



**HAL**  
open science

# Real-time Eye-Tracking Processing during Pilot-UAV Interaction

Gaganpreet Singh, Raphaëlle N Roy, Caroline P C Chanel

► **To cite this version:**

Gaganpreet Singh, Raphaëlle N Roy, Caroline P C Chanel. Real-time Eye-Tracking Processing during Pilot-UAV Interaction. 1st International Workshop on Eye-Tracking in Aviation, Mar 2020, Toulouse, France. 10.3929/ethz-b-000407660 . hal-04927589

**HAL Id: hal-04927589**

**<https://hal.science/hal-04927589v1>**

Submitted on 3 Feb 2025

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Conference Paper

## Real-time Eye-Tracking Processing during Pilot-UAV Interaction

**Author(s):**

Singh, Gaganpreet; Roy, Raphaëlle N.; Chanel, Caroline P.C.

**Publication Date:**

2020-03

**Permanent Link:**

<https://doi.org/10.3929/ethz-b-000407660> →

**Rights / License:**

[Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 International](#) →

This page was generated automatically upon download from the [ETH Zurich Research Collection](#). For more information please consult the [Terms of use](#).

# Real-time Eye-Tracking Processing during Pilot-UAV Interaction

Gaganpreet Singh<sup>✉\*</sup>, Raphaëlle N. Roy<sup>\*</sup>, and Caroline P. C. Chanel<sup>\*</sup>

<sup>\*</sup>ISAE-SUPAERO, Université de Toulouse, France

Eye tracking has become a valuable tool to observe and interpret operators' mental state. This may be due to the fact that it is less intrusive than other physiological tools, and also capable of assessing various mental states such as mental fatigue and engagement (e.g. using blink frequency and fixation duration). These states are critical in aeronautical contexts. Therefore, a real-time eye-tracking processing system was developed using eye-tracking glasses worn by the pilot while performing flight and interacting with unmanned aerial vehicles (UAVs). The proposed system is capable of finding areas of interest from different positions, orientations, and distances from a continuously changing field of view. Four computer screens were tracked and in one of them three different areas of interest were defined and tracked in real-time. The goal is to ease the pilot's mental state assessment in ecological settings and ultimately to enhance the pilot-UAVs interaction in Manned-Unmanned Teaming missions.

*Keywords & Phrases:* eye-tracking, pilot, UAV, fixations, manned-unmanned teaming

## 1 INTRODUCTION

Aviation is particularly concerned with the problems of over-engagement and missing critical information such as alarms in hyper active situations. The National Transportation Safety Board (NTSB) and the International Civil Aviation Organization (ICAO) have emphasised that poor flight monitoring by pilots is one of the main causes of major civil aviation accidents [1]. Moreover, Dehais and collaborators have demonstrated in their study the advantages of eye tracking in aviation and how it could be a valuable tool for explicit training of attention allocation to increase flight safety [1, 2]. Hence, features like fixations, saccades, blink frequency and duration, and pupil dilation can provide real-time, fast and minimally intrusive insights of one's mental states like mental fatigue [3–5], workload [6–8] and attentional engagement [9, 10]. This valuable information has several applications to bring safety as well as productivity with a wider interaction channel between humans and machines. Although pilots (flying and remote) have been extensively studied using eye-tracking in recent years, to our knowledge no study has yet assessed pilots' mental state in scenarios in which he/she has to interact with a fleet of UAVs while flying, like in MUM-T missions. This is the purpose of the study that Singh and collaborators have designed [11]. The global study intends to assess pilots' workload level using various physiological features, but in this particular article we only focus on the eye-tracking part, and more especially on the implementation of the eye-tracking system for such a task.

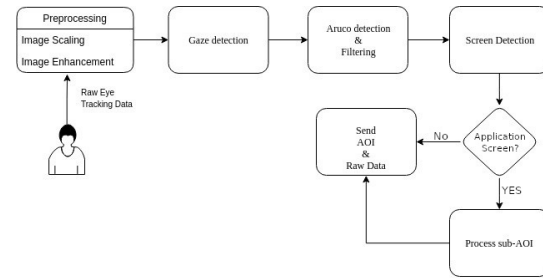
Eye-tracking in ecological settings is a challenge in itself, at both its hardware and software levels [12]. Desktop based eye-trackers that have a bar like structure with built in infrared sensors are pre-fixed in front of the person and have to face the person to track at all times. This leads to a very small field of view of just one screen, but provides precise localization of the gaze on the screen. Moreover, it enables real-time processing [13]. On the other hand, wearable eye-trackers in the form of glasses have a wider field of view, with the help of built-in cameras and eye-tracking infrared sensors that provide what a person is viewing along with the gaze with respect to the image. These systems' data are more accurate because of the placement of sensors just near an eye, yet their real-time processing is complicated and they are currently used for post analysis [13].

---

✉Contact: [gaganpreet.singh@isae-supero.fr](mailto:gaganpreet.singh@isae-supero.fr)



**Figure 1:** System setup



**Figure 2:** Eye tracking and processing flow diagram

Therefore, to have a system that could be used in real-time, during pilot-UAVs operations, and that would also be able to provide gaze information from a wider viewing field, one has to either use several desktop based eye-trackers, one for each screen, or to create an eye-tracker from scratch by assembling several cameras and infrared sensors in a desired area. This could be time consuming, expensive, and hard to install new devices in already designed and tested environments. The other solution is to use glasses with an embedded eye-tracking system that allows in-the-field measurements and to work on the processing of the acquired data to allow an online extraction of features such as fixation ones. Therefore, the focus of this article is to deal with the data processing performed to tackle the following challenges: unrestricted field of view, real-time eye tracking and processing, defining and finding gaze position in one of the areas of interest from any position, orientation, and distance from a continuously changing field of view.

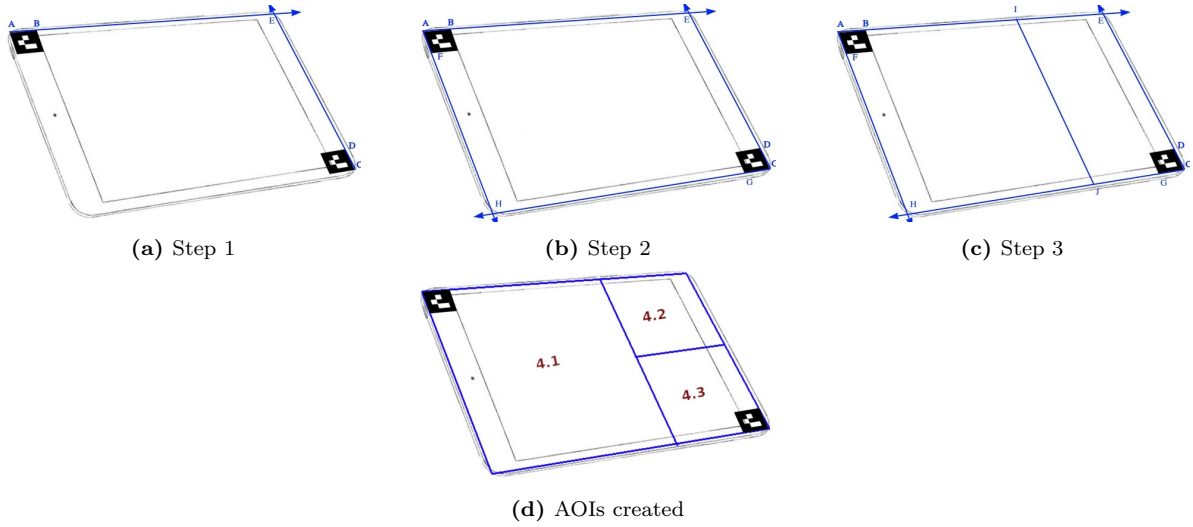
## 2 SYSTEM DESIGN

The system is comprised of three 23 inch computer screens (numbers 1, 2, 3) placed horizontally together to form a joined image for the flying task, running Aerofly simulator. A 13 inch computer tablet (number 4) was also used to allow interaction with the UAV fleet, through an in-house U-track application. It shows the geolocation of the plane flown by the pilot, along with that of the UAVs. For eye tracking purposes, 'Tobii pro glasses 2' was used, at 100 Hz for eye data and 25 Hz for scene camera data. It uses a one point calibration procedure, and connects to the system via WiFi. Alongside, to identify areas of interest (AOIs), aruco codes were used and placed at the top left and bottom right corners of each screen (4 screens in total were used). The whole system setup is shown in figure 1.

### 2.1 Eye Tracking and Processing

The workflow of the system is divided into 6 main stages which are shown in figure 2. For the eye-tracking process, image frames (scene camera) and the eyes data from the tracking glasses are first collected and sent to the '*Preprocessing*' stage. During this step the image is first scaled to half of its length and height to facilitate faster processing. Then to facilitate and enhance aruco code detection process, image is brightened by shifting the colour format from RGB to HSV, and maximizing the 'V' field of the HSV colour format. Next, at the '*Gaze detection*' stage both image (scene camera frames) and eye data (gaze position of both eyes, pupil size, etc.) are synchronized. This is done by calculating the time difference between image and eye data. Selected samples have a time difference of less than the time between two consecutive image frames.

After getting the overlapping image frame and the eye tracking data, the image frame is then sent to the '*Aruco detection & Filtering*' stage. In this stage, the detection of the aruco codes in the image frame is performed, as well as a filtering phase allowing to remove those codes which are not associated



**Figure 3:** Step by step sub-AOI processing for the U-track application screen

with the screen currently under focus. More precisely, in this stage all visible aruco codes are detected in the image frame, however only two codes are extracted: one with an odd id with respect to which gaze position is present below and at the right side; and the other with an even id, with respect to which gaze position is present at the top and at the left side of the aruco code. It defines the two aruco codes for which the gaze is supposed to be present in between. Thus, these two codes are matched in the next stage called '*Screen Detection*' to confirm if they belong to the same screen or not. These steps give the screen at which the person is looking at, at a particular point of time.

As explained in the previous section, in one of the four screens (screen 4 in figure 1) is running the UAV interaction application U-Track. In this screen, 3 different sub Areas of Interest (AOIs) are defined, which are named AOI4.1, AOI4.2, and AOI4.3. If the person is identified as looking at the application screen, the gaze data is further processed to identify the focused area of interest out of three sub-AOIs. In the '*Processing sub-AOI*' stage, the principal challenge is to deal with the fact that the eye tracking glasses are not fixed in the 3D space, which makes the position and orientation of the screens in the image frame dynamic. That means if the person moves close to the screens their size increases in the image frame and if the person moves far from the screens their size becomes small. Note that the head movement also changes the angles of the screens in the image frame generating roll, yaw, and tilt dynamics.

## 2.2 Dynamic Sub-system Processing

The identification of the AOIs' of the U-track application screen (number 4) is done with the help of 4 coordinates given by each aruco code. From aruco detection and filtering stage, we get 2 aruco codes with 4 corners, each corner with x and y coordinates in the image plane. One aruco code is fixed at the top left (Code1) and one at the bottom right (Code2) of the application screen. Then with the help of Code1, a line passing from top left and top right corner (A and B respectively) of aruco Code1 is computed (figure 3(a)), and y-intercept of line AB is computed with equation 2:

$$m_1 = \frac{B_y - A_y}{B_x - A_x} \quad [Slope \ of \ line \ AB] \quad (1)$$

$$c_1 = A_y - (m_1 * A_x) \quad [y \ intercept \ of \ line \ AB] \quad (2)$$

Then with the help of Code2, a line segment passing from the bottom right and top right (C and

D respectively) corners of aruco Code2 is computed and its y intercept is computed with equation 4 (figure 3(a)):

$$m_2 = \frac{D_y - C_y}{D_x - C_x} \quad [\text{Slope of line } CD] \quad (3)$$

$$c_2 = C_y - (m_2 * C_x) \quad [\text{y intercept of line } CD] \quad (4)$$

And then coordinates x and y of intersection point of these 2 line segments are computed with equation 5 and 6:

$$x = \frac{c_2 - c_1}{m_1 - m_2} \quad [x \text{ intersection point}] \quad (5)$$

$$y = (m_1 * x) + c_1 \quad [y \text{ intersection point}] \quad (6)$$

This gives the top right corner of the screen. However, to deal with divide by zero problem when either of the lines is in vertical position (x at every location is same) an extremely small value of 0.0000001 is assigned to x difference ( $B_x - A_x$  &  $D_x - C_x$ ) of x coordinates while computing the slope. This will still keep the slope computation correct up to 6 decimal points, while eliminating the divide by zero problem.

Similarly, 2 line segments are computed using the above formulas, one passing from the top left and bottom left corners of the aruco Code1 and another passing from the bottom right and bottom left corners of the aruco Code2. Then their intersection point is computed as shown in figures 3(b). This gives the bottom left corner of the screen 4, and provides the outer boundaries of the application screen. Then, to divide the screen into three sub-AOIs, top and bottom lines are divided into a 70-30 ratio with the following formulas:

$$x = x_1 + [(x_2 - x_1) * 0.7] \quad (7)$$

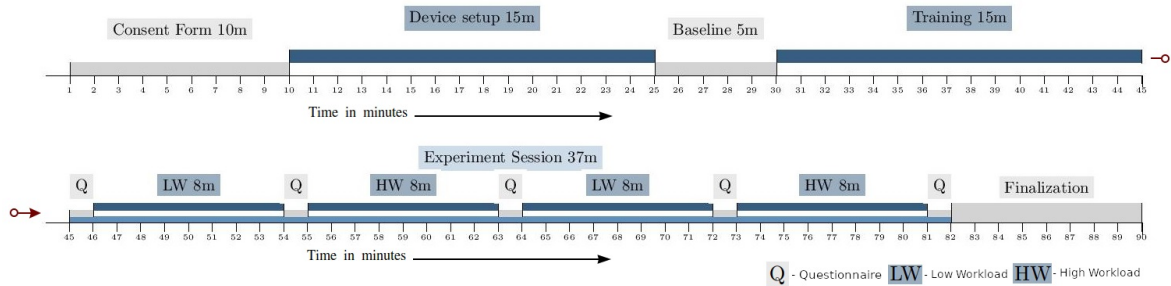
$$y = y_1 + [(y_2 - y_1) * 0.7] \quad (8)$$

Next by joining these points we create two halves of the screen, shown in figure 3(c). Then the right and middle lines are divided into 60-40 ratios, and joined to create two halves of the right area, shown in figure 3(d). This creates 3 sub-AOIs (named 4.1, 4.2, & 4.3) in the application screen, shown in figure 3(d) and is adaptable to dynamic changes in dimension, roll, yaw, and tilt.

### 3 EXPERIMENT

Other than several individual testings, we also performed a real-time experiment with one female participant of 23 years of age, who performed the pilot flying task while interacting with the UAVs through the U-track application in the 4th screen (Figure 5). The experimental protocol was validated by our local ethics committee (id number 2019-137). The experiment was divided into four 8-minute experimental blocks, with 1-minute breaks between blocks, and an additional break in the very beginning. Hence, the whole experiment and data collection lasted for 37 min, 16 min of high and 16 min of low workload (Figure 4). During the breaks, the application screen showed a white dot on a black background on which the participant was instructed to concentrate. Two blocks corresponded to the low workload condition, and the two remaining ones to the high workload condition. Their order was pseudo-random. The participant had to perform 3 tasks: a detection and identification task, a working memory task and a flying task as detailed below:

- *Detection and identification task*: Simulated the process of UAV requests in the form of pop-ups in the U-track application in screen 4 (AOI 4.2). A beep sound initiates the request and the pilot has to search if there is any human present in a group of 9 gray scale images extracted from the Norb database [14], and has to answer with Yes or No.
- *Working memory task*: Air Traffic Control (ATC) instructions are played in the form of audio messages with heading for the plane and communication channels for UAVs. The pilot has to remember all the information and put back communication channels for the UAVs in AOI 4.3.
- *Flying task*: The pilot has to remember the heading instruction from the ATC commands and maintain that heading of the plane in the Aerofly simulator using the joystick while avoiding restricted zones in red.



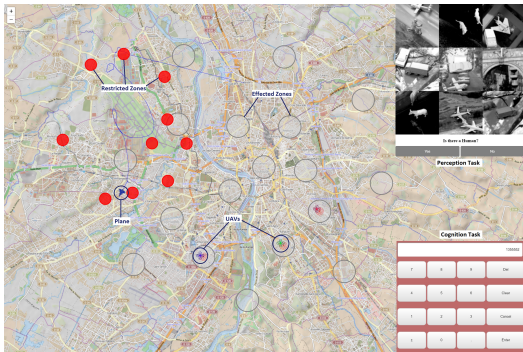
**Figure 4:** Experiment timeline

Workload was varied thanks to modifications in both the working memory and the flying tasks. Indeed, the Low workload and High workload conditions were different in: the number of communication channel inputs (i.e. 1 vs 2 randomly selected UAVs), and the distance of the restricted zones from the plane's path (i.e. very big vs small). Additionally, the path selected for the low workload condition comprised less sharper turns as compared to the high workload condition.

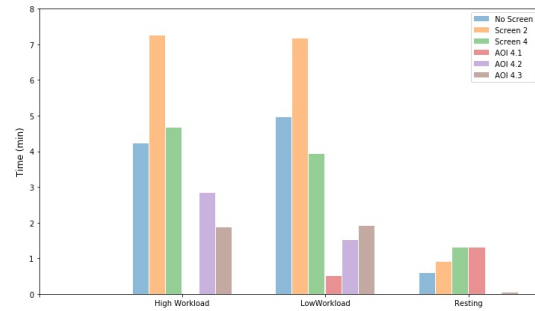
## 4 RESULTS

Preliminary results showed that the computation of the aruco codes is very fast and quite reliable in real-time conditions. The processing of AOIs is also very adaptable to orientation and dimensional changes of the image frames. Even 180 degrees of roll of the image frame has not affected the detection and processing, and creation of the AOIs. In this 37 min experiment, 54170 samples of eye tracking data with 44 features (gaze position, pupil size, gyroscope values, and other) per sample were collected. Out of this dataset, 15595 samples have a no screen detection. This could be due to the algorithm which is not able to detect aruco codes in the view (false negatives), or the participant was really not looking towards any screen (true positives).

Moreover, figure 6 shows the gaze position (not fixation) that corresponds to each screen for each workload condition. It is to be noted that gaze position shown in figure 6 for screen 4 is the sum of the gaze position of each of its sub-AOIs i.e. AOI4.1, AOI4.2, and AOI4.3. Moreover, the time spent by the participant on Screen 2 was approximately the same in both high and low workload conditions. However, an increase in workload resulted in an increase of the time spent on Screen 4 (the U-track application screen containing all three sub-AOIs) and a decrease of the time spent outside of the screens. This might be due to the fact that the demands of both the UAV and flying task increased and the participant engaged her resources to meet these demands to perform the task. Moreover, it is also seen in the results that in the high workload condition, the time spent on detection and identification tasks as well as on working memory task increases. On the other hand a decrease in the time spent on Screen 4 and an increase of 'no screen' time is the indicator of a decrease in engagement, which can



**Figure 5:** UAV interaction application U-track



**Figure 6:** Gaze present in screens 2, 3 and 4, sub-AOIs of screen 4 (as 4.1, 4.2, & 4.3), and also outside the screens for each workload condition.

eventually increase the occurrence of mind wandering episodes. In the resting period the participant spent most of her time staring at Screen 4, which was expected given the instructions.

## 5 DISCUSSION & FUTURE WORK

The designed system is able to perform exceptionally well, as required. Indeed, this system is also able to handle orientation and dimensional changes pretty well and has detected roll up to  $180^\circ$  (tested value) and we are confident it could perform even higher than  $180^\circ$ . Moreover, this system is able to perform correct detection of gaze position for 72% of the time. However, regarding the remaining 28%, i.e. 15595 samples that are labeled 'no screen', they could correspond to correct detection (true positives) or wrong ones (false negatives). A validation algorithm is currently under development to assess this properly by checking the previous gaze position and previously detected aruco codes, and then compare with the change in gaze position and change in gyroscope values (that is head movement, which is also provided by the eye tracker). This way, one could compute if the gaze is still inside the last detected position of the screen and the algorithm has missed the arucos, or if the gaze is actually out of the screen, and can therefore verify false negatives and true positives. Moreover, there are a lot of dimensions one can go in the future with this work, such as eliminating the use of aruco codes and defining the areas of interest with image recognition algorithms. One would have to provide areas of interest at the beginning as images and then use recognition to find out if the gaze is present in any of these areas. However, the use of aruco codes is serving well in our project and can be very good in adapting to any new system changes. Moreover, our interest in the future is to compute eye features such as fixations, blink frequencies, blink rates and saccades from raw data extracted from eye-tracker in real-time (it is to be noted that current results in figure 6 show gaze in each recorded sample, not fixation). These ocular features will be extracted alongside other physiological measures such as cerebral and cardiac activity measurements in order to perform mental state estimation of the human pilot and then model an intelligent system to enhance human-robot interaction and to increase mission performance.

## REFERENCES

- [1] Frédéric Dehais, Julia Behrend, Vsevolod Peysakhovich, Mickaël Causse, and Christopher D Wickens. Pilot flying and pilot monitoring's aircraft state awareness during go-around execution in aviation: A behavioral and eye tracking study. *The International Journal of Aerospace Psychology*, 27(1-2):15–28, 2017.



- [2] Vsevolod Peysakhovich, Frédéric Dehais, and Andrew T Duchowski. Why is eye tracking an essential part of neuroergonomics? In *Neuroergonomics*, pages 27–30. Elsevier, 2019.
- [3] Giuseppe Barbato, Vittoria De Padova, Antonella Raffaella Paolillo, Laura Arpaia, Eleonora Russo, and Gianluca Ficca. Increased spontaneous eye blink rate following prolonged wakefulness. *Physiology & behavior*, 90(1):151–154, 2007.
- [4] Kenichi Kaneko and Kazuyoshi Sakamoto. Spontaneous blinks as a criterion of visual fatigue during prolonged work on visual display terminals. *Perceptual and Motor Skills*, 92(1):234–250, 2001.
- [5] TL Morris and James C Miller. Electrooculographic and performance indices of fatigue during simulated flight. *Biological psychology*, 42(3):343–360, 1996.
- [6] Gianluca Borghini, Laura Astolfi, Giovanni Vecchiato, Donatella Mattia, and Fabio Babiloni. Measuring neurophysiological signals in aircraft pilots and car drivers for the assessment of mental workload, fatigue and drowsiness. *Neuroscience & Biobehavioral Reviews*, 44:58–75, 2014.
- [7] Morris K Holland and Gerald Tarlow. Blinking and mental load. *Psychological Reports*, 31(1):119–127, 1972.
- [8] Yuu Tanaka and Kiyoshi Yamaoka. Blink activity and task difficulty. *Perceptual and motor skills*, 77(1):55–66, 1993.
- [9] Tony Renshaw, Richard Stevens, and Paul D Denton. Towards understanding engagement in games: an eye-tracking study. *On the Horizon*, 2009.
- [10] L Sciulli, Parimal S Bhagat, and Charlene P Bebeko. Eye tracking analysis: Engagement levels and donor tendencies using print advertisements with emotional appeals. *Innovative Marketing*, 8(2):91–98, 2012.
- [11] Gaganpreet Singh, Raphaëlle N Roy, and Caroline Ponzoni Carvalho Chanel. Towards multi-uav and human interaction driving system exploiting human mental state estimation. In *10th International Conference on Bioinformatics Models, Methods and Algorithms*, 2019.
- [12] Peter Kiefer, Ioannis Giannopoulos, Martin Raubal, and Andrew Duchowski. Eye tracking for spatial research: Cognition, computation, challenges. *Spatial Cognition & Computation*, 17(1-2):1–19, 2017.
- [13] Haifeng Bao, Weining Fang, Beiyuan Guo, and Peng Wang. Real-time eye-interaction system developed with eye tracking glasses and motion capture. In *International Conference on Applied Human Factors and Ergonomics*, pages 72–81. Springer, 2017.
- [14] Yann LeCun, Fu Jie Huang, and Leon Bottou. Learning methods for generic object recognition with invariance to pose and lighting. In *Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2004. CVPR 2004.*, volume 2, pages II–104. IEEE, 2004.