Magic Gaze: Enabling Seamless Control of IoT Devices Through Eye Tracking

Kenan Bektaş
kenan.bektas@unisg.ch
University of St. Gallen
St.Gallen, Switzerland

Tobias Ettling
tobias.ettling@unisg.ch
University of St. Gallen
St. Gallen, Switzerland

Simon Mayer
simon.mayer@unisg.ch
University of St. Gallen
St.Gallen, Switzerland

Jannis Strecker-Bischoff
jannis.strecker-bischoff@unisg.ch
University of St. Gallen
St.Gallen, Switzerland

Abstract

Hands-free control offers natural and intuitive interaction with devices, particularly in scenarios where traditional input methods are impractical. We introduce an extensible framework that integrates eye tracking, object detection, and gesture recognition to study intended and unintended interactions with Internet of Things (IoT) devices. To develop our framework, we conducted a structured experiment with 9 participants, focusing on identifying natural and intuitive interaction behaviors in different situations. The results showed that users intuitively combined gaze- and head-based gestures, showing the potential of head/gaze combinations as input mechanisms, specifically for directional movements. On this basis, we propose a system for hands-free interaction and control of IoT devices with intuitive gaze- and head-based gestures. We report on our promising findings as well as on limitations with respect to accurately distinguishing intention in real-world conditions. All our code[1] is publicly available, ensuring the reproducibility and extension of our findings.

CCS Concepts

• **Computing methodologies** → **Perception**; **Object detection**; • **Human-centered computing** → **Ubiquitous and mobile computing systems and tools**; **Gestural input**.

ACM Reference Format:
Kenan Bektaş, Tobias Ettling, Simon Mayer, and Jannis Strecker-Bischoff. 2026. Magic Gaze: Enabling Seamless Control of IoT Devices Through Eye Tracking. In *2026 Symposium on Eye Tracking Research and Applications (ETRA '26), June 01–04, 2026, Marrakesh, Morocco.* ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/3797246.3804834

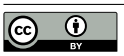
1 Introduction

From physical switches to voice commands, the possibilities of human interaction with devices have evolved significantly and technology continues to push the boundaries of seamless interaction. One promising frontier is the use of gaze as an input modality. Gaze-based interaction capitalizes on the natural human tendency to look at objects of interest (cf. [Bednarik et al. 2012; Bektaş 2020; Bolt 1980, 1981; Velloso et al. 2016]), offering a hands-free and intuitive way to control devices (cf. [Heravian et al. 2019; Istance et al. 2010]). This approach is particularly valuable in contexts where traditional interaction methods may not be feasible, such as accessibility for people with physical disabilities or in environments that require hands-free operation, such as operating rooms or industrial settings. Although eye movements have been extensively studied as a cue for understanding human intention, their potential to control Internet of Things (IoT) devices remains underexplored. Many existing smart devices provide functional Application Programming Interfaces (APIs) that support automation and remote control, but these solutions primarily rely on conventional input methods such as smartphones or voice assistants. Bridging natural human gaze flow—e.g., through tracking gaze-contingent overt attention—and functional capabilities of smart devices hence represents a significant opportunity for innovation.

To enable seamless and intuitive control of IoT devices (cf. [Mayer et al. 2014; Weiser 1999]), we aim to build a meaningful mapping between *device-affordances* and *user-intentions* by leveraging eye tracking; hence, we name our approach *Magic Gaze*. The central idea is to minimize the need for explicit intermediary interfaces and create more natural interaction experiences by detecting gaze patterns (cf. [Bednarik et al. 2012; Bulling and Gellersen 2010; Istance et al. 2010]) and mapping them to specific device interactions while combining with other modalities, such as head movements—think of a smart lamp that users could look at and "nudge brighter" using their head; or a robot that a user could slow down by giving it a skeptical frown.

Implementing such a system poses several challenges: First, detecting and interpreting eye-movement-related gestures (e.g., head nods or shakes while fixating at objects) with high accuracy requires robust algorithms that can handle variations in lighting, user behavior, and device configurations. Second, associating such gestures with specific objects in real time involves precise object detection and event synchronization, ensuring that the system accurately identifies the interaction target. This work makes two contributions: First, we developed a prototype system that combines eye tracking

[1] https://github.com/Interactions-HSG/MagicGaze



ETRA '26, Marrakesh, Morocco

ACM ISBN 979-8-4007-2519-7/2026/06
https://doi.org/10.1145/3797246.3804834

with object detection and head movement tracking to enable gaze-based device control. Second, we evaluated the effectiveness of this system through a user study that demonstrates its applicability in a smart home environment.

2 Related Work

Gaze-based interaction has received significant attention in recent years as an intuitive and hands-free modality for controlling devices. Early studies, such as those by [Belardinelli et al. 2015], explore how gaze strategies are goal-oriented and adapt to object interaction. Similarly, [Brouwer et al. 2009] demonstrate differences in gaze patterns when viewing versus grasping objects and highlighted the nuanced relationship between gaze and intention. [Istance et al. 2010] define spatial layouts for a gaze gesture interface that was particularly suitable for command selection to control a multiplayer online game. [Fuchs and Belardinelli 2021] propose methods to estimate user intention through gaze during shared autonomy tasks, emphasizing its utility in robotics and industrial settings.

Our research builds on the premise that gaze can act as a reliable non-verbal cue to supplement or replace verbal or manual inputs and without presupposing specific interface layouts (cf. [Istance et al. 2010]). For example, eye tracking can be used in augmented reality (AR) environments to predict human activities and provide them with contextual feedback when they interact with physical objects [Bektaş et al. 2023]. Similarly, gaze-enabled AR headsets can be used to facilitate collaboration and mutual understanding through non-verbal communication between humans [Bektaş et al. 2024]. In combination with visual (e.g., [Jocher and Qiu 2024]) or multi-modal (e.g., [Strecker et al. 2023]) frameworks for precise and real-time detection and identification of objects in dynamic settings, this creates a credible pathway for systems that can interpret and act on user intent in multi-device environments; and pupillometry may, on top, provide insights into the cognitive workload of users when they interact with devices [Hostettler et al. 2023]. These findings motivate research on integrating eye tracking with advanced object recognition and human-computer interaction applications, and we see significant opportunities for this combination in IoT systems.

3 Methodology

To enable gaze-based device control, we developed a framework (Figure 1-a) comprising three core components. 1) **Gaze at Object Detection (Section 3.1)** identifies the device the user intends to interact with. 2) **Gaze Gesture Detection (Section 3.2)** captures interaction behavior from gaze and head movements. 3) **Device Control Service (Section 3.3)** defines behavior triggers and execute actions via the device's Thing Descriptions (TDs) [W3C 2023].

Our framework is built on Pupil Core [Kassner et al. 2014], utilizing a lightweight mobile eye tracker that includes two infrared (IR) eye cameras (200 Hz) and a scene camera ($720p$ at $60Hz$) to capture the user's field of view. The open-source Pupil Core software suite is built on an extensible plugin-driven architecture written in Python. We integrate plugins that analyze both the gaze and camera feed to capture user interactions and intentions. However, eye movements can also be non-intentional. To overcome issues such as the Midas Touch problem [Jacob 1995], we orchestrate our custom plugins to create sequence patterns of staged interaction events. For instance, the user can (1) look still at a device, (2) perform a gesture, (3) look still again and (4) look away (cf. [Bednarik et al. 2012; Land and Hayhoe 2001]). We conducted an observational user experiment (see Section 4) to identify intentional interaction patterns that allow fine grained control of various IoT devices. The findings obtained from this experiment allowed us to finetune the mapping between gaze gestures and device affordances and informed the design of **Device Control Service** component in our framework (Figure 1-c).

3.1 Gaze at Object Detection (GOD)

In our setup, we explore the interaction with diverse IoT devices, including a robotic arm, a lamp, a tractor bot, a speaker, and a microphone. Using the front-facing scene camera of the eye tracker, we used the *YOLOv11s* object detection model from the *Ultralytics* repository [Jocher and Qiu 2024] as the foundation and fine-tuned it on a custom dataset created within our lab. The model achieved an accuracy of 99.5% on the test split of the dataset, demonstrating high performance when used in the lab environment where the training data was collected. Although building a highly generalizable model was not our primary objective, it could be achieved through additional measures such as collecting diverse training data across multiple settings, employing data augmentation techniques, or fine-tuning the model for specific contexts.

Running *YOLOv11s* at the frame rate of the scene camera was not feasible for real-time processing. Specifically, the inference time for *YOLOv11s* is approximately $2.5ms$ on a *TensorRT10* (T4) GPU. Therefore, we opted to run inference only upon a trigger event (i.e., a fixation) published by the Pupil Capture's built-in fixation plugin (see Figure 1). The trigger event can be easily configured to calibrate its sensitivity for different users. If a fixation event is published, the GOD plugin starts predicting the bounding boxes around all detected devices (ignoring low confidence results < 0.5):

$$B_i = \{(x_{min}, y_{min}), (x_{max}, y_{max})\}$$

Then it checks if the position of the recent fixation (x_{pos}, y_{pos}) is within one of the detected bounding boxes.

$$x_{min} \leq x_{pos} \leq x_{max} \quad and \quad y_{min} \leq y_{pos} \leq y_{max}$$

As soon as this condition is fulfilled, GOD publishes an event providing details such as the object's identifier, the fixation position, and the bounding box coordinates. Due to Midas Touch problem, a fixation alone may not always reliably determine the object of interest. However, our work relies on the assumption of overt attention, that is, *a user's deliberate interaction or intent to control a device starts with a fixation event.* Now imagine a device that first becomes aware of the (overt) visual attention of a user. In the next step, the device is expected to react (i.e., implement one of its affordances) with respect to the intention of the user.

3.2 Gaze Gesture Detection

In our framework, we focus on detecting three gaze-contingent gestures: (1) head movements during a fixation, (2) blinks while gazing at a device, and (3) saccadic movements in relation with a device of interest. We integrate each method as a plugin within the Pupil Capture software, enabling simultaneous capture of various interaction behaviors. To evaluate the intuitiveness and seamlessness of gesture behaviors, we conducted an experiment to assess

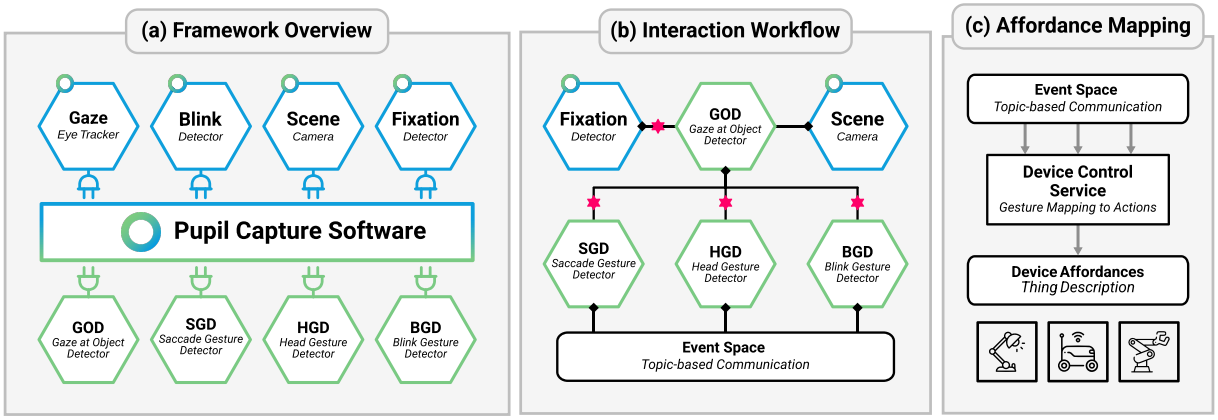


Figure 1: (a) Overview of the Magic Gaze framework. The built-in plugins (blue) include the gaze tracker, blink detector, fixation detector, and the camera stream onto which gaze is mapped. Our custom-developed plugins (green) comprise the *Gaze at Object Detector* (GOD), *Saccade Gesture Detector* (SGD), *Head Gesture Detector* (HGD), and *Blink Gesture Detector* (BGD). (b) General interaction workflow among the plugins. Once a fixation is detected, the Gaze at Object Detector (GOD) identifies known devices in the current scene. If the fixation falls on one of these devices, the gesture detection plugins begin monitoring subsequent gaze behavior. Then, the detected gesture is published to the event space. If no valid gesture is recognized, no event is published and the system waits for the next GOD event. (c) Affordance mapping workflow: The *Device Control Service* (DCS) can subscribe to all gesture events or only a selected subset, and maps these events to device affordances using the corresponding thing description [W3C 2023]. Please see our public repository: https://github.com/Interactions-HSG/MagicGaze

which behaviors felt most natural to participants. The results are presented in Section 4.

3.2.1 Head Gesture Detector. Due to vestibulo-ocular reflex (VOR), the eyes move smoothly in the opposite direction of head motion to stabilize the line of sight in space [Fetter 2007]. This allows us to track head movements from gaze recordings and define head gestures based on movement paths, such as nodding, tilting, or shaking [Špakov and Majaranta 2012]. Our *Head Gesture Detector (HGD)* monitors the point-to-point (ptp) distance for incoming gaze points after a fixation-on-object event is published by the *Gaze at Object Detector* (GOD). For subsequent gaze points, we check if the *ptp* distance falls within the smooth transition range, using the distance to the fixation position for the first gaze point. This ensures accurate detection of head movements while fixating on a stationary object. In contrast to previous solutions [Špakov and Majaranta 2012], the smooth gaze path begins when the gaze leaves a *predefined proximity region* around the initial fixation and ends when the gaze returns to that region, and a fixation is detected (see Figure 2).

Now that we can track head movements while fixating on the object, we can define unique head movements to control a device (Figure 2). We define a head gesture as a smooth path that begins at the fixation position $FO_{pos} = (x_{pos}, y_{pos})$ of the GOD event, moves a minimum distance $\sigma_{norm} = 0.25$ (25% of the normalized coordinate space) away, and ends as soon as the gaze returns within a distance around the FO_{pos} and a fixation is detected. We then calculate the angle between the average gaze position for points outside the minimum distance threshold and the fixation position. Then we map the angles to control signals: up (225° to 315°), down (45° to 135°), left (0° to 45° and 315° to 360°), and right (135° to 225°), considering that the movement is opposite to the head motion.

3.2.2 Blink Gesture Detector. This gesture is relatively simple and has already been used to emulate a button press action [Królak and Strumiłło 2012], which requires explicit mapping of unintentional (passive) and intentional (active) eye blinks. [Stern et al. 1984] found that natural eye blinks typically last between 150 and 400 ms, while voluntary blinks (active) are longer. The built-in Blink Detector (BD) of Pupil Core identifies a blink event based on a drop in pupil detection confidence, which occurs when the eyelids obscure the pupils. Using BD we effectively measure only the onset o_{start}, offset o_{end}, and duration $d = o_{end} - o_{start}$ of the blink events. Our Blink Gesture Detector (BGD) plugin functions as a conditional data relay for the existing BD and uses a threshold ($d > 500$ ms) to separate intentional from unintentional blinks. It is staged after the GOD plugin and is triggered by a fixation-on-object event. When the trigger is published, it listens for incoming blink events and if a blink falls above the duration threshold we publish a blink-at-object gesture event (Figure 1). In order to determine whether the blink is still towards the object we monitor if consecutive gaze points fall within a proximity threshold of the initial fixation position FO_{pos} of the fixation-on-object event (GOD). We designed the BGD to function exclusively as a binary control input, making it suitable for toggle-based interactions.

3.2.3 Saccade Gesture Detection. Saccades have been used in human computer interfaces [Soltani and Mahnam 2016]. Saccade Gesture Detector (SGD) uses them as an interaction control signal intentionally directed towards a device, detected by the GOD plugin. In SGD, we define a saccade gesture as a fast eye movement from a defined origin, towards a direction, and back to the origin. As

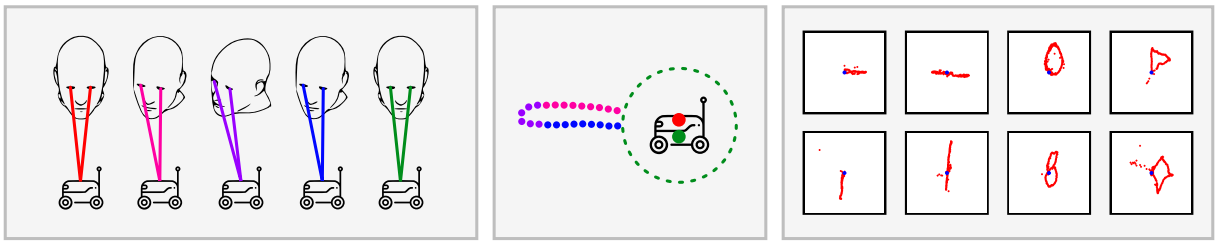


Figure 2: Visualization of the Head Gesture Detector (HGD). Left: the user performs a rightward head sway while maintaining fixation on the target device. Center: the resulting gaze trajectory, color-coded by head position. The red dot marks the initial Gaze at Object (GOD) event (fixation on the object), and the green dashed circle indicates the proximity threshold around the fixation. Once the gaze leaves this region, subsequent samples are evaluated for a smooth trajectory. When the gaze returns within the threshold and a fixation is detected (green dot), the gesture is considered complete and published to the event space. Right: Example recordings (from the HGD plugin) of how a head gesture can be used to *draw* different smooth gaze trajectories.

the GOD plugin publishes the FO_{pos} (fixation-on-object position), SGD monitors saccadic movements [Salvucci and Goldberg 2000]. If the gaze returns to FO_{pos} after a saccadic event and within a time window of 500 ms, the gesture is considered complete. Finally, the plugin publishes an event that contains the object name, gesture type, and direction of saccade.

3.3 Device Control Service

The *Device Control Service (DCS)* maps the high-level interaction cues—HGD, BGD, and SGD—to actionable commands that modify the state of IoT devices that are detected by GOD and connected via MQTT or HTTP protocols. Leveraging Thing Descriptions (TDs) to provide a standardized semantic interface [Strecker et al. 2022], the DCS retrieves device affordances and properties in JSON format to interpret gestures such as turning devices on/off or adjusting brightness.

Upon receiving a trigger event from one of the gesture detection plugins, the DCS queries the corresponding device's TD to retrieve both its current state and affordance details. Because the TD reflects the real-time status of the device, the DCS can dynamically map detected gestures to context-appropriate actions. For example, if a lamp is currently off, the system can define any head tilt gesture to turn it on. Subsequently, once the device state is on, a *head tilt up* gesture can be mapped to increase its brightness. Once the desired action is determined, the DCS communicates with the IoT device using the interfaces defined in the TD. The DCS operates in a topic-based communication framework (Figure 1-b), subscribing to gesture events and publishing action confirmation messages. This architecture ensures modularity, allowing the system to easily accommodate additional devices or interaction modalities without reconfiguring the entire pipeline. Additionally, the event-driven communication model ensures that the system remains responsive and scalable in multi-device environments. The event space consolidates real-time data from all components of the system, creating a synchronized interaction loop.

4 User Experiment

To identify intentional interaction patterns that allow fine grained control of various IoT devices, we conducted a controlled experiment. In this experiment, we asked users to interact with IoT devices in a proxy smart environment using gaze and head gestures. In this setting, our objective was to observe their natural behavior and, accordingly, finetune our framework. Hence, we tested the following hypotheses:

- H1: Interaction Time will be longer for Adjustment and Movement prompts compared to Toggle prompts.
- H2: All interactions will begin with a fixation on the target object.
- H3: For Movement prompts, head movements will align with the prompted direction.
- H4: Saccadic movements for Movement prompts will follow the prompted direction.
- H5: Intentional blinks for controlling devices will exceed 400 ms in duration.

Participants: We recruited 12 voluntary bachelor's and master's students aged between 18 and 28 years, but due to technical issues, we used data from nine (three female and six male) participants.

Environment and Setup: The setup included multiple IoT devices that are positioned at about 2 meters around the participant to ensure a clear view and accessible interaction via gaze and head gestures. Table 1 provides a detailed summary of the interaction types, specific actions performed on each device, and the number of trials conducted for each action.

Procedure: Each session began with a nine-point eye calibration. The main experiment consisted of prompts issued via the computer's speakers, instructing participants to perform specific interactions with the devices using only their gaze and head gestures. These prompts were categorized into three interaction types: *Toggle* (Binary actions such as turning a lamp on or off), *Adjustment* (Continuous actions such as dimming a lamp or adjusting microphone volume), and *Movement* (Directional actions such as moving a robotic arm or toy car). Upon hearing a prompt, they interacted with the specified device using gaze and head gestures. The eye tracker then recorded gaze data, head movements, and blink patterns during each interaction. A Python script ensured the

Table 1: Summary of control interaction types, devices, and actions. "Trials" indicates the number of times each action was prompted. Each trial used unique devices: three lamps, two robotic arms, one microphone, one conveyor belt, one toy car, and one mouse.

Type	Device	Action	Trials	Type	Device	Action	Trials
Toggle	Lamp	Turn On	3	Adjustment	Lamp	Brighten	3
		Turn Off	3			Dim	3
	Robot Gripper	Open	1		Microphone	Increase Volume	1
		Close	1			Decrease Volume	1
	Microphone	Mute	1	Movement	Robotic Arm	Move Up	2
		Unmute	1			Move Down	2
					Arm + Conveyor	Move Left	2
						Move Right	2
					Toy Car (A) + Mouse (B)	Move A→B (Left)	1
						Move B→A (Right)	1

Table 2: Annotations used to describe participant behavior observed during the eye-tracking experiment.

Annotation	Description
Object Glance (OG)	Participant only looked at the object for a period of time.
Object Blink (OB)	Participant blinked while focusing on the object.
Short-Saccades (SS)	Participant exhibited Short-Saccades within the object's bounding box.
Head Gesture (HG)	Participant moved their head in a direction (Up, Left, Right, Down) to interact with the object.
Directed Saccades (DS)	Saccadic movements toward a direction, starting from the object, moving outwards, and back.
Sequential Fixations (SF)	Participant fixated on the object and then on a direction toward the movement prompt.

experiment's standardization by automating prompt delivery and timing between interactions. Each interaction began with the examiner manually starting the recording in the Pupil Core software, and the recording ended once the participant completed the interaction. Random delays of 3–7 seconds were introduced between prompts to minimize anticipation effects. In this experiment, we measured *interaction time* (i.e. duration from gaze initiation on the target object to the return of gaze to the cognitive task); the count and duration of *blinks* before, during, and after interactions; *head movements* (i.e. smooth pursuit movements and angles relative to the target object), and *saccadic movements* (i.e. rapid gaze shifts during movements). The independent variable was the type of prompt (Toggle, Adjustment, or Movement) given to participants.

5 Results and Discussion

The data contained 252 samples recorded from 9 participants for 28 interaction prompts. In addition, three participants repeated seven trials with different gestures. Hence, the analysis included 259 sample recordings. We manually cropped each recording to the *interaction time* window, noting whether the participant blinked shortly before or after the window, as this information was excluded. Additionally, we annotated the observed interaction behavior (see Table 2).

Almost all participants began the interaction by focusing on the object, and only in 32 out of 259 recordings, no initial fixation was detected, likely due to a drop in gaze data confidence, as participants blinked approximately 55% of the time before the interaction time window began. In 73% of the samples, the interaction ended with a fixation on the object. Figure 3 (left) illustrates the frequency of each interaction behavior for the different control interaction types (see Table 1). In toggle tasks, participants predominantly exhibited *Object Glance (OG)* and *Object Blink (OB)* behaviors. In Adjustment tasks, *Short-Saccades* and *Head Gestures (HG)* were most common, while in Movement tasks, *Sequential Fixations (SF)* and *Head Gestures (HG)* were the most prominent. We did not observe *Object Glance (OG)* and *Object Blink (OB)* behaviors during the Movement tasks. For each action type, we analyze the normalized interaction duration across all participants (Figure 3). Toggle actions, except for *Toggle Open* and *Toggle Close* (relates to the gripper of the robotic-arm), had the shortest interaction durations. In contrast, *Movement Right/Left* and *Adjustment Increase/Decrease* actions generally took longer on average.

To analyze *Head Gesture* behaviors, we applied OpenCV's Lucas-Kanade optical flow algorithm to estimate the displacement of key features in each frame of the video recorded during the interaction [Lucas and Kanade 1981]. In all trials that involved movement actions and where a *Head Gesture* was expected (49 out of 259), we compared the estimated camera movement with the prompted movement direction and found an alignment between them in 43 out of the 49 trials (87.76%). Head movements while fixating on the object were observed in 4 out of the 9 participants. In 27 of 259 trials with *Directed Saccades* participants were expected to exhibit saccadic movements away from the object towards a specified direction. We found that *Directed Saccades* aligned with the prompted direction in 26 of the 27 (96.30%) trails. The *Object Blink* behavior occurred in 35 of the 259 trials. The mean blink duration was approximately 259 ms, with a standard deviation of 151 ms. Additionally, we observed that blinks were often combined with other behaviors. Below, we interpret these preliminary findings that motivate us to conduct more detailed studies in the future.

Hypothesis H1: *Interaction Time will be longer for Adjustment and Movement prompts compared to Toggle prompts.* Our findings support this hypothesis. As shown in Figure 3, toggle actions had

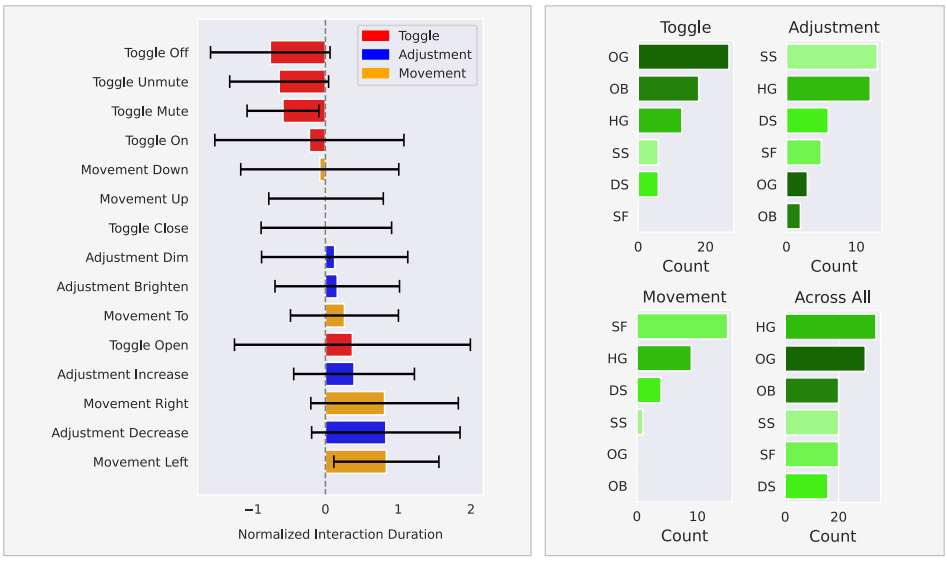


Figure 3: Overview of the experiment results. Left: Duration of interaction time windows for different action types, normalized (z-score) per participant. Colors indicate the high-level interaction type (Toggle, Adjustment, and Movement Interactions). Right: The count of annotations (observed interactions) for each control interaction type: Object Glance (OG), Object Blink (OB), Short-Saccades (SS), Head Gesture (HG), Directed Saccades (DS) and Sequential Fixations (SF).

the shortest duration, whereas *Adjustment* and *Movement* actions took significantly longer. Among toggle actions, *Toggle Open/Close* for the robotic arm's gripper exhibited slightly longer durations, which can be interpreted as wrong assumption of the intention-affordance mapping. Opening the gripper can arguably be seen as an adjustment due to the need for precise alignment before triggering the action. *Move Left/Right* and *Adjustment Increase/Decrease* actions had the longest durations, suggesting that fine-grained control actions require longer interactions.

Hypothesis H2: *All interactions will begin with a fixation on the target object.* In 87.65% of the trials, the interaction began with a fixation on the target device, highlighting its importance in gaze-based device control. However, in 32 trials, no initial fixation was detected. This was likely due to low tracking confidence often caused by the blinks just before the interaction (55% of cases).

Hypothesis H3: *For Movement prompts, head movements will align with the prompted direction.* In trials involving movement actions, we observed a strong alignment between the prompted direction and the head movements. This suggests that head movements were naturally aligned with user's intention.

Hypothesis H4: *Saccadic movements for Movement prompts will follow the prompted direction.* In 96.30% of the trials, where *Directed Saccades (DS)* were expected, the estimated saccadic direction aligned with the direction of the prompted movement. This suggests that users naturally employ saccadic movements as part of their interaction strategy, making them a reliable control input.

Hypothesis H5: *Intentional blinks for controlling devices will exceed 400 ms in duration.* Across 35 trials where we expected the *Object Blink (OB)* behavior, the mean blink duration (259 ms, SD = 151 ms) remained within the natural blink duration (shorter than the 400 ms threshold) as reported in previous studies. Unless a blink is part of a deliberately staged interaction sequence (that is, when the users are trained or informed), for intentional blink-based gestures, we recommend setting the threshold longer than 500 ms as we implemented in the BGD (Section 3.2.2).

5.1 Limitations and Challenges

Despite promising findings, we must acknowledge several limitations of our work. First, while the system successfully detected gaze-based interactions, there were cases of data loss due to blinks or low tracking confidence. This highlights the need for more robust tracking, particularly in uncontrolled environments. Another limitation concerns the variability in user behavior observed across participants. Some individuals relied more on head gestures, while others favored saccadic movements. This variability suggests that a personalized interaction model (cf. [Strecker et al. 2025]) — one that adapts to individual user preferences — could enhance overall usability. Furthermore, our experiment was conducted with only nine participants and a limited number of recordings, which does not represent broader user behavior. Finally, we conducted the experiment in a controlled setting with predefined prompts. Although this ensured consistency, it may not fully reflect spontaneous real-world interactions. Future research should evaluate gaze-based interactions in realistic environments, such as smart homes or industrial settings, to better capture the complexity of natural user behavior.

We propose several recommendations for future work in gaze-based interaction design: Since gaze behavior varies between users, integrating machine learning techniques (cf. [Bednarik et al. 2012; Bektaş et al. 2024]) to dynamically adapt control sensitivity could improve precision through the development of adaptive gaze models. Furthermore, future studies should document the merits of *Magic Gaze* by systematically comparing it to more conventional

hands-free interaction modalities (e.g., voice or touch). Future systems could also incorporate contextual information (cf. [Dogan et al. 2024]), such as user intent prediction, environmental conditions, or device states, to enhance interaction reliability through context-aware gaze control. Finally, given the variability in blink behavior, refining blink-based interactions through dwell-based validation mechanisms could enhance selection accuracy.

Challenges remain in designing robust and user-friendly gaze-based systems. Strecker et al. [Strecker et al. 2023] underscored the importance of combining heterogeneous data sources to improve the reliability of gaze-based interactions in mixed reality (MR) environments. In addition, ethical and privacy considerations must be addressed in the design of gaze-based interfaces [Kröger et al. 2020]. For example, recent research proposed privacy-preserving mechanisms in human-agent collaboration [Grau et al. 2024] and human activity recognition [Bektaş et al. 2024], making sure gaze data are used responsibly.

6 Conclusion

We explored the feasibility *Magic Gaze*: integrating eye tracking with object detection and gesture recognition to enable seamless hands-free control of IoT devices. Through a structured experiment, we systematically assessed how individuals intuitively employ gaze- and head-based gestures to control devices. Our findings indicate that gaze-based control is intuitive and viable, particularly for binary (toggle) actions and directional (movement) commands. We observed that users naturally rely on fixations, head gestures, and directed saccades as part of their interaction strategy. While our results are promising several challenges remain. Variability in user behavior suggests that a *personalized interaction model* could enhance usability by adapting to individual preferences. Additionally, *gaze tracking accuracy* must be improved, especially in real-world conditions where lighting variations and involuntary blinks impact detection reliability. Looking ahead, we propose *two key directions* for advancing gaze-based interaction systems: The first involves incorporating machine learning techniques to dynamically adjust control sensitivity based on individual user behavior, enabling more adaptive gaze models that respond to varying interaction patterns. The second direction focuses on leveraging environmental context and user intent prediction to improve interaction accuracy, allowing future systems to become more context-aware and responsive to the conditions in which they are used. This work contributes to the growing field of gaze-based human-computer interaction, demonstrating its potential for hands-free control in smart environments.

Acknowledgments

We thank Besmir Kadrii for his support during the data collection phase of the experiments and for his assistance with annotating the experimental data. We also acknowledge his contributions to the development of the *Blink Gesture Detection* (BGD) plugin.

References

Roman Bednarik, Hana Vrzakova, and Michal Hradis. 2012. What do you want to do next: a novel approach for intent prediction in gaze-based interaction. In *Proceedings of the Symposium on Eye Tracking Research and Applications* (Santa Barbara, California) *(ETRA '12)*. Association for Computing Machinery, New York, NY, USA, 83–90. doi:10.1145/2168556.2168569

Kenan Bektaş, Adrian Pandjaitan, Jannis Strecker, and Simon Mayer. 2024. GlassBoARd: A Gaze-Enabled AR Interface for Collaborative Work. In *Extended Abstracts of the CHI Conference on Human Factors in Computing Systems* (Honolulu, HI, USA) *(CHI EA '24)*. Association for Computing Machinery, New York, NY, USA, Article 181, 8 pages. doi:10.1145/3613905.3650965

Kenan Bektaş, Jannis Strecker, Simon Mayer, Dr. Kimberly Garcia, Jonas Hermann, Kay Erik Jenß, Yasmine Sheila Antille, and Marc Solèr. 2023. GEAR: Gaze-enabled augmented reality for human activity recognition. In *Proceedings of the 2023 Symposium on Eye Tracking Research and Applications* (Tubingen, Germany) *(ETRA '23)*. Association for Computing Machinery, New York, NY, USA, Article 9, 9 pages. doi:10.1145/3588015.3588402

Kenan Bektaş. 2020. Toward A Pervasive Gaze-Contingent Assistance System: Attention and Context-Awareness in Augmented Reality. In *ACM Symposium on Eye Tracking Research and Applications*. ACM, Stuttgart Germany, 1–3. doi:10.1145/3379157.3391657

Kenan Bektaş, Jannis Strecker, Simon Mayer, and Kimberly Garcia. 2024. Gaze-enabled activity recognition for augmented reality feedback. *Computers & Graphics* 119 (April 2024), 103909. doi:10.1016/j.cag.2024.103909

Anna Belardinelli, Oliver Herbort, and Martin V Butz. 2015. Goal-oriented gaze strategies afforded by object interaction. *Vision Research* 106 (2015), 47–57. doi:10.1016/j.visres.2014.11.003

Richard A. Bolt. 1980. "Put-that-there": Voice and gesture at the graphics interface. In *Proceedings of the 7th Annual Conference on Computer Graphics and Interactive Techniques* (Seattle, Washington, USA) *(SIGGRAPH '80)*. Association for Computing Machinery, New York, NY, USA, 262–270. doi:10.1145/800250.807503

Richard A. Bolt. 1981. Gaze-orchestrated dynamic windows. In *Proceedings of the 8th Annual Conference on Computer Graphics and Interactive Techniques* (Dallas, Texas, USA) *(SIGGRAPH '81)*. Association for Computing Machinery, New York, NY, USA, 109–119. doi:10.1145/800224.806796

Anne-Marie Brouwer, Volker H Franz, and Karl R Gegenfurtner. 2009. Differences in fixations between grasping and viewing objects. *Journal of Vision* 9, 1 (2009), 18–18. doi:10.1167/9.1.18

Andreas Bulling and Hans Gellersen. 2010. Toward Mobile Eye-Based Human-Computer Interaction. *IEEE Pervasive Computing* 9, 4 (Oct. 2010), 8–12. doi:10.1109/MPRV.2010.86

Mustafa Doga Dogan, Eric J Gonzalez, Karan Ahuja, Ruofei Du, Andrea Colaço, Johnny Lee, Mar Gonzalez-Franco, and David Kim. 2024. Augmented Object Intelligence with XR-Objects. In *Proceedings of the 37th Annual ACM Symposium on User Interface Software and Technology* (Pittsburgh, PA, USA) *(UIST '24)*. Association for Computing Machinery, New York, NY, USA, Article 19, 15 pages. doi:10.1145/3654777.3676379

Michael Fetter. 2007. Vestibulo-Ocular Reflex. In *Neuro-Ophthalmology: Neuronal Control of Eye Movements*. S.Karger AG. arXiv:https://karger.com/book/chapter-pdf/2094794/000100348.pdf doi:10.1159/000100348

Stefan Fuchs and Anna Belardinelli. 2021. Gaze-based intention estimation for shared autonomy in pick-and-place tasks. *Frontiers in Neurorobotics* 15 (2021), 17. doi:10.3389/fnbot.2021.647930

Jan Grau, Simon Mayer, Jannis Strecker, Kimberly Garcia, and Kenan Bektas. 2024. Gaze-based Opportunistic Privacy-preserving Human-Agent Collaboration. In *Extended Abstracts of the CHI Conference on Human Factors in Computing Systems* (Honolulu, HI, USA) *(CHI EA '24)*. Association for Computing Machinery, New York, NY, USA, Article 176, 6 pages. doi:10.1145/3613905.3651066

Sobhan Heravian, Nima Nouri, Mojtaba Behnam Taghadosi, and Seyed Mohammad Hossein Seyedkashi. 2019. *Implementation of Eye Tracking in an IoT-Based Smart Home for Spinal Cord Injury Patients*. Research Article 16. Modern Care Journal. doi:10.5812/modernc.96107

Damian Hostettler, Kenan Bektaş, and Simon Mayer. 2023. Pupillometry for Measuring User Response to Movement of an Industrial Robot. In *Proceedings of the 2023 Symposium on Eye Tracking Research and Applications* (Tubingen, Germany) *(ETRA '23)*. Association for Computing Machinery, New York, NY, USA, Article 52, 2 pages. doi:10.1145/3588015.3590123

Howell Istance, Aulikki Hyrskykari, Lauri Immonen, Santtu Mansikkamaa, and Stephen Vickers. 2010. Designing gaze gestures for gaming: an investigation of performance. In *Proceedings of the 2010 Symposium on Eye-Tracking Research & Applications - ETRA '10*. ACM Press, Austin, Texas, 323. doi:10.1145/1743666.1743740

Robert J. K. Jacob. 1995. Eye tracking in advanced interface design. In *Virtual Environments and Advanced Interface Design*, W. Barfield and T. A. Furness (Eds.). Oxford University Press, New York, 258–288.

Glenn Jocher and Jing Qiu. 2024. *Ultralytics YOLO11*. https://github.com/ultralytics/ultralytics

Moritz Kassner, William Patera, and Andreas Bulling. 2014. Pupil: An Open Source Platform for Pervasive Eye Tracking and Mobile Gaze-based Interaction. In *Adjunct Proceedings of the 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing* (Seattle, Washington) *(UbiComp '14 Adjunct)*. ACM, New York, NY, USA, 1151–1160. doi:10.1145/2638728.2641695

Aleksandra Królak and Paweł Strumiłło. 2012. Eye-blink detection system for human–computer interaction. *Universal Access in the Information Society* 11, 4 (2012), 409–419. doi:10.1007/s10209-011-0256-6

Jacob Leon Kröger, Otto Hans-Martin Lutz, and Florian Müller. 2020. What Does Your Gaze Reveal About You? On the Privacy Implications of Eye Tracking. In *Privacy and Identity Management. Data for Better Living: AI and Privacy: 14th IFIP WG 9.2, 9.6/11.7, 11.6/SIG 9.2.2 International Summer School, Windisch, Switzerland, August 19–23, 2019, Revised Selected Papers*, Michael Friedewald, Melek Önen, Eva Lievens, Stephan Krenn, and Samuel Fricker (Eds.). Springer International Publishing, Cham, 226–241. doi:10.1007/978-3-030-42504-3_15

Michael F. Land and Mary Hayhoe. 2001. In what ways do eye movements contribute to everyday activities? *Vision Research* 41, 25 (Nov. 2001), 3559–3565. doi:10.1016/S0042-6989(01)00102-X

Bruce Lucas and Takeo Kanade. 1981. An Iterative Image Registration Technique with an Application to Stereo Vision (IJCAI). *[No source information available]* 81.

Simon Mayer, Andreas Tschofen, Anind K. Dey, and Friedemann Mattern. 2014. User interfaces for smart things – A generative approach with semantic interaction descriptions. *ACM Transactions on Computer-Human Interaction* 21, 2 (Feb. 2014), 1–25. doi:10.1145/2584670

Dario D. Salvucci and Joseph H. Goldberg. 2000. Identifying fixations and saccades in eye-tracking protocols. In *Proceedings of the symposium on Eye tracking research & applications - ETRA '00*. ACM Press, Palm Beach Gardens, Florida, United States, 71–78. doi:10.1145/355017.355028

Sima Soltani and Amin Mahnam. 2016. A practical efficient human computer interface based on saccadic eye movements for people with disabilities. *Computers in Biology and Medicine* 70 (2016), 163–173. doi:10.1016/j.compbiomed.2016.01.012

John A. Stern, Larry C. Walrath, and Robert Goldstein. 1984. The Endogenous Eyeblink. *Psychophysiology* 21, 1 (1984), 22–33. arXiv:https://onlinelibrary.wiley.com/doi/pdf/10.1111/j.1469-8986.1984.tb02312.x doi:10.1111/j.1469-8986.1984.tb02312.x

Jannis Strecker, Khakim Akhunov, Federico Carbone, Kimberly García, Kenan Bektaş, Andres Gomez, Simon Mayer, and Kasim Sinan Yildirim. 2023. MR Object Identification and Interaction: Fusing Object Situation Information from Heterogeneous Sources. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* 7, 3, Article 124 (Sept. 2023), 26 pages. doi:10.1145/3610879

Jannis Strecker, Kimberly García, Kenan Bektaş, Simon Mayer, and Ganesh Ramanathan. 2022. SOCRAR: Semantic OCR through Augmented Reality. In *Proceedings of the 12th International Conference on the Internet of Things*. ACM, Delft Netherlands, 25–32. doi:10.1145/3567445.3567453

Jannis Strecker, Simon Mayer, and Kenan Bektaş. 2025. Towards Societally Beneficial Personalized Realities: A Conceptual Foundation for Responsible Ubiquitous Personalization Systems. In *Proceedings of the 2025 ACM Designing Interactive Systems Conference*. ACM, Madeira Portugal, 1792–1814. doi:10.1145/3715336.3735709

Eduardo Velloso, Markus Wirth, Christian Weichel, Augusto Esteves, and Hans Gellersen. 2016. AmbiGaze: Direct Control of Ambient Devices by Gaze. In *Proceedings of the 2016 ACM Conference on Designing Interactive Systems*. ACM, Brisbane QLD Australia, 812–817. doi:10.1145/2901790.2901867

W3C. 2023. Web of Things (WoT) Thing Description 1.1. https://www.w3.org/TR/wot-thing-description11/ Last accessed 15 March 2026.

Mark Weiser. 1999. The computer for the 21st century. *ACM SIGMOBILE Mobile Computing and Communications Review* 3, 3 (July 1999), 3–11. doi:10.1145/329124.329126

Oleg Špakov and Päivi Majaranta. 2012. Enhanced gaze interaction using simple head gestures. In *Proceedings of the 2012 ACM Conference on Ubiquitous Computing*. ACM, Pittsburgh Pennsylvania, 705–710. doi:10.1145/2370216.2370369