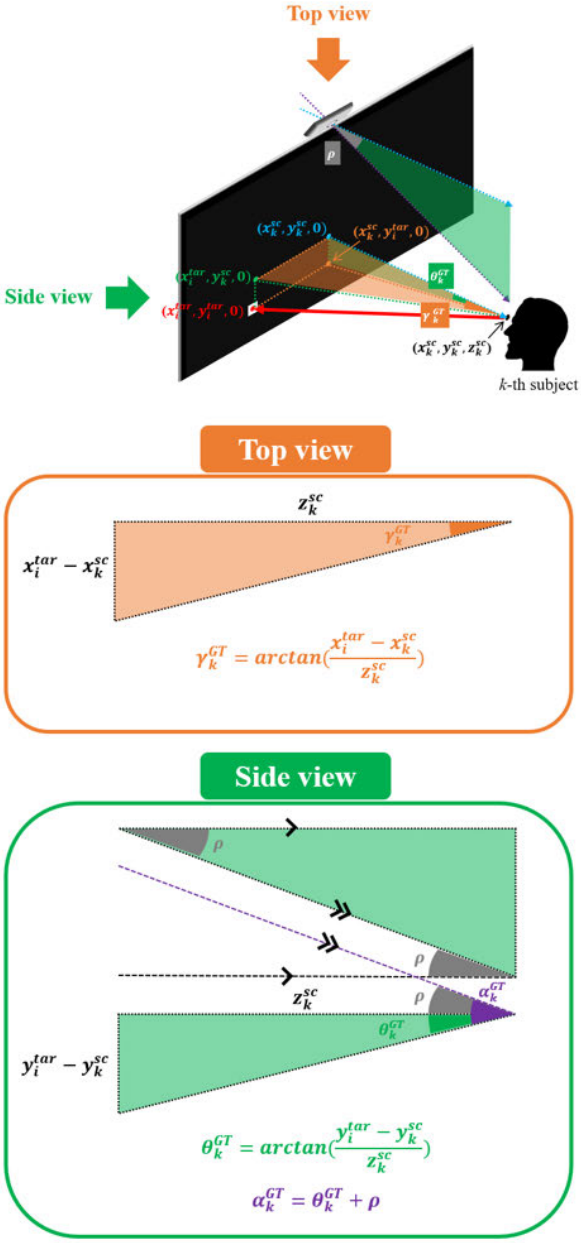

FIGURE 6. Geometric modeling structure of gaze in our work. Derivation of yaw can be easily represented in the top view, and for the geometric meaning of the pitch, we depict the side view.

geometry-based interpretations used in (1) and (2) are from the work of Gudi et al. [5], and here we shortly describe the core concepts in Fig. 6. In Fig. 6, we depict a case where a subject is looking at the target on the screen and calculate pitch and yaw using simple trigonometry.

Meanwhile, the predictions for pitch and yaw can be obtained in the process of making the subjects look at the targets. In particular, standing a certain distance away from the screen, subjects stare at 10 targets on the screen until some numbers of frames are stacked. The face of each frame is recognized through an off-the-shelf pose-estimation network, and the gaze vector of the k-th subject $(\alpha_k^{pred}, \gamma_k^{pred})$ are obtained using ResNet [7] pre-trained on ETX-XGaze [30]. Then, after obtaining the average of all gaze vectors for 100 frames, which is defined as $g = (x_k^{gaze}, y_k^{gaze}, z_k^{gaze})^T$, we can calculate the following:

$$\alpha_k^{pred} = arctan2(y_k^{gaze}, -z_k^{gaze}), \tag{3}$$

$$\gamma_k^{pred} = arctan2(-x_k^{gaze}, -z_k^{gaze}), \tag{4}$$

where *arctan*2 receives the relative coordinates between the two points and returns a value in the range of $-\pi$ to π. The geometric derivations of (3) and (4) are shown in Fig. 8. We assume a right-handed coordinate system for the camera coordinate system; therefore, the sign for each element of the gaze vectors is dependent on the orientations of the coordinate system.

This process was performed for all subjects. Accordingly, we acquired the screen coordinate vector and ground-truth and prediction of the gaze vector corresponding to each target for all subjects through the abovementioned processes.

D. SYMMETRIC ANGLE AMPLIFYING FUNCTION

Since unspecified people use a system in real-world situations, devising a function for gaze calibration using data obtained from a small number of people is necessary. This calibration function should generate gaze from any user to hit the target. Hence, we designed the calibration function SAAF based on the basis of previously collected data such as the screen coordinates, ground-truth, and the prediction of pitch and yaw. Fig. 7 shows the data points (in blue) from 10 people using the targets as defined in the previous section. Therefore, the data points can be grouped into five clusters, which is reasonable because the targets divide the entire screen into five parts in the horizontal direction. From data, we reveal that as the absolute value of the yaw angle increases, more points are scattered. This result corresponds with our intuition that as the movement of the eyeball increases, the effect of person-dependent eye movement also increases. In addition, we found that when a subject is staring at the edge part of the screen, the appearance perceived in the camera is almost the same as that of the neighborhood region. To circumvent the issues, we propose a novel fitting function, SAAF, as shown in the Fig. 7. The function has the greatest influence at the extremes where red arrows are located. We assumed that as the screen has its limit, the farmost regions can be reached by moving the gaze coordinates as far as possible. Therefore, this function amplifies the yaw angle as it increases. By using the proposed fitting function, users are able to obtain their gaze coordinates to the leftmost and rightmost regions of the screen. For the pitch, we used the same approach to find the function that best fit the pitch data. The proposed function F can be formulated as follows:

$$(x, y) = F(\boldsymbol{c}, \boldsymbol{g}, \rho, \boldsymbol{R_{cam}}, \boldsymbol{T}), \tag{5}$$

where $\boldsymbol{c}$ indicates the center coordinate of a recognized user's face in camera coordinate, $\boldsymbol{g}$ indicates the 3D gaze vector

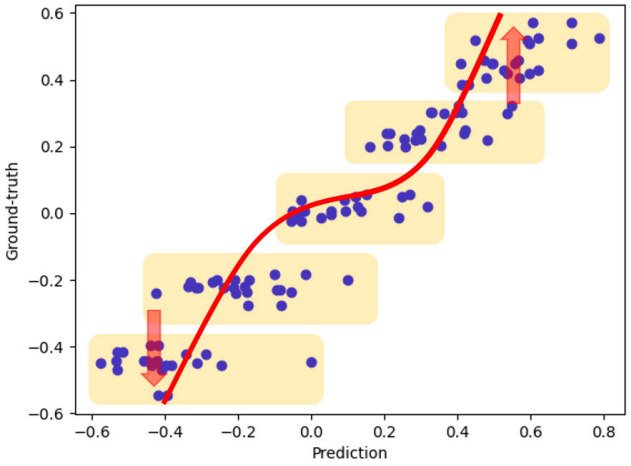

FIGURE 7. Proposed SAAF. The data points that are indicated by blue circles represent the yaw angle of the subjects in radians. The two red arrows show the effect of SAAF on the large angles (edge area of the screen).

obtained from the pretrained gaze estimation model, ρ indicates the vertical pitch angle between the camera and the screen in radians, and $\boldsymbol{R_{cam}}$ and $\boldsymbol{T}$ indicate the rotation matrix and the translation vector of the camera, respectively. After obtaining $v = (\alpha, \gamma)$ from $\boldsymbol{g}$, we can fit v to $v_{fit} = (\alpha_{fit}, \gamma_{fit})$ as follows:

$$\alpha_{fit} = C \times \alpha^2, \tag{6}$$

$$\gamma_{fit} = arcsin(D \times (\gamma - E)), \tag{7}$$

where C, D and E are constants that can be modified to fit the collected data. In addition, to calculate $sc^{pred} = (x_k^{\hat{sc}}, y_k^{\hat{sc}}, z_k^{\hat{sc}})$, the 3D screen coordinate of the user, the following equation is required:

$$sc^{pred} = \boldsymbol{R_{cam}}\boldsymbol{c}^T + \boldsymbol{T} \tag{8}$$

Therefore, with v_{fit} and the predicted screen coordinates sc^{pred}, we can obtain the gaze coordinate (x, y) as the follows:

$$x = tan(\hat{\gamma}) \times z_k^{\hat{sc}} + x_k^{\hat{sc}}, \tag{9}$$

$$y = tan(\hat{\alpha} - \rho) \times z_k^{\hat{sc}} + y_k^{\hat{sc}} \tag{10}$$

E. USER EXPERIENCE OPTIMIZATION

This section discusses two different types of optimization. The first optimization is the deep learning-based optimization, which is performed by using a well-known optimization framework. The designed system contains three deep learning networks: object detection, pose estimation, and gaze estimation. As we mentioned at the beginning of this section, immediate reaction is critical for the user experience, particulary for the gaze tracking. Since deep learning networks are chained to each other in series, all the networks should work as fast as possible. To achieve this goal, we used TensorRT to make the networks lighter and faster.

The second optimization that we propose is the center gravity function. In many cases, the screen is divided into a fixed

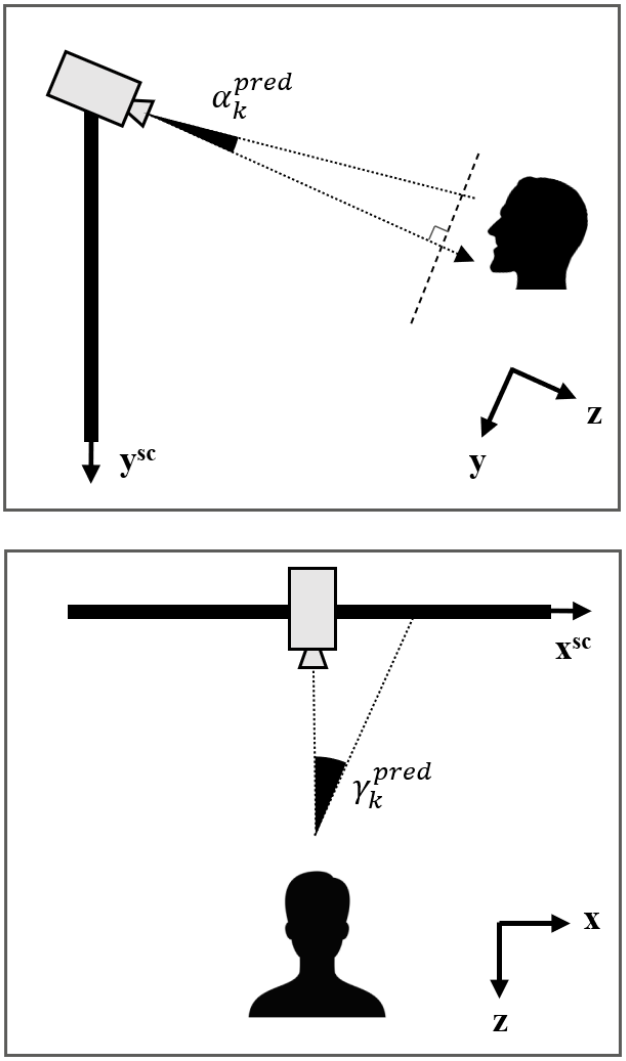

FIGURE 8. Geometric derivation of pitch and yaw. Note that we use a right-hand-side coordinate system for the camera coordinate system.

number of regions in the form of a grid. Since the appearance and mechanism of human eyes are different across the individuals, the gaze vector inferred from the gaze estimation model could be different. This is the motivation of personal calibration. However, as our system is for the use in public, time-consuming personalized calibration is redundant. This results in provision of misguided feedback to the users and a feeling that the gaze is off the target. To solve this problem, we propose the center gravity function, which pulls the gaze coordinates projected on the screen to the predefined center of the grid when the user fixes their gaze for some amount of time. The algorithm procedure is described in Algorithm 1: lines 5 – 6 make the gaze coordinate (x, y) to move toward the center of the clicked region as the accumulated time subject is looking at a certain region, which is implemented as an increase in the length of a stack. The denominators in the equations can be interpreted as an annealing term, which gives the smoothing effect of the coordinates on the

screen. The effect of this algorithm will be described in the experiment section.

Algorithm 1 Center Gravity Function Algorithm

1: **function**CENTEGRAVITYFUNCTION(x, y, cx, cy, l)
2: ▷ x, y: current gaze, cx, cy: center of the current region, l: length of the stack
3: $x = x + \frac{cx-x}{10\exp(\frac{-l}{10})+1}$
4: $y = y + \frac{cy-y}{10\exp(\frac{-l}{10})+1}$
5: **return** x, y
6: **end function**

IV. EXPERIMENTS

This section validates methods presented in the proposed methods section. First, we describe the experimental settings. Second, in the following subsections, we present two quantitative results using F1-score and click accuracy, which show the effect of the proposed function. Finally, we analyse the qualitative results to show the effectiveness of SAAF and the center gravity function.

A. EXPERIMENTAL SETTINGS

The experimental setup is presented in Fig. 9. A large screen with a size of 55 inches is the main difference from other studies and applications, which have considered a small screen on a mobile or desktop monitor. We captured subjects with the RGB sensor of the RealSense D435 camera [38], which was located at the center of the screen. The camera was connected to the NVIDIA Jetson AGX Xavier Developer Kit [39], which is an edge-computing device. To simulate various illumination conditions, we located one light source at the top of the screen, and no other additional control for the ambient light was implemented. For a precise comparison between the models, we changed the composition of subjects from the data-acquisition phase while maintaining gender ratio. We employed 10 people for the experiments: seven males and three females. Data collection and experimentation for this study were conducted after obtaining consent from the participants. For the experiments, two screen settings, which divided the screen into six and eight regions, were used as shown in Fig. 11.

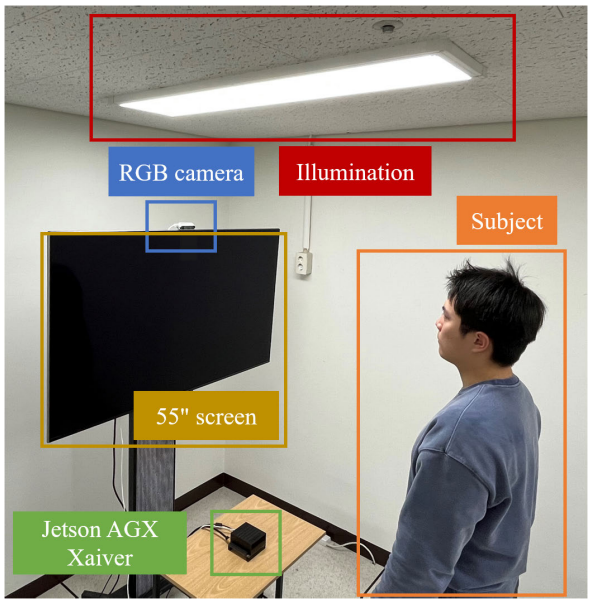

FIGURE 9. Experimental setup for the proposed method. With a large screen, RGB camera, and an edge-computing device, the subject looks at the screen under the indoor illumination.

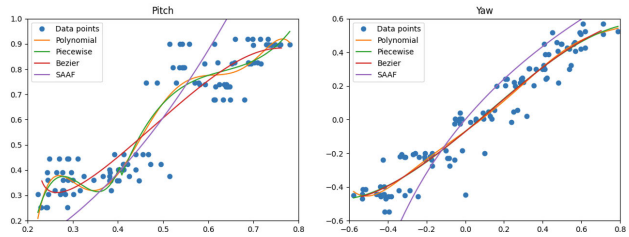

FIGURE 10. Comparisons of the proposed function and other functions. We visualized the results of applying each function to both cases pitch (left) and yaw (right). For both end regions where eye tracking is difficult, SAAF amplifies the gaze value more than other functions.

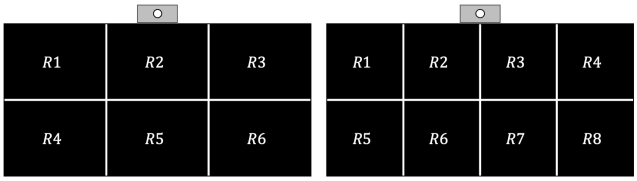

FIGURE 11. Two screen settings used in the experiment. The experiment was conducted in six (left) and eight (right) regions.

B. QUANTITATIVE RESULTS

We compared five models for post-processing the gaze to verify the performance of the proposed method. First, the term "naive model" refers to the baseline gaze tracking model without any post-processing. The second model is the polynomial model, which is a polynomial function fitted to our data from the acquisition process. We chose the best degree for the polynomial by using the R-squared value, which indicates the explainability of the given function. The third model is piecewise polynomial model from the previous work [34]. In the work, it proposed the method for finding the best fitting curve through segmenting intervals and fit each interval with the p-th polynomial fitting model. Since the work provided the source code, we used the code for the experiment. The fourth model is Bezier curve, which is widely used in computer graphics and design of smooth curves of shapes. Recently, it is used for path and velocity planning [31], [32] or fitting the boundary point cloud [33]. We chose the Bezier curve as a reference for comparison because its ability to align its shape to given points of desired shape is similar to the motivation behind our proposed model. Finally, the fifth model is the proposed model SAAF.

We also tested our model under two different conditions: static condition, where the subject is staring at targets on

the screen, and the real-time condition, where the interaction between the subject and gaze tracking module is active. Therefore, two metrics for these different groups were used. For the first condition, we compared the region classification performance among the five different models. Since we measured the classification ability of the models with grid-shaped regions, we used multi-label classification metrics: precision, recall, and F1-score. For the second condition, we proposed a new metric called click accuracy, which measures the number of hits (click at the correct region) out of total clicks. Thus, click accuracy represents the performance of the model in real time. The definition of all the metrics we used in this work are listed below:

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}},$$

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}},$$

$$\text{F1-Score} = \frac{2 \times \text{Recall} \times \text{Precision}}{\text{Recall} + \text{Precision}},$$

$$\text{Click Accuracy} = \frac{\#\text{ of hits}}{\#\text{ of total clicks}}.$$

In our experiment, true positive (TP) is the case where the gaze is located at the target region, while false positive (FP) accounts for the gaze that should not be in the target region. False negative (FN) is the case where the gaze should appear in the target region, but it did not. By using the terminologies defined above, we can interprete precision as the fraction of positive identification that was actually correct, and the recall as the proportion of correctly identified positives from the actual positives. F1-Score which is expressed as harmonic mean of precision and recall, combines precision and recall into a single metric, since there exists a trade-off between precision and recall. A model will obtain a high F1-Score if both precision and recall are high.

1) REGION CLASSIFICATION

We tested the region classification among the five models in two settings: one with six regions and the other with eight regions. The results are presented in Tables 1, and 2.

As revealed by Table 1, since classification tasks on six divided regions are relatively easy, performance difference with other methods was not prominent. However, it was confirmed that our approach achieved the best performance in terms of F1 score. In the case of eight regions, we obtained consistent results in the Table 2, as the proposed model provided best results for five out of eight regions. Bezier curve outperformed our model in upper regions (region 1, 2 and 4), close to the camera installed in the top middle of the screen. However, for all the lower regions (region 5–8) and region 3, our model provided the best F1 scores. This result corresponds with motivation for our proposed method, compensation of the eye movement.

TABLE 1. Classification performance of each method (R = 6).

		Precision	Recall	F1-score
Naive	Region 1	0.710	0.695	0.703
	Region 2	0.539	0.847	0.659
	Region 3	0.577	0.918	0.709
	Region 4	0.915	0.505	0.651
	Region 5	0.965	0.635	**0.766**
	Region 6	0.895	0.582	0.705
Poly	Region 1	0.997	0.508	0.673
	Region 2	0.840	0.680	0.751
	Region 3	0.832	0.717	0.770
	Region 4	0.730	0.935	0.820
	Region 5	0.672	0.848	0.750
	Region 6	0.672	0.857	0.753
Piecewise	Region 1	0.993	0.663	0.795
	Region 2	0.857	0.688	0.763
	Region 3	0.982	0.630	0.768
	Region 4	0.714	0.718	0.716
	Region 5	0.530	0.998	0.692
	Region 6	0.678	0.675	0.677
Bezier	Region 1	0.994	0.793	**0.882**
	Region 2	0.877	0.810	**0.842**
	Region 3	0.991	0.697	0.818
	Region 4	0.807	0.752	0.778
	Region 5	0.598	0.995	0.747
	Region 6	0.738	0.722	0.730
SAAF	Region 1	1.000	0.755	0.860
	Region 2	0.836	0.783	0.809
	Region 3	0.905	0.853	**0.878**
	Region 4	0.848	0.867	**0.857**
	Region 5	0.647	0.918	0.759
	Region 6	0.838	0.775	**0.805**

2) REAL-TIME CLICK ACCURACY

The region classification presented in the previous subsection measured the gaze accuracy under the static condition without direct feedback. However, to evaluate the performance under the practical condition, we designed an evaluation method considering immediate feedback. The evaluation procedure is as follows. First, a random number indicating the target region is generated at each time. Then, when a subject is looking at the target in that region and the gaze coordinate stays in some region for more than a predefined duration, a click is triggered and we record the number of correct clicks each time, which we refer to as hit. The results are presented in Table 3, which shows the preferred performance of the proposed model. These results reveal that the proposed model outperforms other models under practical conditions. The performance of the Bezier method was close to the proposed method SAAF, which corresponds to the result in Table 1 and Table 2.

C. QUALITATIVE RESULTS

We compared the naive model and the proposed model to demonstrate the superiority of the proposed method qualitatively. The polynomial model was not visualized due to its poor performance, so we compared only two models.

The subjects were asked to look at the center of each region of the screen, depicted in Fig. 11; and the actual gaze is

TABLE 2. Classification performance of each method (R = 8).

		Precision	Recall	F1-score
Naive	Region 1	0.664	0.713	0.688
	Region 2	0.564	0.787	0.657
	Region 3	0.584	0.710	0.641
	Region 4	0.598	0.933	0.729
	Region 5	0.927	0.382	0.541
	Region 6	0.736	0.552	0.630
	Region 7	0.635	0.317	0.423
	Region 8	0.641	0.702	0.670
Poly	Region 1	1.000	0.180	0.305
	Region 2	0.501	0.410	0.451
	Region 3	0.874	0.510	0.644
	Region 4	0.882	0.625	0.732
	Region 5	0.634	0.450	0.526
	Region 6	0.317	0.585	0.411
	Region 7	0.403	0.658	0.500
	Region 8	0.599	0.912	0.723
Piecewise	Region 1	0.997	0.588	0.740
	Region 2	0.800	0.767	0.783
	Region 3	0.749	0.720	0.734
	Region 4	1.000	0.563	0.721
	Region 5	0.859	0.435	0.577
	Region 6	0.412	0.895	0.564
	Region 7	0.511	0.692	0.588
	Region 8	0.615	0.550	0.580
Bezier	Region 1	0.997	0.628	**0.771**
	Region 2	0.783	0.792	**0.787**
	Region 3	0.804	0.758	0.780
	Region 4	1.000	0.677	**0.807**
	Region 5	0.870	0.480	0.619
	Region 6	0.423	0.887	0.573
	Region 7	0.525	0.627	0.571
	Region 8	0.616	0.553	0.583
SAAF	Region 1	0.942	0.572	0.712
	Region 2	0.641	0.620	0.631
	Region 3	0.802	0.815	**0.808**
	Region 4	0.978	0.585	0.732
	Region 5	0.713	0.732	**0.722**
	Region 6	0.562	0.760	**0.646**
	Region 7	0.658	0.877	**0.752**
	Region 8	0.710	0.782	**0.744**

TABLE 3. Click accuracy test results.

	Click Accuracy	
	R = 6	R = 8
Naive	0.700	0.650
Poly	0.750	0.613
Piecewise	0.833	0.825
Bezier	0.883	**0.8625**
SAAF	**0.917**	**0.8625**

represented by colored dots in Figs. 12, 13, and 14. Details can be found through the legends of the figures. In addition, the transparency of the points was adjusted to visualize the movement of the gaze. In particular, the transparent points are the early stages of gaze tracking, and the colors of the points become more opaquer over time.

Fig. 12 shows the results of applying the naive model (top) and the proposed model (bottom) in the case of $R = 6$. The naive model has various color points distributed in a specific area, while the proposed model has mostly one-color point distributed in a specific area. Because of this, the gaze

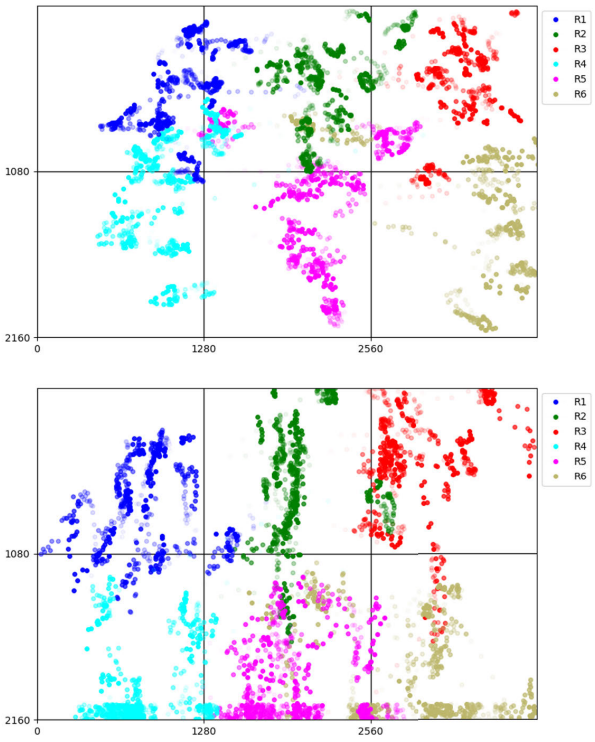

FIGURE 12. Qualitative results of collected gaze when $R = 6$. We visualized the performance of the Naive model (top) and the proposed model (bottom). The transparent points are the early stages of gaze tracking, and the colors of the dots get darker over time.

tracking performance for the proposed model in the case of $R = 6$ is powerful compared to the naive model.

Similarly, Fig. 13 shows the results of applying the naive model (top) and the proposed model (bottom) in the case of $R = 8$. The naive model has diverse color points distributed in a specific area, while the proposed model shows that one color point in a specific area occupies most of the area. This shows that the gaze tracking performance of the proposed model is excellent even in the case of $R = 8$, which is a more challenging situation than in the case of $R = 6$.

In addition, the visualization of demonstrating the effectiveness of the center gravity function is shown in Fig. 14. The description of the color and brightness of the points is consistent with the description shown in Fig. 13. The experiment was conducted in the case of $R = 8$ on a specific subject. In Fig. 14, the upper result is that before applying the center gravity function, and the lower one is that after applying the center gravity function. Both are assumed to have applied SAAF. Thus, the center gravity function is a function that improves the ability of reaching the center of a specific area.

V. IMPLEMENTATION

The gaze tracking module presented in the previous sections is a part of an integrated system. What we refer to as the aggregated system is the set of hardware systems installed in an autonomous driving vehicle for the tourists, and the inte-

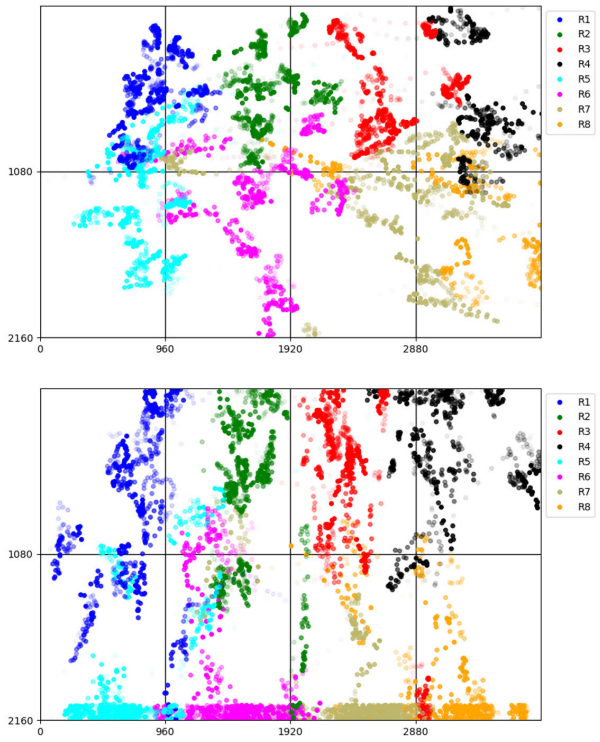

FIGURE 13. Qualitative results of collected gaze when $R = 8$. We visualized the performance of the naive model (top) and the proposed model (bottom). The transparent points are the early stages of gaze tracking, and the colors of the dots get darker over time.

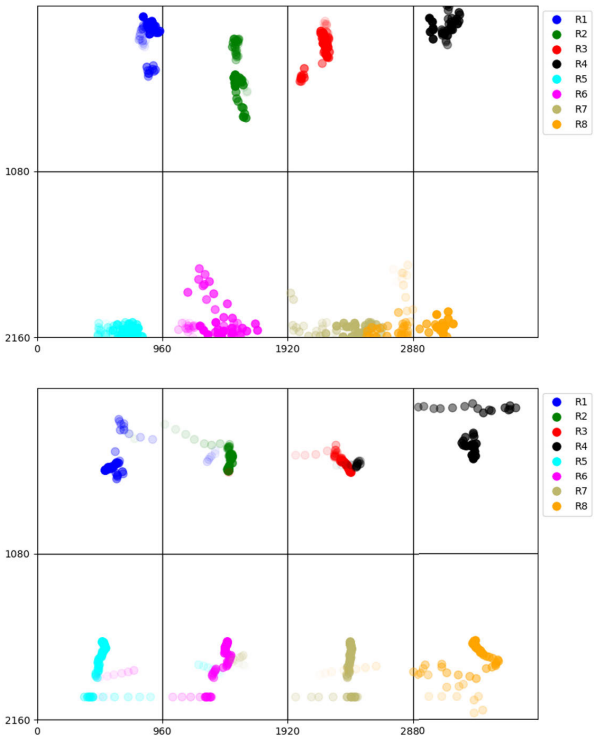

FIGURE 14. Qualitative results of the proposed center gravity function before (top) and after (bottom) the application of the center gravity function. The transparent points are the early stages of gaze tracking, and the colors of the dots get darker over time.

FIGURE 15. Installation of the screen inside the Navya shuttle. (Left) The blueprint of the screen installation. (Right) Actual installation of the two screens in perpendicular position.

FIGURE 16. Installation of the screen inside the Robo shuttle. (Left) The outside view of the Robo. (Right) Actual installation of the screen inside the Robo shuttle.

grated system refers to the software system composed of gaze estimation, pose estimation, and object detection combined with the optimization process. In particular, the aggregated system is installed in the Navya [40], and Robo [41] shuttle buses and it becomes the reason for consideration of its use in the public environment. Fig. 15 and Fig. 16 present that the OLED displays installed in a perpendicular position, which makes enables two users to own two displays respectively.

To show further details for the testing environment inside the vehicle, Navya is represented in Fig. 17. We only show the bird-eye-view of Navya, since two vehicles have almost the same size, but a slight difference in the ratio of width to height. The distance that yielded the best results through experimentation was determined to be between 0.8–1.0 meter from the screen. If the subject is positioned too far from the screen, the gaze tracking performance decreases as the recognition of human eye becomes more difficult. Conversely, if the subject is too close to the screen, using gaze to control the UI becomes redundant as touch can also be used for this purpose. User should be positioned aligned to the center of screens since the subject of the public gaze dataset located at the center of the screen. The position where a person is expected to stand is depicted as the orange box, and the position of displays is indicated in blue region as shown in Fig. 17.

Inside the vehicle, users are provided with the various types of information about the vehicle they are riding, and the local guides through the user interface (UI); for example, vehicle operation information, safety manual, internet web service, tour guide, and even entertainment contents are provided. The UI is controlled by using multimodal inputs from the users. To receive the multimodal inputs from the users, several

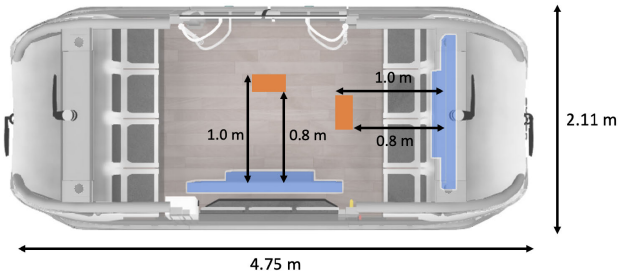

FIGURE 17. Testing environment in bird-eye-view inside the Navya. Two screens are installed in perpendicular position and operate independently. At most two people can utilize gaze tracking module distance within 0.8m - 1.0m from the screen.

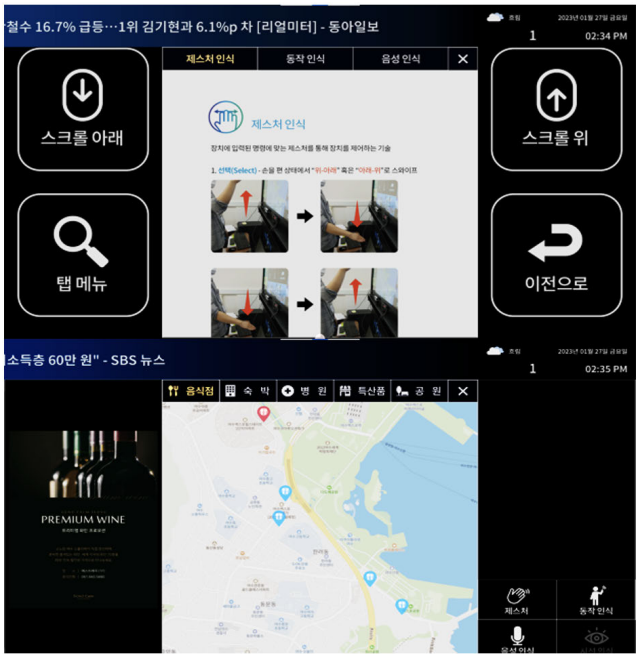

FIGURE 18. Example pictures of the user interface. (Top) Manual for the hand gesture recognition. (Down) Local map for the tourists.

sensors such as touch display, voice sensor, gesture sensor, and RGB camera for gaze tracking are included in the system. Because the entire system comprises many different modules and sensors receiving inputs simultaneously, optimizing the network is essential as we explained in the proposed method section.

VI. CONCLUSION

In this paper, we proposed a function named SAAF, which is designed to solve the gaze-to-screen mapping problem that occurs at the edge part of a large screen. In addition, for better user experience, we optimized our system from two aspects: inference speed and feedback. We implemented the network optimization using TensorRT to achieve low latency. Also, we proposed a center gravity function that compensates for person-dependent movement of the eyes of each user. Our gaze tracking system was implemented as a part of an aggregated system along with other modules, such as voice assistance, gesture recognition, and touch screens, and installed in an autonomous vehicle that serves as a tour shuttle. Although we achieved accurate gaze tracking performance in a large screen, some limitations remain. For example, because the gaze dataset was collected by capturing the subject located at the center of the screen, the space in which a user can utilize the system is constrained. For the future research, we expect a dataset containing position of subjects and the corresponding gaze to resolve the abovementioned issue, providing a larger space to the users.

ACKNOWLEDGMENT

(Joseph Kihoon Kim, Junho Park, Yeon-Kug Moon, and Suk-Ju Kang contributed equally to this work.)

REFERENCES

[1] X. Zhang, S. Yuan, M. Chen, and X. Liu, "A complete system for analysis of video lecture based on eye tracking," *IEEE Access*, vol. 6, pp. 49056–49066, 2018.
[2] H. Liu and I. Heynderickx, "Visual attention in objective image quality assessment: Based on eye-tracking data," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 21, no. 7, pp. 971–982, Jul. 2011.
[3] Y. Ang, P. S. Sulaiman, R. W. O. K. Rahmat, and N. M. Norowi, "Swing-in-place (SIP): A less fatigue walking-in-place method with side-viewing functionality for mobile virtual reality," *IEEE Access*, vol. 7, pp. 183985–183995, 2019.
[4] L. Jigang, B. S. L. Francis, and D. Rajan, "Free-head appearance-based eye gaze estimation on mobile devices," in *Proc. Int. Conf. Artif. Intell. Inf. Commun. (ICAIIC)*, Feb. 2019, pp. 232–237.
[5] A. Gudi, X. Li, and J. van Gemert, "Efficiency in real-time webcam gaze tracking," in *Proc. Eur. Conf. Comput. Vis. Workshops (ECCVW)*, 2020, pp. 529–543.
[6] S. A. Sabab, M. R. Kabir, S. R. Hussain, H. Mahmud, H. A. Rubaiyeat, and Md. K. Hasan, "VIS-iTrack: Visual intention through gaze tracking using low-cost webcam," *IEEE Access*, vol. 10, pp. 70779–70792, 2022.
[7] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 770–778.
[8] W. Li, Q. Dong, H. Jia, S. Zhao, Y. Wang, L. Xie, Q. Pan, F. Duan, and T. Liu, "Training a camera to perform long-distance eye tracking by another eye-tracker," *IEEE Access*, vol. 7, pp. 155313–155324, 2019.
[9] X. Zhang, Y. Sugano, M. Fritz, and A. Bulling, "Appearance-based gaze estimation in the wild," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 4511–4520.
[10] K. Krafka, A. Khosla, P. Kellnhofer, H. Kannan, S. Bhandarkar, W. Matusik, and A. Torralba, "Eye tracking for everyone," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 2176–2184.
[11] S. Park, A. Spurr, and O. Hilliges, "Deep pictorial gaze estimation," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, Sep. 2018, pp. 721–738.
[12] X. Zhang, Y. Sugano, M. Fritz, and A. Bulling, "MPIIGaze: Real-world dataset and deep appearance-based gaze estimation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 41, no. 1, pp. 162–175, Jan. 2019.
[13] X. Zhang, Y. Sugano, M. Fritz, and A. Bulling, "It's written all over your face: Full-face appearance-based gaze estimation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jul. 2017, pp. 2299–2308.
[14] K. Wang, R. Zhao, and Q. Ji, "A hierarchical generative model for eye image synthesis and eye gaze estimation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 440–448.
[15] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," *Commun. ACM*, vol. 60, no. 6, pp. 84–90, May 2017.
[16] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," 2014, *arXiv:1409.1556*.
[17] A. Newell, K. Yang, and J. Deng, "Stacked hourglass networks for human pose estimation," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, Oct. 2016, pp. 483–499.
[18] S. Jyoti and A. Dhall, "Automatic eye gaze estimation using geometric & texture-based networks," in *Proc. 24th Int. Conf. Pattern Recognit. (ICPR)*, Aug. 2018, pp. 2474–2479.

[19] C. Palmero, J. Selva, M. Ali Bagheri, and S. Escalera, "Recurrent CNN for 3D gaze estimation using appearance and shape cues," 2018, *arXiv:1805.03064*.
[20] X. Zhou, J. Lin, J. Jiang, and S. Chen, "Learning a 3D gaze estimator with improved itracker combined with bidirectional LSTM," in *Proc. IEEE Int. Conf. Multimedia Expo. (ICME)*, Jul. 2019, pp. 850–855.
[21] K. Wang, H. Su, and Q. Ji, "Neuro-inspired eye tracking with eye movement dynamics," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 9823–9832.
[22] P. Kellnhofer, A. Recasens, S. Stent, W. Matusik, and A. Torralba, "gaze360: Physically unconstrained gaze estimation in the wild," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 6911–6920.
[23] C. Palmero Cantarino, O. V. Komogortsev, and S. S. Talathi, "Benefits of temporal information for appearance-based gaze estimation," in *Proc. ACM Symp. Eye Tracking Res. Appl.*, Jun. 2020, pp. 1–5.
[24] S. Nonaka, S. Nobuhara, and K. Nishino, "Dynamic 3D gaze from afar: Deep gaze estimation from temporal eye-head-body coordination," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 2182–2191.
[25] B. A. Smith, Q. Yin, S. K. Feiner, and S. K. Nayar, "Gaze locking: Passive eye contact detection for human-object interaction," in *Proc. 26th Annu. ACM Symp. User Interface Softw. Technol.*, Oct. 2013, pp. 271–280.
[26] T. Fischer, H. J. Chang, and Y. Demiris, "RT-GENE: Real-time eye gaze estimation in natural environments," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, Sep. 2018, pp. 334–352.
[27] Q. Huang, A. Veeraraghavan, and A. Sabharwal, "TabletGaze: Dataset and analysis for unconstrained appearance-based gaze estimation in mobile tablets," *Mach. Vis. Appl.*, vol. 28, nos. 5–6, pp. 445–461, Aug. 2017.
[28] K. A. Funes Mora, F. Monay, and J.-M. Odobez, "Eyediap: A database for the development and evaluation of gaze estimation algorithms from rgb and RGB-D cameras," in *Proc. Symp. Eye Tracking Res. Appl. (ETRA)*, 2014, pp. 255–258.
[29] S. Park, E. Aksan, X. Zhang, and O. Hilliges, "Towards end-to-end video-based eye-tracking," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, Aug. 2020, pp. 747–763.
[30] X. Zhang, S. Park, T. Beeler, D. Bradley, S. Tang, and O. Hilliges, "ETH-XGaze: A large scale dataset for gaze estimation under extreme head pose and gaze variation," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2020, pp. 365–381.
[31] T. Chen, Y. Cai, L. Chen, and X. Xu, "Trajectory and velocity planning method of emergency rescue vehicle based on segmented three-dimensional quartic Bezier curve," *IEEE Trans. Intell. Transp. Syst.*, vol. 24, no. 3, pp. 3461–3475, Mar. 2023.
[32] A. Vinayak, M. A. Zakaria, K. Baarath, and A. P. P. A. Majeed, "A novel Bezier curve control point search algorithm for autonomous navigation using N-order polynomial search with boundary conditions," in *Proc. IEEE Int. Intell. Transp. Syst. Conf. (ITSC)*, Sep. 2021, pp. 3884–3889.
[33] E. K. Ueda, M. S. G. Tsuzuki, and A. Barari, "Piecewise Bézier curve fitting of a point cloud boundary by simulated annealing," in *Proc. 13th IEEE Int. Conf. Ind. Appl. (INDUSCON)*, Nov. 2018, pp. 1335–1340.
[34] J. Duan, Q. Wang, and Y. Wang, "HOPS: A fast algorithm for segmenting piecewise polynomials of arbitrary orders," *IEEE Access*, vol. 9, pp. 155977–155987, 2021.
[35] J. Redmon and A. Farhadi, "YOLOv3: An incremental improvement," 2018, *arXiv:1804.02767*.
[36] K. Sun, B. Xiao, D. Liu, and J. Wang, "Deep high-resolution representation learning for human pose estimation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 5686–5696.
[37] Y. Sugano, Y. Matsushita, and Y. Sato, "Learning-by-synthesis for appearance-based 3D gaze estimation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2014, pp. 1821–1828.
[38] (Dec. 2022). *Depth Camera D435*. Accessed: Feb. 15, 2023. [Online]. Available: https://www.intelrealsense.com/depth-camera-d435/
[39] *Deploy Ai-Powered Autonomous Machines at Scale*. Accessed: Feb. 15, 2023. [Online]. Available: https://www.nvidia.com/en-us/autonomous-machines/embedded-systems/jetson-agx-xavier/
[40] (Mar. 2022). *Self-Driving Shuttle for Passenger Transportation*. Accessed: Feb. 15, 2023. [Online]. Available: https://www.navya.tech/en/solutions/moving-people/self-driving-shuttle-for-passenger-transportation/
[41] *Unmanned Solution*. Accessed: Feb. 25, 2023. [Online]. Available: http://www.unmansol.com/sub02_en.html

JOSEPH KIHOON KIM received the B.S. degree from Handong Global University, South Korea, in 2021. He is currently pursuing the M.S. degree with Sogang University. His current research interests include computer vision and generative models.

JUNHO PARK (Graduate Student Member, IEEE) received the B.S. degree from Sogang University, South Korea, in 2022, where he is currently pursuing the M.S. degree. His current research interests include computer vision and deep learning.

YEON-KUG MOON received the B.S. and M.S. degrees in electronics engineering from Inha University, Inchon, South Korea, in 1998 and 2000, respectively, and the Ph.D. degree in biomicrosystem technology engineering from Korea University, Seoul, South Korea, in 2014. Since 2005, he has been the Director of the Data Convergence Platform Reaserch Center, Korea Electronics Technology Institute, South Korea. His research interest includes Metaverse platform.

SUK-JU KANG (Member, IEEE) received the B.S. degree in electronic engineering from Sogang University, South Korea, in 2006, and the Ph.D. degree in electrical and computer engineering from the Pohang University of Science and Technology, in 2011. From 2011 to 2012, he was a Senior Researcher with LG Display, where he was the Project Leader for resolution enhancement and multi-view 3D system projects. From 2012 to 2015, he was an Assistant Professor of electrical engineering with Dong-A University, Busan. He is currently a Professor of electronic engineering with Sogang University. His current research interests include image analysis and enhancement, video processing, multimedia signal processing, circuit design for display systems, and deep learning systems. He was a recipient of the IEIE/IEEE Joint Award for Young IT Engineer of the Year, in 2019.