GAZE-AUGMENTED PROXIMITY AND REMOTE DRONE NAVIGATION SCHEMES

Kayla Pineda
Advisor: Sampath Jayarathna

Department of Computer Science, Old Dominion University

Abstract

As the use of unmanned aerial vehicles (UAVs), or drones, continues to increase, the need for effective and efficient control techniques increases. Many human-computer interactive control methods have been explored, but one control technique in development is the use of gaze. We propose a proximity-based control scheme using eye-tracking. In our method, we use gaze extracted from an eye-tracker, allowing the operator to navigate the drone to designated checkpoints. We navigated the drone through a series of waypoints and compared the results to traditional, joystick controls. Ultimately, we found that gaze is a feasible control method for drone navigation. We plan to extend this work to a remote-controlled navigation scheme that uses the object detection model You Only Look Once (YOLO).

Introduction

UAVs or drones are used in various domains and applications, including, but not limited to, military usage, search and rescue, transportation of goods, farming, and building inspection. It is anticipated that the UAV market will exceed $92 billion, surpassing the 2020 value of $9.5 billion [5].

As drones become increasingly popular for everyday use, the number of user interactive methods increases with different types of control methods, such as hand gestures, voice control, and even brain control [21]. A type of control method that is still in development is gaze-augmented control. Gaze-based interactions have a variety of uses such as aiding those with disabilities [14, 24], search and rescue [16], driving [20], programming [22], gaming [13], and simulation [12, 19]. Unlike traditional control mechanisms that use handheld controls, gaze-augmented navigation offers users additional mobility. Moreover, combining autonomous control can prevent potential user errors such as overshooting and undershooting, often associated with traditional controllers.

Related Work

In the context of gaze-based drone navigation, eye-tracking is commonly used as a companion input to another input method or requires additional work to navigate the drone [17]. Gaze-based control follows the same principle of looking in the direction of movement [12]. This creates problems as the user is essentially looking at two places at once; the user must look at the area to navigate to while making sure the device in flight remains stable. One solution is to filter eye movements as input [10, 11]. Using eye movements as input has been compared to the Midas Touch: all of the user's gaze is taken as valid input [9]. As a starting point, some type of on/ off switch can be implemented, requiring some additional input systems or methods. In [8] authors present keyboard controls as the companion input to eye-tracking using a desktop eye-tracker and keyboard input to control four degrees of freedom of the drone: rotation, speed, altitude, and translation. They found that the best control mode was using eye-tracking to control the rotation and speed of the drone and using the keyboard to control the translation and altitude of the drone. For methods that use only eye-tracking as its input, the user is required to do extra steps, such as following specific patterns with their eyes [6]. In [23], authors present single-stroke gaze gestures to navigate a drone through a path. They found that control using gaze gestures did perform slower compared to keyboard, joystick, and dwell time controls, but participants reported gaze gestures required a lower mental workload compared to the other methods. In GazeGuide [4], authors present AR-based drone navigation using eye tracking optics and markers. The work is limited to the maneuvering of the camera with the UAV fixed in a predefined direction.

Methodology

We design our application architecture based on two tasks involved in controlling a drone using eye-tracking; 1) Identify and locate the user's AOI based on gaze, and 2) Search and navigate a drone to the given AOI. Based on this architecture, we compose processes for each task and define AOIs using ArUco markers [7] for the simplicity of the application.

Eye movements were captured using the PupilLabs Core eye-tracking headset with a 200 Hz sampling rate. We used a DJI Tello drone, manufactured by Ryze Robotics. This lightweight drone is equipped with an HD camera. Joystick control is done using the Tello app, and the movement for the drone was programmed using the DJI Tello Python library. The checkpoints were designated by ArUco markers: synthetic square markers with a wide black border and an inner binary matrix that determines its identifier. In our experiment, we used four ID size 5x5 markers that measure 175x175 mm.

Gaze Tracking Process

The gaze tracking process starts by sampling the gaze positions from the eye-tracker of the user for a predefined period. We sample gaze positions along with the field of view (FOV) for 5 seconds. During the period, we ignore samples that correspond to blinks, missing data points, and low-confidence gaze estimates. Then, we scan for markers at each FOV and compute the distances from the gaze location to each marker in the FOV. Finally, we obtain the average distance of each marker and determine the marker corresponding to the gaze location by considering the least distance.

Drone Navigation Process

Our algorithm for drone navigation comprises two steps: 1) Scan for the marker, and 2) navigate to the located marker. During the scanning process, the drone iterates through a pre-defined set of relative angles for which the drone will rotate and scan for the selected marker. Detection of the marker during a scanning step causes the application to start navigating the drone to the marker. Should the selected marker not be detected at all, the user must reselect the marker or choose the next marker in the sequence to be scanned.

For the navigation task, we consider a coordinate system passing through the drone: the x-axis passing through the front to back of the drone, the y-axis passing through the sides of the drone, and the z-axis passing through the top of the drone. From this, we consider three main types of motions for navigating the drone: 1) Horizontal motion (along x-axis), 2) Vertical motion (along z-axis), and 3) Yaw rotation (about z-axis). For the horizontal motion, we use a piecewise function based on the area of the ArUco marker as observed by the drone camera. For the vertical motion and yaw rotation, we use a modified Proportional Integral Derivative (PID) controller based on the work of related setups [3, 2]. To successfully track the markers, these three motions must be in specified ranges. Since the drone's rotation during the initial scanning phase and some sudden movements destabilizes the camera feed, the implementation includes short delays while waiting for camera feed stabilization. Moreover, to counter possible decode errors during transmission, we add retries where we try to read frames from the drone camera.

Baseline Task

During the baseline task, participants were instructed to navigate through a series of markers using virtual joystick controls on a mobile application (see Figure 2 (a)). After a briefing and training session, the proctor revealed the navigation sequence. For each marker, the proctor manually checked the approximate distance to ensure consistency.

Gaze-Augmented Task

The gaze-augmented task uses the same sequence using the eye-tracking control as described earlier. In the experimental setup, we used audio feedback to note key events in the system: start of gaze sampling, end of gaze sampling, arrival at a marker, and completion of marker tracking.

Results

We found that participants favored the eye-tracking control over mobile joystick control. Participants noted that the joystick control was too sensitive, making it difficult to control the drone. Also, the eye-tracking control required little action from the user, whereas the user's attention was required all the time while using the joystick control. Next, we quantitatively compared the two control methods by comparing the marker tracking time and the total time during each trial.

Marker Tracking Time

We define the marker tracking time as the time between proctor instruction and the proctor acknowledgment of success during the baseline task. For the gaze-controlled task, we define it as the time from the drone scanning and reaching the target marker. Then we obtained the mean marker tracking times for each marker for evaluations across all trials.

Marker ID	Joystick	Eye-tracking
0	21.73	33.01
1	19.46	33.50
2	25.31	29.65
3	20.27	27.37
Average	21.69	30.88

Table 1: Average Individual Tracking Times (sec)

Total Time

We define the total time as the time between the drone initialization and the marker tracking success acknowledgment (proctor or audio feedback) of the final marker. We report the average time taken by each user to complete the task using both control mechanisms.

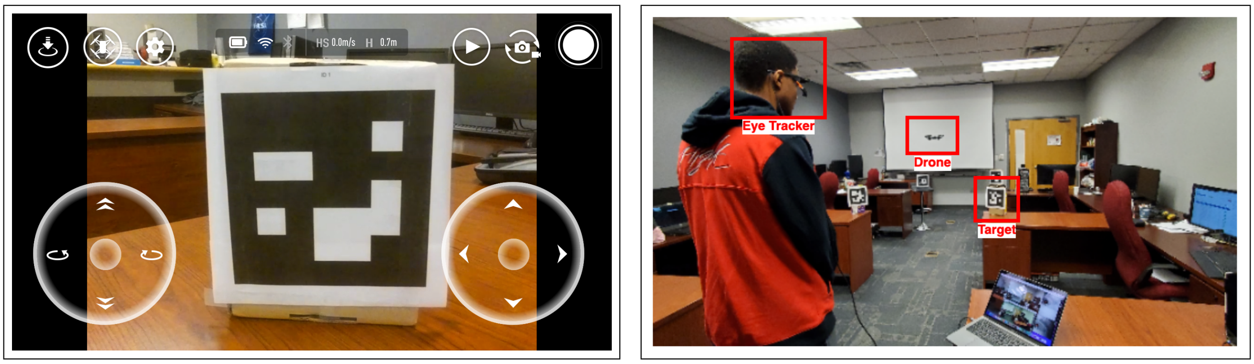

Figure 1: **Experimental Tasks** (a) Navigating drone with virtual controls on the mobile application, and (b) Navigating drone using gaze.

Participant	Joystick	Eye-tracking
1	112.67	121.60
2	107.83	128.11
3	102.48	124.35
4	88.01	140.18
5	114.92	109.02
Average	103.68	123.53

Table 2: Average Total Times (sec)

Remote Navigation Scheme

Our previous work focused on a proximity based navigation scheme using artificial ArUco markers. Our future work plans on adapting that work to a remote navigation scheme using object detection.

Proposed Methodology

In our proposed work, the setup has the user watching the drone's camera feed on a screen. This video is the focal point for the user, and the drone may or may not be in the user's field of view. The user would navigate the drone remotely until they see an object of interest in the video feed. Once the user sees an object they wish to investigate, a similar process to our previous work would start. The user would focus on the object as their gaze is sampled by the eye-tracker. After the chosen object is found, the information would be sent to the drone. The drone would scan for that object and begin tracking it.

Object Detection Using YOLO

The object detection model used is the You Only Look Once, or YOLO, object detection model created by Redmon et. al [18]. The model uses bounding boxes to predict where objects in a scene are located.

There are two types of YOLO models: the ”full” version and a tiny version [1]. In simple terms, the ”full” model has more accurate detections compared to the tiny version at the cost of speed, and the tiny model has faster detections at the cost of accuracy. For the drone, we use the tiny versions of YOLO v3, v4, and v7 pretrained on the MSCOCO dataset [15] to find detections.

Currently, we settled on using YOLO v4 Tiny as our main model.

Discussion

In our initial research, we found our control method took longer compared to typical joystick control. We recognize three reasons as contributing factors to the lower performance. The first factor is the delay introduced by the scanning procedure in our algorithm, which is often time-consuming. The drone does not record the positions of other markers for future

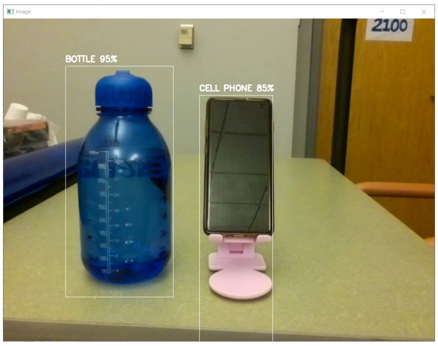

Figure 2: Drone detecting a bottle and phone using YOLO v4 Tiny

use. Thus, the drone must rescan the environment to detect the target marker irrespective of having seen the target previously. In comparison, the scanning time human-controlled approach is negligible. We expect to incorporate a stateful scanning and navigation procedure to eliminate the existing delays. Secondly, we used a strict objective to achieve a tracking success state during gaze-control, resulting in the drone oscillating to adjust the position. In contrast, we used a subjective measure, the proctor's judgment, for confirmation when using joystick controls. Finally, the gaze-controlled approach encompasses delays for camera and gaze stabilization, which are not present in joystick controls.

Our proposed work would combine our previous work with gaze-augmented drone navigation with object detection using YOLO. In our previous work, we created a proximity gaze-based drone navigation framework and had promising results. We now plan to create a remote gaze-based drone navigation framework. The user will focus on object in the drone's camera feed, prompting the drone to track that object. The YOLO object detection model will allow us to track real world objects in real time, allowing us to expand our work to real world applications.

Conclusion

Our proposed work would combine our previous work with gaze-augmented drone navigation with object detection using YOLO. In our previous work, we created a proximity gaze-based drone navigation framework and had promising results. We now plan to create a remote gaze-based drone navigation framework. The user will focus on object in the drone's camera feed, prompting the drone to track that object. The YOLO object detection model will allow us to track real world objects in real time. We expect that this research will allow us to expand our work to real world applications.

Acknowledgements

This work was supported in part by NSF CAREER 2045523, NSF REU Supplement 2214096, and Virginia Space Grant Consortium Undergraduate STEM Research Scholarship for Kayla Pineda.

References

[1] ADARSH, P., RATHI, P., AND KUMAR, M. Yolo v3-tiny: Object detection and recognition using one stage improved model. In *2020 6th international conference on advanced computing and communication systems (ICACCS)* (2020), IEEE, pp. 687–694.

[2] ATTIGUI, M. Autonomous drone programming feature detection, tracking, and obstacle avoidance.

[3] BABU, V. M., DAS, K., AND KUMAR, S. Designing of self tuning pid controller for ar drone quadrotor. In *2017 18th International Conference on Advanced Robotics (ICAR)* (2017), pp. 167–172.

[4] BN, P. K., BALASUBRAMANYAM, A., PATIL, A. K., AND CHAI, Y. H. Gazeguide: An eye-gaze-guided active immersive uav camera. *Applied Sciences 10*, 5 (2020), 1668.

[5] DALY, D. Drone industry outlook for the united states: 2020-2030. *Consortiq* (Nov 2020).

[6] DREWES, H., DE LUCA, A., AND SCHMIDT, A. Eye-gaze interaction for mobile phones. In *Proceedings of the 4th International Conference on Mobile Technology, Applications, and Systems and the 1st International Symposium on Computer Human Interaction in Mobile Technology* (New York, NY, USA, 2007), Mobility '07, Association for Computing Machinery, p. 364–371.

[7] GARRIDO-JURADO, S., MUÑOZ-SALINAS, R., MADRID-CUEVAS, F. J., AND MEDINA-CARNICER, R. Generation of fiducial marker dictionaries using mixed integer linear programming. *Pattern recognition 51* (2016), 481–491.

[8] HANSEN, J., ALAPETITE, A., MACKENZIE, I., AND MØLLENBACH, E. The use of gaze to control drones.

[9] JACOB, R. J. K. The use of eye movements in human-computer interaction techniques: What you look at is what you get. *ACM Trans. Inf. Syst. 9*, 2 (apr 1991), 152–169.

[10] JAYAWARDENA, G., AND JAYARATHNA, S. Automated filtering of eye gaze metrics from dynamic areas of interest. In *2020 IEEE 21st International Conference on Information Reuse and Integration for Data Science (IRI)* (2020), IEEE, pp. 67–74.

[11] JAYAWARDENA, G., AND JAYARATHNA, S. Automated filtering of eye movements using dynamic aoi in multiple granularity levels. *International Journal of Multimedia Data Engineering and Management (IJMDEM) 12*, 1 (2021), 49–64.

[12] KIM, J., PALMISANO, S., ASH, A., AND ALLISON, R. Pilot gaze and glideslope control. *TAP 7* (06 2010).

[13] KREJTZ, K., BIELE, C., CHRZASTOWSKI, D., KOPACZ, A., NIEDZIELSKA, A., TOCZYSKI, P., AND DUCHOWSKI, A. Gaze-controlled gaming: Immersive and difficult but not cognitively overloading. In *Proceedings of the 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing: Adjunct Publication* (New York, NY, USA, 2014), UbiComp '14 Adjunct, Association for Computing Machinery, p. 1123–1129.

[14] KRISHNA SHARMA, V., SALUJA, K., MOLLYN, V., AND BISWAS, P. Eye gaze controlled robotic arm for persons with severe speech and motor impairment. In *ACM Symposium on Eye Tracking Research and Applications* (New York, NY, USA, 2020), ETRA '20 Full Papers, Association for Computing Machinery.

[15] LIN, T.-Y., MAIRE, M., BELONGIE, S., HAYS, J., PERONA, P., RAMANAN, D., DOLLÁR, P., AND ZITNICK, C. L. Microsoft coco: Common objects in context. In *Computer Vision–ECCV 2014: 13th*

European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13 (2014), Springer, pp. 740–755.

[16] MARDELL, J., WITKOWSKI, M., AND SPENCE, R. Gaze-contingent enhancements for a visual search and rescue task. In *Proceedings of the ACM SIGGRAPH Symposium on Applied Perception in Graphics and Visualization* (New York, NY, USA, 2011), APGV '11, Association for Computing Machinery, p. 109.

[17] PFEIFFER, C., WENGELER, S., LOQUERCIO, A., AND SCARAMUZZA, D. Visual attention prediction improves performance of autonomous drone racing agents. *Plos one 17*, 3 (2022), e0264471.

[18] REDMON, J., DIVVALA, S., GIRSHICK, R., AND FARHADI, A. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (2016), pp. 779–788.

[19] SMOLYANSKIY, N., AND GONZALEZ-FRANCO, M. Stereoscopic first person view system for drone navigation. *Frontiers in Robotics and AI 4* (2017).

[20] TALL, M., ALAPETITE, A., SAN AGUSTIN, J., SKOVSGAARD, H. H., HANSEN, J. P., HANSEN, D. W., AND MØLLENBACH, E. Gaze-controlled driving. In *CHI '09 Extended Abstracts on Human Factors in Computing Systems* (New York, NY, USA, 2009), CHI EA '09, Association for Computing Machinery, p. 4387–4392.

[21] TEZZA, D., AND ANDUJAR, M. The state-of-the-art of human–drone interaction: A survey. *IEEE Access 7* (2019), 167438–167454.

[22] THOMSCHKE, A., STOLPE, D., TAEUMEL, M., AND HIRSCHFELD, R. Towards gaze control in programming environments. In *Proceedings of the Programming Experience 2016 (PX/16) Workshop* (New York, NY, USA, 2016), PX/16, Association for Computing Machinery, p. 27–32.

[23] YU, M., LIN, Y., SCHMIDT, D., WANG, X., AND WANG, Y. Human-robot interaction based on gaze gestures for the drone teleoperation. *Journal of Eye Movement Research 7* (09 2014), 1–14.

[24] ZHANG, G., HANSEN, J. P., MINAKATA, K., ALAPETITE, A., AND WANG, Z. Eye-gaze-controlled telepresence robots for people with motor disabilities. In *Proceedings of the 14th ACM/IEEE International Conference on Human-Robot Interaction* (2019), HRI '19, IEEE Press, p. 574–575.