UAV Control Using Eye Gestures: Exploring the Skies Through Your Eyes

Brandon Dominic Vilela*†
vilela_brandon_domin@student.smc.edu
Department of Engineering
Santa Monica College
Santa Monica, California, USA

Kshitij Kokkera†
kokkerak@vcu.edu
Department of Computer Science
Virginia Commonwealth University
Richmond, Virginia, USA

Amirahmad Chapnevis
chapnevisa@vcu.edu
Department of Computer Science
Virginia Commonwealth University
Richmond, Virginia, USA

Eyuphan Bulut
ebulut@vcu.edu
Department of Computer Science
Virginia Commonwealth University
Richmond, Virginia, USA

ABSTRACT

Unmanned aerial vehicle (UAV) technology has become increasingly pivotal in various industries including agriculture, emergencies, and transportation. However, there is a growing need for more intuitive and unobtrusive control mechanisms. In response, our team has developed a groundbreaking technique that optimizes drone control and tracking through operator gaze. Through the use of eye-tracking interaction, we have created a more intuitive approach to human operation, which reduces operator workload and improves overall efficiency. After extensive testing on the Parrot ANAFI drone, we have concluded that this implementation has the potential to revolutionize drone control and elevate it to new heights.

CCS CONCEPTS

• **Networks → Cyber-physical networks**; • **Hardware →** *Analysis and design of emerging devices and systems*; • **Human-centered computing → Interaction devices**; **Mobile devices**.

KEYWORDS

Unmanned Aerial Vehicles, eye tracking, gaze control, drones.

ACM Reference Format:
Brandon Dominic Vilela, Kshitij Kokkera, Amirahmad Chapnevis, and Eyuphan Bulut. 2023. UAV Control Using Eye Gestures: Exploring the Skies Through Your Eyes. In *The Twenty-fourth International Symposium on Theory, Algorithmic Foundations, and Protocol Design for Mobile Networks and Mobile Computing (MobiHoc '23), October 23–26, 2023, Washington, DC, USA.* ACM, New York, NY, USA, 6 pages. https://doi.org/10.1145/3565287.3617619

1 INTRODUCTION

Unmanned Aerial Vehicles (UAVs) or drones have become increasingly popular for various applications including surveillance, photography, agriculture, delivery services, communication and more [2, 4, 14]. The research and development in this area have also been motivated by the desire to enable users, including individuals with physical disabilities or limited mobility to operate drones effortlessly. For example, [9] expressly designs a multi-modal human machine interface intended for individuals for disabilities, and while their interface is integrated into a smart wheelchair, it is evident that the integration of gaze tracking plays a critical role in creating accessible technology.

In exploration missions, drones play a crucial role in reaching remote, rural and hard-to-access locations [1, 3, 6]. They provide valuable aerial perspectives that aid researchers, geologists, and environmentalists in studying terrains, wildlife habitats, and natural disasters. For example, during search and rescue missions, every second counts. UAVs equipped with cameras and thermal imaging can be deployed rapidly to locate missing individuals, assess disaster-stricken areas, or search for survivors in hazardous conditions. Eye gesture-controlled drones can offer a significant advantage in these high-pressure situations, where operators may need to handle other equipment, communicate with ground teams, or attend to critical tasks while simultaneously piloting the drone. To give an instance of this concept we can look at [11] where the authors merit the idea of gaze tracking in UAV video feeds, as it can potentially avoid false-negatives in long term observational periods. Moreover, as shown in [5], usage of gaze for exploration can significantly reduce cognitive workload, improve natural human-robot interaction, and bolster spacial comprehension.

The rest of the paper is organized as follows. Section 2 discusses other works in gaze control for drone operation and some of their shortcomings. Section 3 presents an overview of our system and goes over the algorithms and methodologies we use. We collect user feedback in Section 4 by having participants perform a series of tasks to analyze our system's effectiveness. In Section 5, we address

---

[*] The student was REU student at Virginia Commonwealth University when this work is performed.
[†] The authors contributed equally.

MobiHoc '23, October 23–26, 2023, Washington, DC, USA

ACM ISBN 978-1-4503-9926-5/23/10…$15.00
https://doi.org/10.1145/3565287.3617619

the user feedback from experiments, and discuss the challenges we faced while implementing this system, and our future work. Lastly, Section 6 details our closing remarks.

2 RELATED WORK

In this section, we provide an overview of studies that have used eye trackers and eye gaze to control and interact with robots and aerial vehicles including drones. In [18], the authors introduce a method to enable human-MAV teams to use human eye gaze. It decouples head orientation from gaze, identifies the aerial vehicle, and allows controlling the MAV by gazing at specific points. The approach could improve control, multi-agent collaboration, and anticipative interfaces. However, only a limited user testing is performed in that study. In addition, the proposed idea is only a partial solution for human-robot interaction.

The work in [7] is similar to ours, with participants asked to complete a navigational task with various control schemes. However, the control schemes being evaluated include both gaze-based control and keyboard-based control, whereas in this study we explore only a standalone control with gaze. Furthermore, the evaluation of the system in that study is performed only for an indoor environment, which is different from average flying conditions; factors such as air turbulence, temperature, and signal interference must be accounted for when comparing control mechanisms.

Similarly the control schemes presented in [17] are a glimpse of possible future work. The authors of this study ask participants to experience three different modes of control: manual, hybrid, and hands-free. The difference however, is their system is based on a ground based robot in a simulation environment whereas we use a drone in real environments.

The authors of [13] develop a method of drone control with gaze positions. They capture this gaze position with a mobile eye tracker, specifically using the Pupil Labs Core eye tracking headset. The issue with mobile headsets is that they can limit the natural movement of users as they are fastened to a user's body. Remote headsets, such as the Tobii Pro Nano, provide a more intuitive experience for users as users do not have to attach any electronics to themselves. Additionally, the means to navigate the drone are constrained to a line of sight because gaze positions are used to find fiducial markers, meaning the system can only be operated by line of sight. Another prohibitive feature is using fiducials as an indicator for an area of interest because the drone is limited to the markers' discrete locations.

The work in [10] studies the use of a virtual reality headset for individuals with mobility disability to let them gain a live visual feed of any environment by converting head movements into commands sent to the drone. The use of the headset however is impeding, as users have to purchase, charge, and wear this device in order to manipulate the drone. The convenience of remote gaze tracking devices far outweighs the immersive experience of a virtual reality headset.

Another interesting study is also presented in [8]. Different from the aforementioned works which focus on gaze based interaction with mobile robots or drones, this study focuses on gaze interaction through the assistance of the drone. The goal is to avoid the limitations due to the headsets or remote trackers and obtain a more natural eye tracking from the drone's camera which can sense the user's input and send it to other pervasive systems. The benefits of having a drone-assisted eye tracker would be auto-adjustment depending on user height and control sensitivity preferences, flexibility in environmental conditions where fixed eye trackers may not be stably placed, and user following where if a user decided to move to another location, the drone could theoretically follow them. Note that such a system is orthogonal to our system. That is a drone-assisted eye tracker can be integrated to our solution to control another drone in the environment.

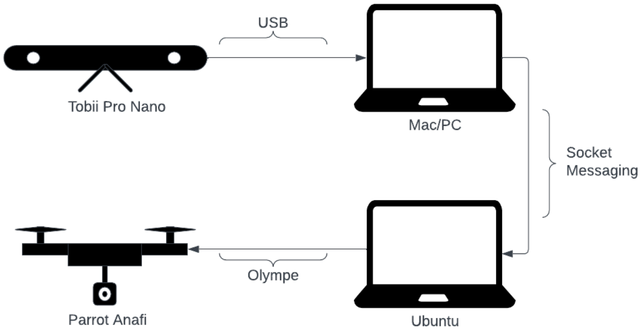

Figure 1: Overview of the proposed system. A machine running MacOS or Windows communicates flight commands to a machine running Ubuntu, which operates the Anafi drone using the Parrot Olympe SDK.

3 OVERVIEW OF THE PROPOSED SYSTEM

Fig. 1 illustrates an overview of the proposed system with all components. Next, we elaborate on these components and discuss how the control of the drone is achieved.

3.1 Eye Tracker

3.1.1 Hardware Description. We use the Tobii Pro Nano for our eye tracker-based design to obtain the gaze information or where the person is looking. The Tobii Pro Nano has a sampling frequency of 60Hz [15] and outputs data consisting of a timestamp, gaze origin, gaze point, and pupil diameter. It can be operated from 45-85 centimeters from the eye tracker. It can track the user's eyes in an approximately 40-centimeter by 40-centimeter box, assuming the user is 70 centimeters from the eye tracker.

The primary coordinate system our system uses is the Active Display Coordinate System (ADCS) [16], which is a 2-dimensional coordinate system aligns with a display like a monitor or a laptop screen (see Fig. 2). The top left of the computer's screen is set to (0,0), and the bottom right is set to (1,1).

The tracker provides the gaze vector, which is a vector that begins in the eye and ends when it intersects an opaque object. By using the intersection of the gaze vector and the active display area plane, we can obtain an ordered pair that we call the gaze point. With this gaze point we can map the position onto a set of actions in order to control the drone.

We assume that the user, who is the individual operating this system, wants to have their area of interest in the center of their computer screen, given a live video stream. We want the drone

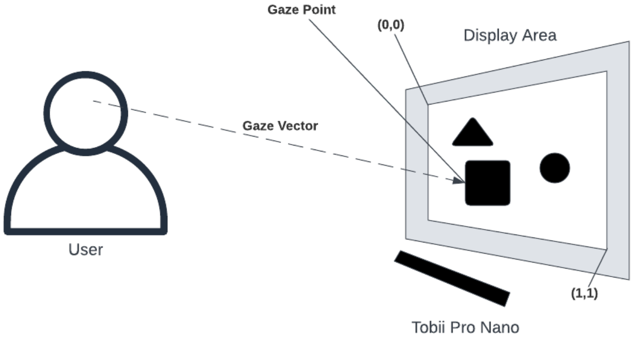

Figure 2: Depiction of the coordinate system for the Tobii Pro Nano.

actions to revolve around aligning itself towards that given area of interest.

By rounding the gaze point to nine possible points, it allows us to create eight different actions, with the center point of the screen not generating any action. These actions are implemented as strings; so for every frame we concatenate the actions the user wants to perform. For example "Fly Up", "Turn Left", and "Move Forward" extracted from three consecutive frames are combined as "Fly Up Turn Left Move Forward".

Because Olympe, the Parrot Anafi's SDK, only operates on Linux, we have to transmit our intended command to a Linux machine. To this end, we used a separate laptop at first but then switched to a virtual machine. Both options used Ubuntu Linux 22.04.

3.1.2 Gesture specifications. Gesture control is initiated based on if a user's eye or eyes are shut. We can detect this by appending the eye validity data from the Tobii Pro Nano, which is a binary value representing if the eye tracker can detect the user's eye or not, into a fixed size array. If every element in the array is a zero, we know that the user has either closed that eye or has walked away from the eye tracker. For our takeoff we close our right eye for approximately two seconds and to land we can either walk away from the eye tracker or shut both our eyes for two seconds.

For longitudinal (forward-backward) and lateral (left-right) movements, we use tilt. The implementation we use is based on the Tobii Pro Nano's Track Box Coordinate System (TBCS). TBCS is the volume which the eye tracker can theoretically still follow the user's eyes. For forward and backward control we check to see if the user's eyes are located outside of some range on the z-axis. For example, if the position along the z-axis is less than 0.3 then we can send the command to move forward, and if the position is greater than 0.6, a move backward can be triggered. Controlling left and right is similar, we check if the user's eye position along the x-axis is greater than or less than some value.

To control vertical movement as well as rotation, we collect the user's gaze point in the current frame, clamp the coordinate values between 0 and 1, and then round the coordinate values by 1/2, which then gives us only nine possible states, as shown in Fig. 3 and Table 1.

Table 1: Mapping of rounded gaze point location to flight command.

Rounded Gaze Point	Flight Action
(0,0)	Fly Up & Turn Left.
(1,0)	Fly Up & Turn Right.
(0,1)	Fly Down & Turn Left.
(1,1)	Fly Down & Turn Right.
(0.5,0.5)	Null Action.
(0.5,0)	Fly Up.
(0.5,1)	Fly Down.
(0,0.5)	Turn Left.
(1,0.5)	Turn Right.

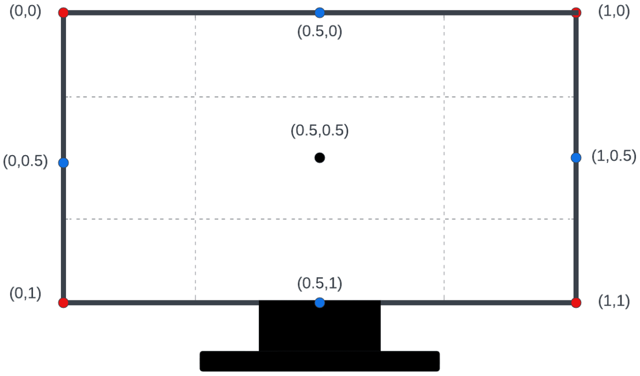

Figure 3: Diagram of display and the possible points which the gaze point can be rounded to. Blue dots mark single action commands while red dots mark multi action commands.

3.2 Drone Control

Traditionally, the Parrot Drone ANAFI, can be controlled using either a physical controller, or through their in house SDK called Olympe [12]. Olympe provides a Python controller programming interface for the Parrot ANAFI drone. Although not limited to, it can be used to connect to and control a drone from a remote Python script running on a Linux based computer.

The Parrot Drone SDK Olypme is a set of libraries and frameworks allowing developers to operate Parrot drones remotely. With the Parrot Drone SDK, one can control the drone from a ground station controller application.

In addition with the Olympe SDK, we use the Parrot Sphinx which is the simulation software for Parrot drones. This unique simulation software allows us to create virtual drones and control them with our Olympe-based application and algorithms. The main objective while using the Sphynx simulator is to execute the drone firmware, with all its sensors and actuators, in a visually and physically realistic environment. The Parrot Sphinx not only enables us to visualize and record flight data in real-time but also simulates all sensors, including cameras. The Parrot Sphinx has played a crucial

Algorithm 1: Converting given flight command captured by eye tracker to Olympe SDK format. C_ψ, C_z, and C_m representing rotational, vertical, and planar movement constants, respectively.

```
Input: String $\mathcal{S}$: Given Flight Command
if "Turn Left" in $\mathcal{S}$ then
    $\Psi \leftarrow -(C_\psi)$
end
else if "Turn Right" in $\mathcal{S}$ then
    $\Psi \leftarrow C_\psi$
end
if "Fly Up" in $\mathcal{S}$ then
    $\mathcal{Z} \leftarrow -(C_z)$
end
else if "Fly Down" in $\mathcal{S}$ then
    $\mathcal{Z} \leftarrow C_z$
end
if "Move Left" in $\mathcal{S}$ then
    $\mathcal{Y} \leftarrow -(C_m)$
end
else if "Move Right" in $\mathcal{S}$ then
    $\mathcal{Y} \leftarrow C_m$
end
if "Move Forward" in $\mathcal{S}$ then
    $\mathcal{X} \leftarrow C_m$
end
else if "Move Backward" in $\mathcal{S}$ then
    $\mathcal{X} \leftarrow -(C_m)$
end
moveBy($\mathcal{X}, \mathcal{Y}, \mathcal{Z}, \Psi$)
```

role in assisting us with prototyping our algorithms before testing them in a controlled real-time experiment. Fortunately, we are able to imitate the exact experiment on the Sphynx simulator and replicate it in real-time without any problem.

The main command which we use to control the drone is the *moveBy* function. The parameters for this function are ($d_\mathcal{X}$, $d_\mathcal{Y}$, $d_\mathcal{Z}$, d_Ψ) which are the relative displacement along the longitudinal axis, lateral axis, vertical axis, and yaw axis, respectively.

This implementation is naive and in future work we would like to be able to pass values themselves rather than translating flight commands into movements defined by fixed constants.

Live video streaming is made using PDrAW. PDrAW is Parrot's video player library and desktop application which primarily operates with an Real-time Transport Protocol (RTP) stream but can play recorded MP4 videos.

By connecting to the ANAFI drone's WiFi and running the PDrAW desktop application, we are able to get a video stream. We also use the HUD option for monitoring battery usage and network connection status.

We have also implemented our own video streaming using OpenCV however we are not able to send video frames across machines efficiently, so we decided to stay with PDrAW.

Figure 4: Video streaming from Parrot ANAFI drone with HUD. User can control the drone by eye movements on this view.

4 EXPERIMENTS

4.1 Usability based Evaluation

With the assistance of other REU Computer Science research interns, we are able to formulate a conclusive analysis in which we compare multiple ways to operate the ANAFI drone. In a controlled environment, we ask the test subjects to takeoff and move the UAV 20 yards directly in front of them, then rotate the drone one hundred and eighty degrees to the right, then return to the point of origin. Finally, we ask the users to land the drone.

We first allow them to operate the drone using the Parrot controller. Next, we ask the subjects to operate the drone via *laptop control*, which is pre-programmed to operate the drone using the laptop's keyboard control. For this specific experiment, the arrow keys are used to either rotate the drone or change the height of the trajectory. In addition, the keyboard keys "A, W, S, D" are used to move the drone to the left, forward, backward, and to the right, respectively. The laptop control has real-time video camera feed assisting the subject to visually see where they are going during this process (see Fig.4). Finally, we ask the subjects to operate the drone using the Tobii Pro Nano, using eye gestures to complete this specific task.

In addition to operating the drone via eye gestures, the UAV drone operators are asked to complete specific tasks called phases, that they need to complete while operating the drone simultaneously (without any prior training or experience). These testing phases are designed to represent real-time movement or scenarios in which the UAV operator is operating the UAV and needing to assist or use other limbs in conjunction with their eyesight. We specifically aim for UAV operators who are in a more uncontrolled environment, cases including search and rescue missions, where time is crucial and of the essence. We believe in these high pressure situations, the drone operator is in an advantage and can use other equipment, communicate with other individuals from their team, or perform critical tasks while simultaneously piloting the UAV via eye gestures.

4.2 UAV Operators

We enlist the help of three UAV operators to assist us with this unique experiment, each with different backgrounds and drone operation skills. The first drone operator, is of large frame build and has no prior experience operating UAVs. In addition, drone operator one plays video games in their spare time as a hobby. Drone operator two is a smaller build and also has no experience with drone control. However, this UAV operator is athletic and frequently goes rock climbing as a hobby. Our last drone operator has the same build as the second drone operator, but has actual drone experience in the past. This drone operator frequently uses the gym. We believe having different types of drone operators will yield a more realistic result.

4.3 Task Descriptions

We devise the testing period into five phases each of which needs to be completed while operating the drone using eye gestures with the Tobii Pro Nano. After completing all five phases, we ask the three drone operators to rate their overall experience with drone operation via eye gesture control, from a scale from one through five, one being the most difficult while operating the drone, and five being the most simple or effortless experience while operating the drone.

For phase one, the operator is asked to operate the drone via takeoff, then move directly 20 feet in front of point of origin. Next, the drone will rotate a full one hundred and eighty degrees, and return to the original position. Then finally, land the drone.

For the second phase of the experiment, we ask the operator to do the same task as the first phase, however this time they are asked to pick up a water bottle, open the cap, dump out approximately a quarter out of the water bottle and finally return the cap on the top of the water bottle, and set it down besides them. In this phase we want to simulate a real-time action in which a drone operator can physically drink or use an object of the same dimensions, while simultaneously operating the drone.

For phase three of this experiment, the drone operator is asked to move the drone approximately five feet in front of them, rotate the drone one hundred eighty degrees to the right, and finally return to the point of origin. In this phase, about half way during the project time, we ask the operator to open the cap of an Expo marker, then rotate the marker three hundred and sixty degrees on its y and x axis. All drone operators were able to successfully complete this task simultaneously while operating the drone via eye gesture using the Tobii Pro Nano. In this phase, we wanted to closely simulate a real-time action in which a drone operator can physically operate another mini controller or device, while operating the drone.

In phase four of the experiment, we asked the operator to repeat the procedure, then open a case of an in-ear headphones, remove them from their charging pod, insert the earphone to the left and right ears, remove them from their ears and place them back inside of the pod, then finally land the drone. This phase was designed to replicate a real time action in which a drone operator could use other devices that require your hearing, while operating the drone.

Our last phase of the experiment includes the same beginning procedure, only this time the drone operator was asked to write and then sign their first and last name, while operating the drone with their eyes. This phase was specifically designed to simulate a drone operator needing to sign or write freely during mid flight.

Table 2: Experiment Phase 1-5 Results (Scale 1: most difficult to 5: most simple)

	Phase				
UAV Operator	**1**	**2**	**3**	**4**	**5**
Operator 1	4.5	2.5	4	3.8	3.5
Operator 2	2	1.5	2.5	2	4
Operator 3	2.5	3.5	3.5	3	3

4.4 Results

First of all, all the three drone operators were successfully able to operate the ANAFI drone via eye gesture as well as their specific objectives during their run-time. All three drone operators were not only able to move the drone in front of them, but also complete the five different phases. Table 2 shows the feedback we received from each drone operator regarding the difficulty of the tasks in each phase while controlling the drone via eye tracker. As the results show, different operators had varying experiences. While the first operator's experience is more inclined towards the simplicity, the second one thought they were harder in general, the last task being the hardest. On the other hand, the third operator considered most of the tasks with similar and middle hardship.

In our future work, we plan on testing this experiment with individuals with disabilities. Our hypothesis is that the individuals who have disabilities will have a more effective and efficient run-time because they will have more inspiration and motivation to explore their surrounding environment without the assistance of personal aids or devices.

5 DISCUSSION

Throughout the development of the proposed system, we faced some challenges and learned some lessons. For example, using hard-coded messages was time consuming and more importantly inefficient. To introduce flexibility into our system we plan to migrate to a Robot Operating System (ROS) based implementation of our system. This update is advantageous as it will provide us with the ability to write custom message types; thus, hard-coding IP values will not be needed, and passing values will become simpler.

The Tobii Pro Nano only operates on Mac or PC hosts and the Parrot Olympe SDK only operates on Linux hosts. This discrepancy means that we needed two machines in order to run our system. Communication between these two machines is facilitated using the python socket module. By hosting a socket server on the eye tracker machine we can transmit flight commands to the Olympe machine when requested by a client. Our initial setup used a Macbook which ran the Tobii Pro SDK and a PC which ran Ubuntu 22.04. We decided that it would be beneficial to have one machine that handles the entire system, so we switched to hosting an Ubuntu server virtual machine onto the Macbook. We needed the server edition because Olympe was not designed for ARM architecture so we had to virtualize an x86 machine.

We also came up with an algorithm which allows users to operate the drone with greater precision. The issue with the initial implementation is that the flight commands are constant, meaning when a command is sent, the drone will only move a fixed amount or turn a fixed angle. However, with this new approach we could turn and move the drone relative to the distance of the gaze point to the center of the screen.

By transforming the Active Display Coordinate System such that the origin is placed in the center of the display, we can obtain a decimal value between -0.5 and 0.5 for any x-y coordinate inside the display bounds. By then multiplying this number by 100 and then rounding to an integer, we can compute a speed for the action. This speed value can then be passed into Olympe with the PCMD function, which controls the roll, pitch, yaw, and vertical throttle of the drone.

Since we did not have immediate access to a large open area, real world testing was also challenging. For safety purposes we had to keep the movement speeds of the drone to a reasonable amount, in order to ensure that users had absolute control. This affected the ease of usability and the slowness of the drone is considered as a common issue during our experiments. In our future work, we plan to do more testing in an area such as a large park, so users have the opportunity to operate the drone with a greater movement speed and in larger amounts of position changes without an issue.

6 CONCLUSION

In this study, we introduced an exciting step forward in drone control technology, where UAVs can be controlled effortlessly and intuitively through eye movements, making drone operations accessible to a broader range of users. In addition, the concept of controlling UAVs using eye gestures involves interpreting specific eye movements and patterns as commands to maneuver the drone. This approach can provide a more intuitive and hands-free experience for the drone operator, making it easier to navigate and control the UAV.

Based on our user studies, the proposed system offers a natural hands-free approach that allows the user to operate the UAV in a more calm and effective manner. Not only does this unique advent benefit the user mentally, but physically as well. A user with the ability to operate their two hands more freely has the ability to be more efficient during crucial and critical hours. Other tasks can be performed while also controlling a drone through eye movements. Moreover, instead of risking operators' lives during dangerous search and rescue missions, we envision that our technology can be used to further aid in the outdated operations. Assisting fire fighters with forest fires is another use case scenario where the proposed system can be utilized to reduce the risks and the drones involved during the operation can be controlled efficiently.

For further exploration, we will consider the usage of the proposed system by individuals with disabilities as this system can provide them a chance to explore the world around them without any assistance. They can also increase their efficiency by completing multiple tasks.

We believe a user with this efficient hands free application would not only have an advantage on the ground, but also in the seas. In the near future, we plan on implementing this application for unmanned underwater vehicles (UUV). With this new technology we can save more operators and other emergency responding lives by only risking unmanned vehicles during dangerous operations and tasks.

ACKNOWLEDGMENTS

This work is supported in part by National Science Foundation (NSF) Award# 2050958: REU Site: End-User Programming of Cyber-Physical Systems, and Award# 1815603. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the funding agencies.

REFERENCES

[1] Eyuphan Bulut and Ismail Guevenc. 2018. Trajectory optimization for cellular-connected UAVs with disconnectivity constraint. In *2018 IEEE International Conference on Communications Workshops (ICC Workshops)*. IEEE, 1–6.

[2] Amirahmad Chapnevis and Eyuphan Bulut. 2023. AoI-Optimal Cellular-Connected UAV Trajectory Planning for IoT Data Collection. In *IEEE 48th Conference on Local Computer Networks (LCN)*.

[3] Amirahmad Chapnevis, Ismail Güvenç, Laurent Njilla, and Eyuphan Bulut. 2021. Collaborative trajectory optimization for outage-aware cellular-enabled UAVs. In *2021 IEEE 93rd Vehicular Technology Conference (VTC2021-Spring)*. IEEE, 1–6.

[4] Md Moin Uddin Chowdhury, Sung Joon Maeng, Eyuphan Bulut, and Ismail Güvenç. 2020. 3-D trajectory optimization in UAV-assisted cellular networks considering antenna radiation pattern and backhaul constraint. *IEEE Trans. Aerospace Electron. Systems* 56, 5 (2020), 3735–3750.

[5] Okan Erat, Werner Alexander Isop, Denis Kalkofen, and Dieter Schmalstieg. 2018. Drone-Augmented Human Vision: Exocentric Control for Drones Exploring Hidden Areas. *IEEE Transactions on Visualization and Computer Graphics* 24, 4 (2018), 1437–1446. https://doi.org/10.1109/TVCG.2018.2794058

[6] Zabih Ghelichi, Monica Gentili, and Pitu B Mirchandani. 2021. Logistics for a fleet of drones for medical item delivery: A case study for Louisville, KY. *Computers & Operations Research* 135 (2021), 105443.

[7] John Paulin Hansen, Alexandre Alapetite, I Scott MacKenzie, and Emilie Møllenbach. 2014. The use of gaze to control drones. In *Proceedings of the symposium on eye tracking research and applications*. 27–34.

[8] Mohamed Khamis, Anna Kienle, Florian Alt, and Andreas Bulling. 2018. GazeDrone: Mobile eye-based interaction in public space without augmenting the user. In *Proceedings of the 4th ACM Workshop on Micro Aerial Vehicle Networks, Systems, and Applications*. 66–71.

[9] Saifuddin Mahmud, Xiangxu Lin, Jong-Hoon Kim, Hasib Iqbal, MD Rahat-Uz-Zaman, Sakib Reza, and M Asifur Rahman. 2019. A multi-modal human machine interface for controlling a smart wheelchair. In *IEEE 7th conference on systems, process and control (ICSPC)*. 10–13.

[10] Eleni Mangina, Evan O'Keeffe, Joe Eyerman, and Lizbeth Goodman. 2016. Drones for live streaming of visuals for people with limited mobility. In *22nd International Conference on Virtual System Multimedia (VSMM)*. 1–6.

[11] James Mardell, Mark Witkowski, and Robert Spence. 2009. Detecting search and rescue targets in moving aerial images using eye-Gaze. *COGAIN2009 Proceedings* (2009), 67.

[12] Parrot. 2023. https://developer.parrot.com/docs/olympe/index.html.

[13] Kayla Pineda, Bhanuka Mahanama, Vikas Ashok, and Sampath Jayarathna. 2023. Gaze-Augmented Drone Navigation. In *Proceedings of the Augmented Humans International Conference 2023*. 363–366.

[14] Hazim Shakhatreh, Ahmad H Sawalmeh, Ala Al-Fuqaha, Zuochao Dou, Eyad Almaita, Issa Khalil, Noor Shamsiah Othman, Abdallah Khreishah, and Mohsen Guizani. 2019. Unmanned aerial vehicles (UAVs): A survey on civil applications and key research challenges. *Ieee Access* 7 (2019), 48572–48634.

[15] Tobii Pro Nano Specifications. 2023. https://www.tobii.com/products/eye-trackers/screen-based/tobii-pro-nanospecifications.

[16] Tobii Pro Coordinate System. 2023. https://developer.tobiipro.com/commonconcepts/coordinatesystems.html.

[17] Ginger S Watson, Yiannis E Papelis, and Katheryn C Hicks. 2016. Simulation-based environment for the eye-tracking control of tele-operated mobile robots. In *Proceedings of the Modeling and Simulation of Complexity in Intelligent, Adaptive and Autonomous Systems 2016 (MSCIAAS 2016) and Space Simulation for Planetary Space Exploration (SPACE 2016)*. 1–7.

[18] Liangzhe Yuan, Christopher Reardon, Garrett Warnell, and Giuseppe Loianno. 2019. Human gaze-driven spatial tasking of an autonomous MAV. *IEEE Robotics and Automation Letters* 4, 2 (2019), 1343–1350.