DOI 10.5162/ettc2020/4.1

EYE-GAZE INTERFACE TO OPERATE AIRCRAFT DISPLAYS

LRD Murthy[1], MD Babu[2], V Yelleti[3], KPS Saluja[1], P Biswas[1]
[1] Indian Institute of Science, Bangalore 560012, India,
[2]Aircraft and Systems Testing Establishment, Indian Air Force, Bangalore, India
[3] Indian Institute of Information Technology, Kalyani, India
dillibabu_mohan@yahoo.com, pradipta@iisc.ac.in

Abstract:

Eye gaze trackers have been studied widely for their utility in aviation domain since long time. So far, numerous studies have been conducted in the direction of gaze-controlled interfaces for electronic displays in flights and Head mounted display systems under simulated conditions. In this paper, we present our study on usage of eye gaze trackers in real flight conditions and their failure modes under such usage conditions and illumination. We show that the commercially available off the shelf (COTS) eye gaze trackers with state-of-the-art accuracy fails to provide gaze estimates beyond certain level of illumination on the eyes. We also show that the limited available tracking range of eye gaze trackers limit them to provide gaze estimates even during the pilots' natural operating behavior. Further, we present three approaches of developing eye gaze trackers which are designed to use webcam instead of infrared illumination and are aimed to be functional at high illumination conditions. We present our intelligent tracker, developed using OpenFace framework, provides comparable results to COTS eye tracker in terms of interaction speed for both indoor and outdoor conditions.

Key words: Eye gaze tracking, Eye gaze estimation, Gaze controlled interface, Human Machine Interaction, Cockpit Display

1. Introduction

Eye gaze tracking is the process of estimating where a person is looking at. Eye gaze tracking technology is used to understand eye gaze scanning process, visual search and reading behavior from late 18th century. With availability of portable infrared based eye trackers, researchers also explored controlling digital user interfaces just by looking at it. While the technique sounds intuitive, underlying technical limitations like innate eye movement process and the constraints on direct manipulation of graphical user interfaces so far restricted eye gaze controlled interfaces to limited applications for people with severe disability [3] and as a binary input channel (on/off) for a few smartphone video viewing applications.

In recent time, a set of new applications were explored for eye gaze controlled interfaces where the operators' situation impedes him/her to operate traditional physical or touch screen user interfaces. Examples of such situations include undertaking secondary tasks in automotive [2, 8] and mission tasks inside combat aircrafts [10, 11].

Eye gaze tracking technology is widely explored in aviation domain for pilot training, understating pilots' scanning behavior, optimizing cockpit layout and recently to estimate pilots' cognitive workload [6, 9]. Although a commercial product is not yet available, different defense manufacturers are already investigating to use eye gaze-controlled interfaces inside cockpit. While most of the research on eye gaze-controlled interfaces for military aviation concentrate on Head Mounted Display Systems (HMDS) [4, 11], this paper explores use of eye gaze controlled interface for Head Down Display. The paper initially explored a state-of-the-art wearable eye tracking device in a combat aircraft undertaking representative combat maneuvers and then proposed and evaluated a set of algorithms for developing screen mounted eye gaze tracker for operating head down displays. These algorithms can also be used in transport and passenger aircrafts.

The paper is organized as follows. The next section summarizes our earlier published research on eye gaze-controlled interfaces followed by the study on actual aircraft. Section 4 presents different eye gaze tracking algorithms followed by concluding remarks.

2. Earlier Research

So far, we set up and undertook numerous investigations to study the utility of eye gaze tracking in aviation field. Our studies are set in different conditions like a fixed base flight simulator, a virtual reality (VR) based simulators. We also performed user studies in actual transport and military aircrafts. Our studies focused on evaluating pilot's interaction experience for mission control tasks using gaze-controlled interfaces in simulators and actual flights.

Initially, we set up a flight simulator and evaluated gaze-controlled interface for military fast jets [10]. Our studies found that the gaze-controlled interface statistically significantly increased the speed of interaction for secondary mission control tasks compared to touchscreen and joystick-based control systems. Subsequently, we tested the gaze-controlled system inside a transport aircraft both on ground and in-flight with military pilots. We found that the pilots could undertake representative pointing and selection tasks in less than 2 secs on average.

We also integrated the COTS eye tracker to a flight simulator to evaluate the feasibility of its functions as a part of HMDS and evaluated the latency incurred to point at an aerial target. In parallel, we measured pilots' various ocular parameters like fixation rate, pupil dilation and saccadic intrusion in standard and VR based flight simulators at various G levels in different combat aircrafts. We compared these metrics in various flight phases and during air to ground attack maneuvers [1].

In addition to utilizing COTS based eye trackers for our user studies, we studied the accuracy of gaze control in various acceleration due to gravity (G load) conditions under various flight maneuvers. We used a COTS eye-gaze tracker in an advanced jet trainer aircraft and we found that the error is less than 5º of visual angle up to +3G although it is less accurate at -1G and +5G [7]. In line with our error measurements of COTS eye gaze trackers under various flight conditions, in the next section, we present our analysis on various failure modes of COTS eye gaze trackers in actual flight conditions.

3. Testing COTS tracker in Combat Aircraft

We have recorded data from two flights using the COTS eye tracker (Tobii Pro Glasses 2) which uses infra-red (IR) illumination-based eye gaze estimation principles [12]. The duration of the first flight is 55 minutes 58 seconds *(Flight 1)* and another flight's duration is 56 minutes *(Flight 2)*, the flight profiles are furnished in Table 1 below. The eye tracker contains a front-facing scene camera which records the first

Table 1 Flight Profiles

Sl No	*Objective*	*Profile*
Flight #1	Maneuvering flight with head mounted eye tracker on Pilot in Command	Take-off – climb – level flight to Local Flying Area – Constant G (3G and 5G) level turns both sides each – Vertical loop – Barrel Roll – Air to Ground dive attack training missions – Descent – ILS Approach and landing
Flight #2	Non - Maneuvering flight with head mounted eye tracker on Pilot in Command	Take-off – climb – level flight to Local Flying Area – Straight and Level cruise with gentle level turns – Descent – ILS Approach and landing

person view of the pilot. It also contains four eye-cameras, two cameras per each eye, to record the eye movements. The eye tracker estimates gaze points at a frequency of 100 Hz. The frame rate of scene camera is 25.01 frames/second at 1920 x 1080 resolution and that of each eye camera is around 50 frames/second with a resolution of 240x240. Each gaze point is recorded with a dedicated identifier, called "gidx". We initially used *Tobii Pro Lab* tool to analyze the recorded gaze samples and observed that both flight recordings contain gaze samples only for around 50% of the duration. We investigated this loss of data samples during the flight using the raw data provided by manufacturer in json format and by correlating the raw data with the eye images.

The raw data obtained in json format contains various other information recorded during the flight like gyroscope and accelerometer data. We discarded the irrelevant information and retained the data points required for our investigation of lost gaze points.

At first, we synchronized the raw data stream and the eye camera stream in time scale since eye camera stream starts off with an offset from raw data. This is achieved using the Position Time Stamps (PTS) provided in both data streams. We also find that the different frequencies of these two streams is a challenge for data synchronization. Hence, we considered the time duration between two successive frames of eye camera stream and consider all the corresponding gaze data points recorded during that time window. Thus, the latter frame and these data points together form one pair of synchronized data points. Each time windowed raw data may contain multiple gaze points. Every gaze point with its "gidx" contains a status code,

‘s’ which indicates the error associated to that datapoint, if any. The status code 0 indicates no error and any non-zero value of s indicates an error associated is with the data point. We observed that all the gaze points with a non-zero status code are recorded as zeros for both x and y directions [0.0, 0.0]. The gaze points are provided in normalized values, hence the minimum gaze point is [0.0, 0.0] and the maximum is [1.0,1.0].
We segmented the synchronized data points into two categories. The first category *category1* contains eye stream frames whose corresponding gaze points have zero status code. The second category *category2* contains those eye frames with all corresponding gaze points with non-zero status codes. There are frames whose data points have only a subset of gaze points contains zero status code. We did not consider these frames in our analysis as it brings uncertainty on eye image tagging.

For *Flight 1*, we observed that out of 167,647 frames, only 57,111 frames fall under *category1* and 69,732 frames fall under *category2*. For *Flight 2*, we observed that out of 167,567 frames, only 81,911 frames fall under *category1* and 51,402 frames fall under *category2*.

Summarizing, 41.6% of the frames does not have any gaze points recorded during *Flight 1* and for *Flight 2*, this stands at 30.7%. Further, if we just look at unsynchronized raw data, both flights recorded more than 51% of the gaze samples are error-prone.

We visually inspected these flight recordings and we hypothesize two reasons for this data loss.

1. Higher levels of illumination on eyes may affect the eye tracker resulting in no gaze estimation.
2. Limited field of view (FoV), especially in the vertical direction, renders the eye tracker with no gaze estimates when user looks beyond the tracking range.

We validated our hypothesis 1 using the eye images in the above mentioned two categories. Since the recorded video stream is an IR video, we converted all eye images into grayscale and computed average of all the pixel values for each image present in both categories. Figure 1 represents the histogram of image intensities for *category1* and *category2* for Flight 1. Figure 2 represents the same for Flight 2. Figure 1a indicates that 93% of the images under *category1* have an average intensity less than 131. But, *category2* contains 42% with average

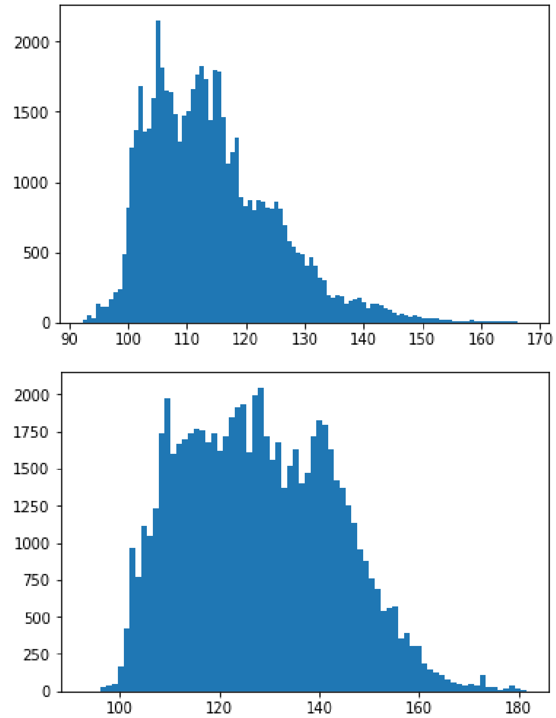


Figure 1. Histogram of image intensities for Flight 1 1a. *Category1*(Top) 1b. *Category2* (Bottom)

intensity higher than 131. Further, this can also be observed in Flight 2 case, shown in Figure 2. Around 42% in *category2* have higher intensity than 150, while *category1* contains 94% of the images with intensity less than 150. This indicates that images with higher illumination, precisely above 131 in Flight 1 and above 150 in flight 2 have low probability to obtain accurate gaze estimates.

While this evidence supports our hypotheses 1 partially, we observed that there is overlap in the left and right histograms plotted in Fig 1 and Fig 2. Hence, we could not identify a clear average image intensity threshold in order to identify all the failure modes of eye gaze estimation.
We further investigated the data points in *category2* to understand the 58% of the datapoints which have lower image intensities than above mentioned thresholds for each flight using our hypotheses 2. Since we observed that the gaze estimates are lost for a sequence of eye image frames, we clustered the datapoints in *category2* based on their “gidx”s. If a sequence of datapoints under *category2* are having successive gidx’s, then all those points are considered as a single cluster. Hence, each cluster can contain one datapoint or several datapoints. Extending our hypotheses 2, we assumed that the pilot must be looking at a

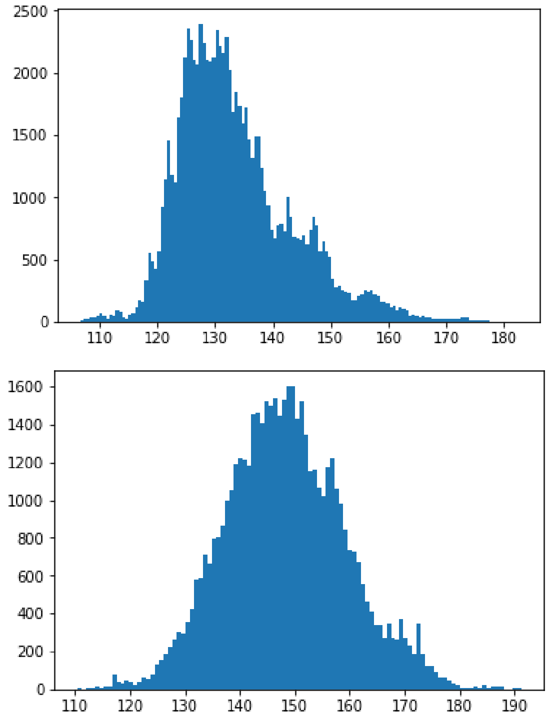


Figure 2. Histogram of image intensities for Flight 2 2a. *Category1*(Top) 2b. *Category2* (Bottom)

position closer to the extreme tracking positions (beyond which eye tracker cannot track), just before or after the eye tracker fails to provide gaze estimates. During our visual inspection of first person video recorded using eye tracker, we observed that the pilot looks down for various activities like looking at the information displayed in the Multi-functional displays (MFD)s while keeping the head faced horizontal (perpendicular to the vertical axis of the aircraft). During such scenarios, we observed that gaze points were not recorded.

Hence, we analyzed three preceding or subsequent datapoints adjacent to each cluster, which we refer to as boundary datapoints. We looked for boundary datapoints with gaze values beyond 0.8 and less than 0.2 (in both x and y). If any of the boundary datapoints satisfy above criterion, then we may infer that the loss of gaze points is due to the pilot looking beyond the tracking range of the eye tracker.

For flight 1, we obtained 12178 clusters for 69,732 datapoints. For these clusters, 11,865 (97.43%) clusters have boundary points that satisfy the above criterion. To understand image intensities for these datapoints, we plotted a histogram of the image intensities for the datapoints whose boundary points satisfy above criterion. We observed that these image intensities lie in the range of [96,145]. This is clearly in the overlap range identified between Figure 1a and Figure 1b.

Similarly, for flight 2, we obtained 8646 clusters for 51,402 datapoints. For these clusters, 8408 (97.24%) clusters have boundary points that satisfy the above criterion. Interestingly here as well, we observed that the histogram of image intensities for the above points lie in the range of [117,164], which is the range of overlap identified in Figure 2a and Figure 2b.

Thus we infer that, this eye gaze tracker could not identify beyond certain illumination level or if the user is looking beyond its tracking range. We should note that the pilot is performing his assigned tasks during the flight and maintained his natural behaviour. This indicates that the tracking range offered by this eye tracker is not sufficient for military aviation environments.

Hence, using our two hypotheses and the raw data, we studied the failure modes of eye gaze tracker in aviation environment. We further add that, while commercial off-the shelf eye trackers may be used in real aviation environments, researchers and practitioners should keep in mind about both the horizontal and vertical tracking range of the eye tracker and it's robustness to external illumination as there is a high chance that the illumination varies rapidly at high altitudes in high speed maneuvers.

4. Developing Eye Gaze Trackers

In this section, we have described three different eye tracking systems and compared them through user studies Initially we described the different algorithms used for estimating gaze followed by two user studies.

HoG based Gaze Tracking System

We used a pre-trained facial landmark detector with iBUG 300-W dataset [iBUG 2019], which works on classic Histogram of Oriented Gradients (HoG) feature combined with a linear classifier to detect facial landmarks [Rosebrock 2019]. In comparison, Haar cascades are a fast way to detect an object but often detect more false positives compared to HoG and linear classifier [Dalal 2005]. HoG features are capable of capturing the face or object outline/shape better than Haar features. On the other hand, simple Haar-like features can detect regions brighter or darker than their immediate surrounding region better. In short, HoG features can describe shape better than Haar features and Haar features can describe shading better than HoG features. In this case the shape is more important as we need the landmarks of the face hence HoG features produced a better result. After detecting eye region, we detected pupil location by selecting the smallest rectangle possible in the eye region where the pupil can exist. After retrieving pupil locations, we calculated the Eye Aspect Ratio (EAR). We have noted that the eye aspect ratio changes with respect to the distance between the user and the camera. We have modified the EAR calculation formula by using the distance between the two eyes as denominator.

Webgazer.js

We implemented a second system using webgazer.js [Webgazer 2020; Agrawal 2019] to compare performance of the proposed system. Webgazer.js runs entirely in the client browser. It requires a bounding box that includes the pixels from the webcam video feed that corresponds to the detected eyes of the user. This system uses three external libraries (clmtracker, js_objectdetect and tracking.js) to detect face and eyes. It has methods for controlling the operation which allows us to start and stop it. We have taken the mean of last thirty points from webgazer.js for better target prediction and accuracy of system. We also calculated the mean value during this time to predict the gaze location on a webpage.

Intelligent System

We have developed a gaze block estimator which maps user's eye movements to 9 screen blocks using OpenFace [Baltrusaitis 2018] toolkit. Since the OpenFace (figure 3) was reported to have an error of 6^0 for gaze point estimation, we designed a calibration routine which uses the gaze vector data from OpenFace and maps user's eye movements to screen blocks, instead of screen pixels. We have divided the screen into 9 blocks of equal area. We designed a smooth pursuit based calibration routine where a marker traverse across all these 9 blocks and user was asked to follow the marker's movement. The corresponding gaze vectors from OpenFace were recorded and stored with the respective block number as the label. Once the marker completes its path, a neural network is trained to map these gaze vectors to 9 blocks of the screen. For this classification task, we used a 2 hidden layer network with 256 and 128 neurons respectively with cross-entropy loss function and with Adam optimizer. We used the 70% of the data we recorded during calibration for training, 15% for validation and the rest for testing. On a i7 processor computer, we observed that each epoch takes around 0.8 seconds and we trained the network till the test accuracy reaches 90%.

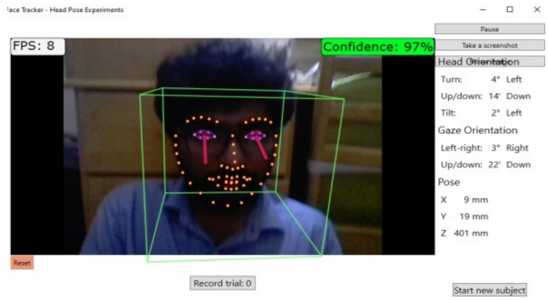


Figure 3. Screenshot from OpenFace Face Tracker

User Study

We undertook the following user study to compare different eye tracker implementations in different lighting conditions and compared them with a COTS screen mounted eye gaze tracker.

Participants: We collected data from 9 participants (8 male, 1 female). All participants were recruited from our university. They do not have any visual or motor impairment.

Material: The user trial was conducted on a Microsoft surface pro tablet powered by dual-core processor and it comes with 8 GB RAM and running Microsoft Windows 10 operating system. The surface has a 5 MP camera, which was used to estimate gaze direction.

Design: We wanted to use the eye tracker to operate a graphical user interface with limited number of screen elements, hence instead of traditional precision and accuracy measurement, we calculated the pointing and selection times for a set of fixed positions in screen.

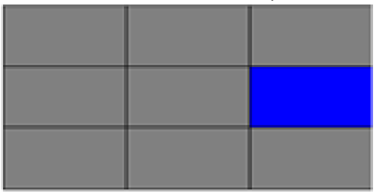
We created a user application in which we divided the screen into nine blocks and one of the blocks gets randomly highlighted with blue colour as shown in figure 4a. If the user clicks on the blue block, it turns green as shown in figure 4b and a different block was highlighted. If the user is unable to click on the highlighted block within 10 seconds, it turns randomly some other block to blue. Using this interface, we calculated the response time by measuring the time difference between appearance of a highlighted block and its selection. Users selected target using the left mouse button.

The trial was performed twice - once in laboratory with lux meter reading 180-200 lux and the other in outdoor condition with lux meter reading between 1800-3000 lux.

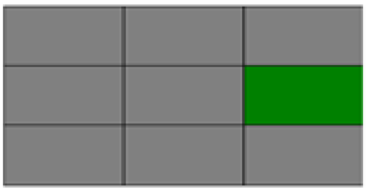
The trial consisted of four eye tracking implementations

- HoG based bespoke webcam based gaze tracker
- Webgazer.js based webcam based gaze tracker

- OpenFace based intelligent webcam based gaze tracker
- Tobii PCeye mini eye-gaze tracker (referred hereafter as COTS tracker)



a) Randomly highlighted block in blue



b) After click, highlighted block turns green

Figure 4. Pointing Task application

For all trial conditions, we conducted trial on the same user application discussed before. The order of conditions was randomized to minimize practice or learning effect. Each participant undertook all trial conditions.

Results: We recorded 322 pointing tasks inside room and 270 pointing tasks outside. We measured the time difference between onset of a target and its correct selection. We removed outliers by identifying points greater than outer fence. We removed only one point from the selection times recorded from the bespoke eye gaze tracker while 12 data points were found values higher than outer fence for the COTS eye gaze tracker.

Figure 5 presents the average selection times and standard deviation. Participants took lowest time to select target using the COTS tracker. We undertook a 2 × 2 unbalance factorial regression based ANOVA (type of eye gaze trackers × lighting conditions) on the response times.

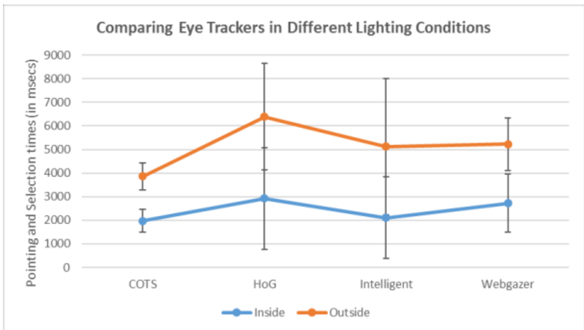


Figure 5. Comparing Average Response Time among different eye trackers

We found

- significant main effect of type of eye gaze tracker $F(3,567)=15.44$, $p<0.01$
- significant main effect of lighting condition $F(1,567)=4.05$, $p<0.05$
- significant interaction effect of type of eye gaze tracker and lighting condition $F(3,574)=3.45$, $p<0.05$

Then we undertook two one-way ANOVAs for each lighting condition and found significant main effect of eye gaze tracker implementations.

- Inside room $F(3,318)=8.43$, $p<0.01$
- Outside room $F(3,266)=11.31$, $p<0.01$

Finally, a pair of unequal variance t-tests did not find any significant difference between COTS tracker and our intelligent eye gaze tracker implementation inside room at $p<0.05$ although the difference in response times between the intelligent system and COTS tracker was significant at outside condition at $p<0.05$.

Discussion: Our initial approach for eye gaze estimation used feature based approach. We used a method that extracts Histogram of Oriented Gradients (HoG) features combined with a linear SVM to detect eye landmarks. These landmarks were used to compute Eye Aspect Ratio (EAR) feature to estimate the gaze block on the screen. Even though HoG based landmark detection had been used earlier widely, we observed that it occasionally failed to detect landmarks of our users and affected the gaze estimation accuracy. Variations in illumination and appearance of facial features like beard or spectacles could affect tracking accuracy based on pre-selected facial features.

The second approach, Webgazer.js proposes to map the pixel data of eye images directly to gaze locations rather than to rely on handcrafted features from eye images. They used a 6 x 10 eye image patch for each eye and converted them to 120-dimensional feature vector. This vector is used as an input for a regression model to map to gaze points on screen. This approach also relies on multiple eye landmark detection algorithms which suffers similar limitations as HoG based algorithms. Further, this approach requires users to click at least 40-50 locations on screen for calibration purpose before it can make predictions which requires significant time.

OpenFace uses a state-of-the-art deep learning approach for landmark detection and gaze estimation. It uses Constrained Local Neural Field (CLNF) for eye landmark detection and tracking. Unlike HoG and Linear SVM approach, which was based on handcrafted features and trained on a relatively smaller dataset, OpenFace uses larger dataset and deep learning approach to learn the estimation of 3D gaze vector from the eye images. Even though OpenFace is not very accurate in predicting gaze points on screen, it does not suffer from illumination and appearance to detect eye landmarks. Further, OpenFace was implemented in C++ which makes real-time gaze estimation possible even on CPUs. In addition to these, the reported state-of-the-art cross-validation accuracy prompted us to test for a gaze block detection application.

It may be noted that the response times were lowest for the COTS tracker compared to webcam-based approaches. The COTS tracker used on board ASIC chip to run image processing algorithms, which has lower latency than general purpose processor used in webcam-based eye trackers [5]. It may be noted that still the difference in response times was not significant inside room between COTS and intelligent tracker. Our future work is investigating to further reduce the latency and make eye tracking work in bright lighting condition.

5. Conclusions

This paper presents a case study of testing and development of bespoke eye gaze tracker for operating Head Down Displays in an aircraft cockpit. Our study showed that present COTS eye gaze trackers are not yet ready to be integrated to combat aircraft in terms of tracking eye gaze at different lighting conditions and vertical field of view. We presented a set of algorithms that can be configured for operating multi-function displays inside cockpit and a particular intelligent algorithm using the OpenFace framework worked better than classical computer vision based algorithms.

References

[1] Babu MD, JeevithaShree DV, G. Prabhakar and P. Biswas (2019), Using Eye Gaze Tracker to Automatically Estimate Pilots' Cognitive Load, 50th International Symposium of the Society for Flight Test Engineer (SFTE).

[2] Biswas P. (2016), Using Eye Gaze Controlled Interfaces in Automotive Environments, Springer, 2016.

[3] Biswas, P. and Langdon, P. (2013), A new interaction technique involving eye gaze tracker and scanning system, ACM International Conference Proceeding Series pp. 67-70.

[4] De Reus A.J.C. , Zon R. and Ouwerkerk R.2012. Exploring the use of an eye tracker in a helmet mounted display, Avionics Europe Conference & Exhibition, Munich, Germany, March 21-22, 2012.

[5] Duchowlowski A. T.(2007), Eye Tracking Methodology. Springer, 2007

[6] Fitts, P.M.; Jones, R.E.; Milton, J.L. Eye movements of aircraft pilots during instrument-landing approaches. Aeronaut. Eng. Rev. 1950, 9, 24–29.

[7] Murthy LRD, Mukhopadhyay A., Yelleti V, Arjun S, Thomas P, Babu MD, Saluja KPS, JeevithaShree DV and Biswas P, Evaluating Accuracy of Eye Gaze Controlled Interface in Military Aviation Environment, IEEE Aerospace 2020

[8] Prabhakar G, Ramakrishnan A, Murthy LRD, Sharma VK, Madan M , Deshmukh S and P Biswas, Interactive Gaze & Finger controlled HUD for Cars, Journal of Multimodal User Interface, Springer, 2020

[9] Rudi, D., Kiefer, P., Giannopoulos, I., & Raubal, M. (2019). Gaze-based interactions in the cockpit of the future: a survey. Journal on Multimodal User Interfaces, 1-24.

[10] Shree, D.V., Murthy, L.R.D., Saluja, K.S. and Biswas, P., 2018. Operating Different Displays in Military Fast Jets Using Eye Gaze Tracker. Journal of Aviation Technology and Engineering, 8(1), p.31.

[11] Thomas, P., Biswas, P. and Langdon, P. (2015), State-of-the-art and future concepts for interaction in aircraft cockpits, Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics) 9176, pp. 538-549.

[12] Tobii Pro Glasses 2 Product Description. 2019. Retrieved 28th May 2019 from https://www.tobiipro. com/siteassets/tobii-pro/product-descriptions/tobii-pro-glasses-2-product-description.pdf/?v=1.95