

## USAGE OF EYE TRACKER TECHNOLOGY IN EXAMINING ATTENTION DISTRIBUTION OF OPERATORS OF UNMANNED AERIAL VEHICLES

**Zoya Hubenova<sup>1</sup>, Konstantin Metodiev<sup>1</sup>, Svetla Dimitrova<sup>1</sup>, Liubomir Alexiev<sup>2</sup>**

<sup>1</sup>Space Research and Technology Institute – Bulgarian Academy of Sciences

<sup>2</sup>Military Medical Academy, Sofia, Bulgaria

e-mail: [komet@space.bas.bg](mailto:komet@space.bas.bg)

**Key words:** *Flight Simulator, Eye Tracker, Computer Vision, Unmanned Aerial Vehicle*

### **Abstract**

*This article proposes yet another approach towards looking into causes for attention distribution of an operator of unmanned aerial vehicle. During examination, the operator is being tested at dedicated flight simulator while data are gathered and visualized through a mobile eye tracker. Two work stages are considered sequentially, i.e. building a geometric 2D transformation of region of interest (homography) within an image, and overlaying a dynamic heatmap as well. In the former stage, spontaneous movements of the operator's head, recorded by the video, are eliminated thus enabling the operator to use the mobile eye tracker instead of a desktop-based one. During the latter stage, the distribution of operator's attention over time is displayed.*

*In order to implement the current research, a source code has been developed in C++ for some features readily available in OpenCV library to be used. In addition, data gathered after carrying out flight session are processed and discussed thoroughly.*

### **Introduction**

Currently, an increasing number of countries around the world are making use of Unmanned Aerial Vehicles (UAV) for the reason that these vehicles are able to keep expenditures low and maintain multi role functionalities simultaneously. Possible applications of UAV, beyond the military one, are quite diverse. These include monitoring of oil pipelines and dams, protection of forests, monitoring of hazardous sites, etc. To control the UAV, it takes the operator a wide range of hands-on experience largely identical to the conventional pilot's skills. From that point of view, UAV operators' training done by means of flight simulators seems to be a pivotal stage in developing even better airplane control habits.

The growing number of UAVs, starting from the smallest and getting to the largest, make both public and relevant authorities in a number of countries pay

increasing attention to training a new type of pilots. These pilots, in particular, neither are subject to high load factor nor exposed to danger.

The operator makes all subsequent decisions based on analysis of the situation. Therefore, it becomes clear that this process is the most critical one both in terms of reliability of making optimal decisions (within the system of detected events, objects, processes) and requirements to be met in order to provide for real-time functionality. The human factor, for example in aviation, is still the main cause of most accidents and fatalities; hence, the problem of objective evaluation of operator training progress is quite acute.

Human role in ensuring proper operation of unmanned aerial systems is essential. It involves organization of process planning, maintenance and usage of the control system, evaluating UAV effectiveness and achieved results. What is more, the “Operator – UAV – Computer” interface within the control system architecture is considered fundamental. Choice and certification of such an interface as well as organization of the relevant operator training is further bounded up with developing special programs and centers.

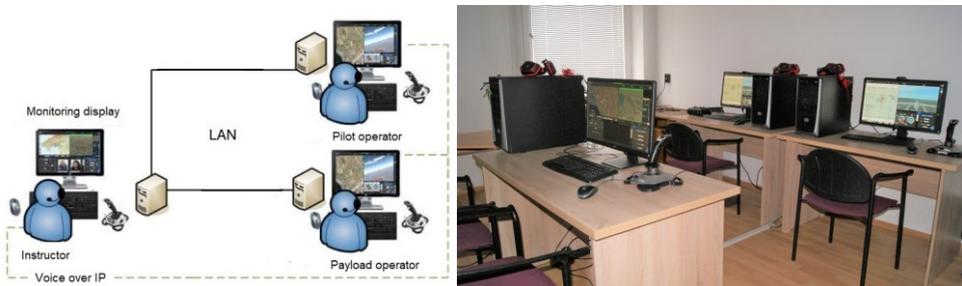
The accompanying software developed for the study needs is a replicated solution to the so-called “Heatmap Explorer”, developed in Python and thoroughly described in [1]. Although the source code is freely available to everybody for downloading (Bitbucket), it is incompatible with new releases of Pupil software, and deprived from cross-platform functionality. The proposed alternative code was developed in C++ starting from the very scratch, borrowing the same idea, and simplifying the implementation to some extent. In addition, some functionality available in OpenCV library, [2], was applied in order to facilitate the development process further.

## **Method**

Training UAV operators by means of simulators appears to be one of the most important factors in ensuring safe operation of the UAV. Training process itself will minimize negative impact of the human factor by diminishing possibility of erroneous actions made by the UAV crew. In general, there are two main concepts in preparing and training UAV operators: training at dedicated simulators (specifically designed for this purpose), and providing real ground control station with a training mode.

Research is carried out in laboratory for selection, training and control of unmanned aerial vehicle operators at Space Research Institute, Bulgarian Academy of Sciences. The laboratory is equipped with simulator C-Star developed by Israeli firm SimLat, [3]. The simulator offers a virtual environment that largely resembles the real one. In addition, the simulator is able to register and store dynamics of interaction between crew members. It is also possible for the instructor to train each crew member (either pilot or payload operator) separately. In this mode, the

instructor replaces the absent crew member and does her/his duties. For example, in case of training the payload operator, it is the instructor who controls the airplane, thus completing the relevant pilot task. On the other hand, whenever pilot is being trained, the instructor handles the payload. The user interface is intuitive, user friendly. Every trainee who is somewhat skillful in handling conventional computer peripherals will succeed in mastering the simulator too, Fig. 1.



*Fig. 1. The C-Star simulator layout (left) and our equipped laboratory*

The simulator basic configuration is composed of working stations for instructor and trainees (pilot and payload operator), one station each. The pilot is trained for controlling and monitoring the UAV which in turn may fly in either of modes: manual or automatic. The trainee monitors not only the airplane and onboard systems but also the flight task. The payload operator is trained in handling the payload (EO / IR camera), identifying and tracking targets in initially designed scenario. The simulator has an intercom for communication between the instructor and the trainees.

In order to examine distribution of the operator's visual attention, we make use of the eye-tracking technology so as to record and evaluate eye movement. The main task of the eye tracking research is recording and transmitting data in real time regarding oculomotor activity. Eye-tracking (oculography) is a research method employing a device to track the trainee's gaze point coordinates or eye movements during task performance. Currently, the most popular eye trackers use a video-based method for non-invasive and non-contact measurement of the eye movements to be made possible.

Eye movement tracking indicators can be applied to various aspects of oculomotor behavior, depending on type of analysis. The most common metrics is based on fixations and / or saccades. The fixations are presented as discrete portions of near-stable points the eye stares at. The saccades are defined as eye movements between fixations. In order to record the aforementioned eye movement parameters, the research team benefited from a mobile eye tracker.

The mobile eye tracker used in the current study is Pupil Labs Core, [4], Fig. 2.



*Fig. 2. Pupil Labs Core binocular headset*

The head set consists of a diode Smartled® SFH4050 IR, emitting light at 850 nm wavelength, an environmental camera, and two removable lenses with fixed focal length and field of view 60 and 100 deg. It was a binocular version of the eye tracker that research team used in the present study, Fig. 2. The binocular implementation is said to produce more robust data in terms of both calibration and measurement. What is more, to carry out the calibration stage is considered highly desirable, not obligatory. The accompanying software provides a readily available calibration values by default. It is also possible for the user to collect data by means of an independent mobile device, such as a tablet or a smartphone, in order to make the process of data gathering smooth. The obtained video is processed afterwards for the accompanying software to be capable of discovering the calibration markers (if any) and mapping the gaze point offline.

Broadly speaking, a heatmap is computed as a Kernel Density Approximation meant for estimating the probability density function of gaze points. There are many examples of symmetrical kernels, such as Epanechnikov, Gaussian, triangle, rectangle, etc. A unit density of Gauss kernel is estimated according to following formula, [5]:

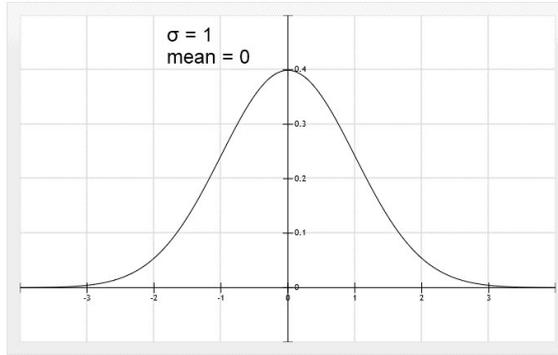
$$(1) \quad K(t) = \frac{1}{\sqrt{2\pi}} \exp(-t^2/2)$$

A one-dimensional Gauss kernel is shown in Fig. 3.

The bivariate estimate, in its most widespread form, depends upon two variables. It is presented as a sum of kernel products in direction of both axes:

$$(2) \quad \hat{f}(x, y) = \frac{1}{Nh_x h_y} \sum_{i=1}^N K\left(\frac{x_i - x}{h_x}\right) K\left(\frac{y_i - y}{h_y}\right)$$

In eq. (2)  $\{x_i, y_i\}$  are sample (gaze) points,  $\{x, y\}$  are pixel coordinates,  $h_x$  and  $h_y$  are smoothing coefficients, and  $N$  is the sample size.



*Fig. 3. Gauss kernel, mean = 0, standard deviation = 1*

For each image pixel, the algorithm calculates exponentially declining gaze intensity relevant to a center of distribution (point of gaze) with coordinates  $\{x_i, y_i\}$  according to formula (2). A pixel is not to be processed if it lies outside  $\pm 3\sigma$  interval. At this point, the algorithm assumes zero gaze intensity to speed up the computational process. After "scanning" all pixels, the algorithm superimposes the heatmap on the image so that the kernels are visible "from above." Finally, the complete heatmap is remapped within the interval 0 ... 1.

The so-called Short Rainbow color map has been used to represent gaze intensity at given frame, Fig. 4.



*Fig. 4. Short rainbow color map*

The heatmap color depends upon its normalized value. Values closer to one are represented by red color while values closer to zero are depicted in blue. In addition, two intermediate colors are visible in Fig. 4, i.e. cyan RGB(0, 255, 255) and yellow RGB(255, 255, 0). It is possible to compute any intermediate color by applying a linear interpolation to selected channels (Red, Green, Blue) separately. In order to work out the transitional color, in terms of two end colors and a weighing variable, it is sufficient to make use of following simple formula:

$$(3) \quad Lerp(start, end, weight) = (end - start) * weight + start$$

$$0 \leq weight \leq 1 \quad 0 \leq start \leq 255 \quad 0 \leq end \leq 255$$

During color computation, four transitional zones might be figured out with regard to the heatmap value. Depending on zone, some channels are held constant whilst others undergo interpolation. Prior to interpolation, the heatmap value should be remapped within 0 ... 1 interval so as to get usable quantity of the weighing coefficient in formula (3). Transitional zones and color channel definitions are shown in Table 1.

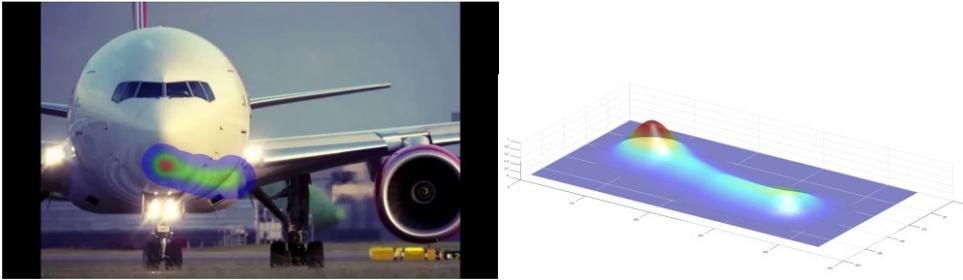
Table 1

Transitional Zone	Heatmap Value, f	Color channels		
		Red	Green	Blue
Blue to Cyan	[0; 0.25)	0	Lerp(0, 255, f)	255
Cyan to Green	[0.25; 0.5)	0	255	Lerp(0, 255, 1-f)
Green to Yellow	[0.5; 0.75)	Lerp(0, 255, f)	255	0
Yellow to Red	[0.75; 1]	255	Lerp(0, 255, 1-f)	0

The algorithm in Table 1 is implemented by means of if ... else if ... else branching. In this way, a smooth transition between two arbitrary end colors is achieved.

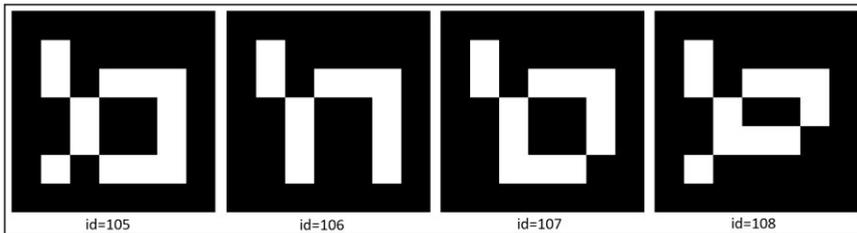
The transparency level is commonly set within 0 ... 255 range and subsequently applied to each color channel. For example, given pixel RGB channels of (180, 195, 192) and alpha channel of 40% transparency. In order to overlay a red pixel (255, 0, 0), derived from the heatmap, the old pixel color and the red one have to be “merged.” The new color might be computed according to expression:  $0.4 * (180, 195, 192) + 0.6 * (255, 0, 0) = (225, 78, 77)$ . In this way, the resulting color channels do not exceed amount of 255 each.

In Fig. 5, an example of dynamic heatmap is visible, so are the normalized Gaussian kernels, right half. Several normalized kernels are shown at a time. The red zone depicts the highest value of gaze intensity whilst the blue one denotes the lowest. The heatmap maximum value is always equal to 1 (read peak). Eventually, the picture to the left is overlaid with the heatmap layer set to a certain transparency level beforehand.



*Fig. 5. A heatmap consisting of several Gaussians: top view (left) and perspective*

During the video recording stage, the eye tracker user is not expected to be standing still which is the reason why objects appear with varying coordinates on the environmental video. The region of interest is not exception to the rule; hence, the bounding rectangle is to be measured by the algorithm frame by frame. One way of doing it is making use of four fiducial markers, either Aruco, [6], [7], or AprilTag, placed at the region corners. OpenCV is capable of detecting the markers and computing their coordinates on the screen. Four markers define a quadratic region of interest for it to undergo a perspective alignment later on (cv::warpPerspective method). Whenever a key is pressed, the region of interest is stretched so that it can fill up the viewing surface, i.e. a screen. In this way, part of the frame outside the region of interest is removed. The operator's spontaneous head shake is compensated. Exemplary Aruco markers used in the current study are shown in Fig. 6, id 105, 106, 107, 108, dictionary id 16.



*Fig. 6. Aruco markers used in the presented study*

In addition, exemplary AprilTag markers are shown in Fig. 7, id 0, 14, 22, 17, dictionary id 36h11

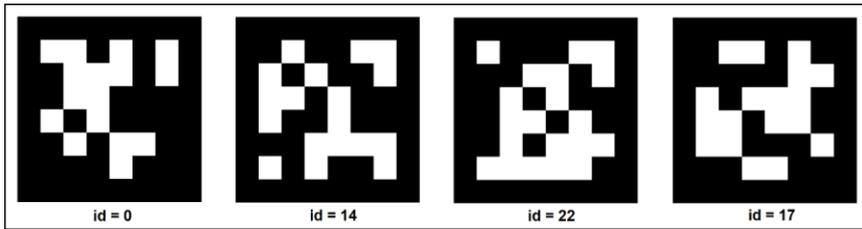


Fig. 7. AprilTag markers used in the presented study

Due to short focal length of lens, the video frame appears distorted with barrel distortion dominating. In order to make up for it, a compensation must be carried out (cv::undistort method) preceded by a compulsory calibration. In the current study, a readily available example in Aruco module has been used to carry out the camera calibration stage, yielding in turn the so-called camera matrix and distortion coefficients vector.

In Fig. 8, the C-Star pilot station screen is shown. For the purpose of this experiment, the screen has been divided into four groups (information sources) according to what type of information each group outputs. Broadly speaking, following groups might be distinguished: Group 1 – navigation map; Group 2 – video channel; Group 3 – instrumentation panel; Group 4 – onboard sensors data. The number of fixations is to be counted for each information group separately.



Fig. 8. Information sources available at the pilot station

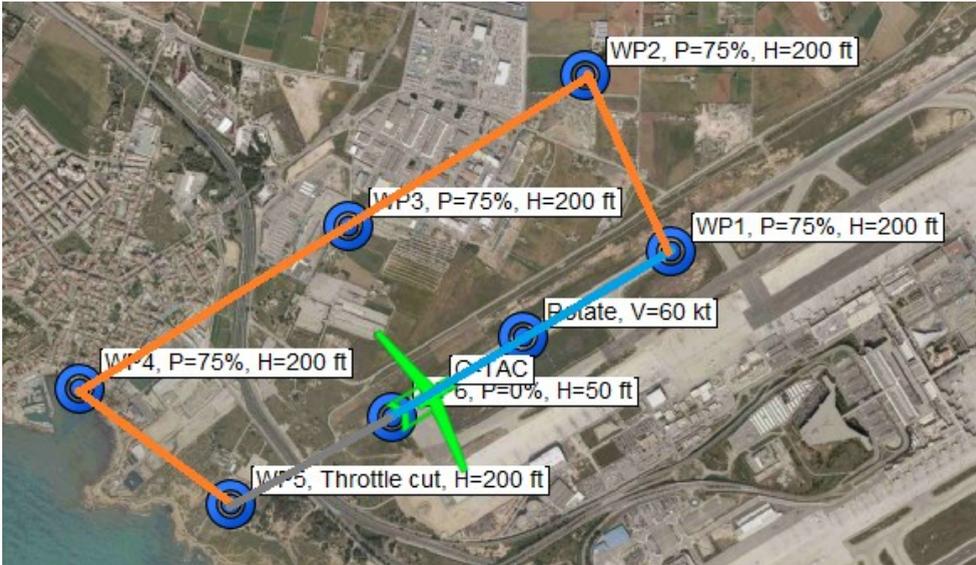


Fig. 9. Flight stages

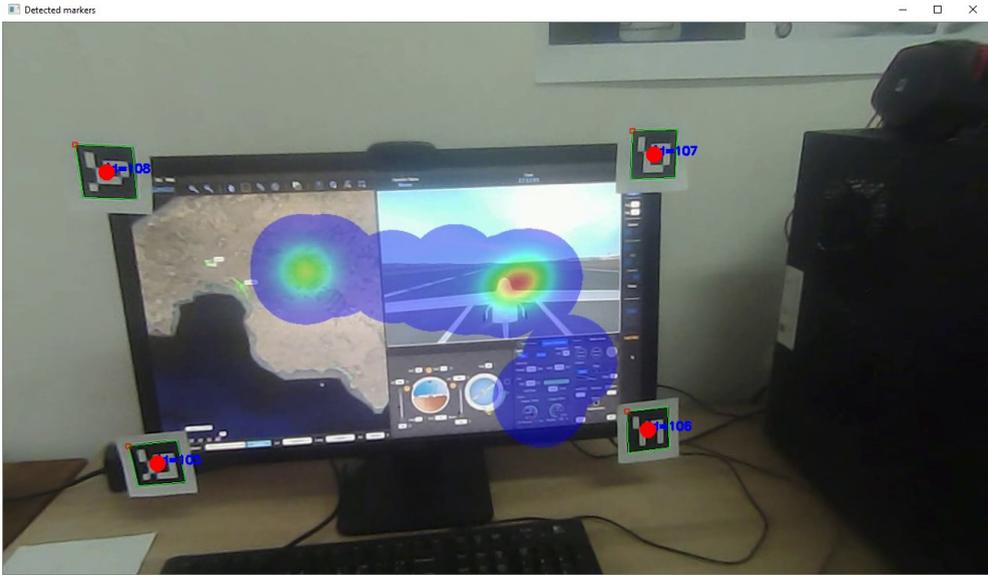
In Fig. 9 three flight stages are shown any flight is formally divided into: 1) from the start to the first turn; 2) up to the 4th turn; 3) to stop. In this case, the distribution of fixations is determined for each flight stage for each information source separately.

## Results

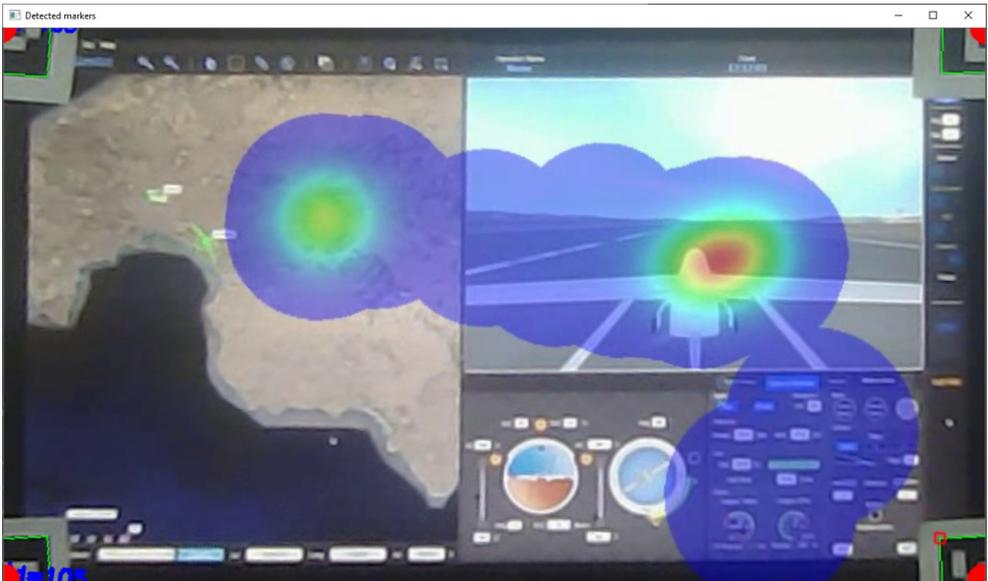
In Fig. 10, a frame taken by the environmental camera is shown.

Four fiducial Aruco markers are attached to corners of the monitor to define a region of interest and measure the relevant coordinates. OpenCV draws a green square, indicating that the method `cv::aruco::detectMarkers` succeeded in recognizing the current marker. Only then can the geometric center of each marker (red dots) be calculated. In addition, the barrel distortion is no longer observable because numerical compensation has been applied to each frame. Finally, a dynamic heatmap, which is built in advance, is superimposed on each frame.

In Fig. 11, the transformed (`cv::undistort` method) region of interest at exactly the same frame is shown. Evidently, that part of the image surrounding the region has been removed after the perspective transformation. The process resembles image cropping. In this way, spontaneous movements of the operator's head are eliminated. The mobile eye tracker works like a desktop based one.



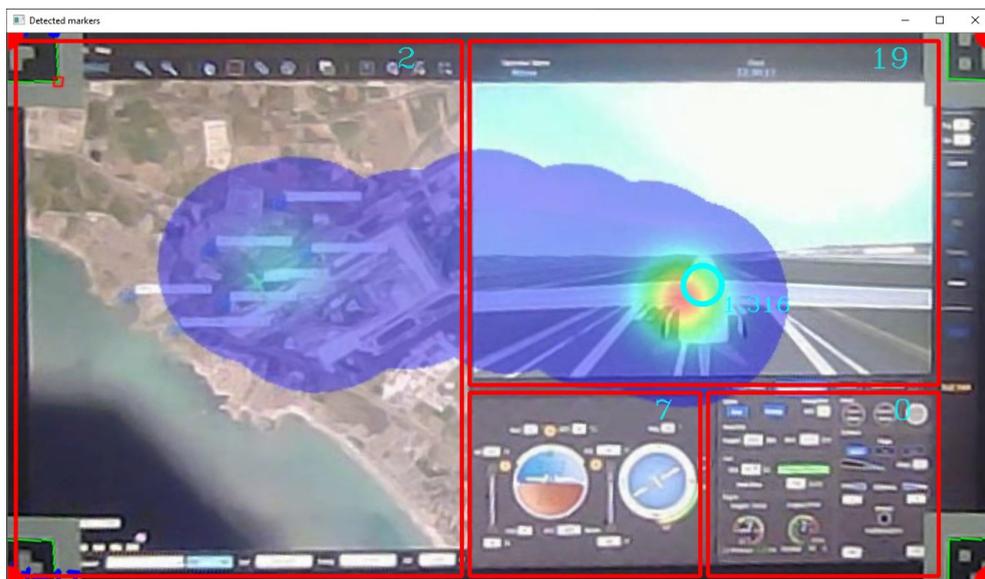
*Fig. 10. C-Star pilot station with recognized Aruco markers, imposed dynamic heatmap and compensated barrel distortion*



*Fig. 11. Transformed region of interest*

In Fig. 12, a detected fixation within the video channel is shown shortly after take-off. The fixation duration is 1 316 ms. Severe meteorological conditions are excluded in this very scenario, so are equipment failures. The flight task is the easiest possible, including stages take-off, flight along the approach pattern, landing approach and landing. The pilot has a lot of experience.

It is necessary for the algorithm to determine the number of fixations both within individual areas of interest, Fig. 8, and flight stages, Fig. 9. In Fig. 12, the current number of fixations is shown in the upper right corner of each information source. For the video channel, for example (second information source), the counted fixations from the start to the “Rotate” point are nineteen.



*Fig. 12. A fixation lasting 1316 ms found within the second information source*

In Table 2, the number of fixations and derived parameters for the three flight stages are shown without taking into account distribution by information sources, i.e. for full screen. The measurement was carried out by a Pupil Labs Core eye tracker on a C-Star simulator, SimLat.

*Table 2*

Stage / Parameter	Fixations number	Fixations per second, sec <sup>-1</sup>	Mean duration, ms	Standard error, ms
Stage 1, 41 sec	75	1.828020	543.600	68.391
Stage 2, 95 sec	214	2.245495	420.243	27.796
Stage 3, 38 sec	36	0.966699	1 030.441	216.606

The fixation durations for each flight stage are shown in Fig. 13.

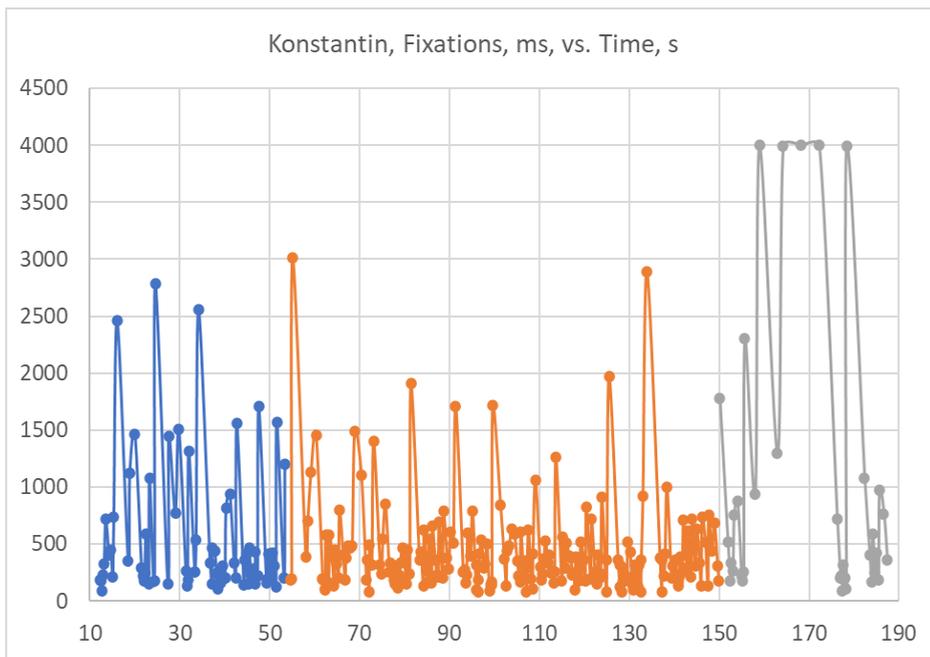


Fig. 13. Fixations distribution among the flight stages

In Table 3, distribution of the fixations is shown among both flight stages, Fig. 9 and information sources, Fig. 8. At this point, a check can be made for correctness of the obtained data. The total sum of fixations in each row of Table 2 must be equal to the number of fixations in column 2 of table 3 for the corresponding stage.

Table 3

Stage \ Information source	1	2	3	4
1 (up to 1 <sup>st</sup> turn), 00:12 – 00:53	10	43	20	2
2 (up to 4 <sup>th</sup> turn), 00:54 – 02:29	96	85	33	0
3 (up to stop), 02:30 – 03:08	2	30	4	0

#### 4. Discussion

Image distortion compensation seems to be quite laborious as compared to other stages of the presented study. As a whole, there are two types of image distortions: radial and tangential. Typically, the former distortion is predominant owing to lens geometric parameters. The latter distortion is due to the imperfect

lens alignment as parallel to the sensing element as possible. Although OpenCV provides readily available methods for compensating image distortion, analytical approaches toward rectifying image exist which are worth mentioning and recommended to further project development. These are Brown–Conrady model and the Division model, [8, 9]. In both models, it is required for the distortion center to be estimated in advance.

Low resolution of the transformed region of interest might be pointed out as a minor drawback. Obviously, the wider the lens' field of view, the worse the transformed image quality. However, for the transformation to be successful, it is necessary all four fiducial markers to be visible and discoverable within the image frame. It is easier to meet this certain requirement by means of a lens with shorter focal length.

The obtained results indicate that the proposed solution might be successfully applied to any kind of monitors and imaging screens. Unlike the desktop-based eye trackers, the mobile device functionality is not limited to the screen size. What is more, the displayed stimulus does not depend upon the window manager (DirectX, OpenGL) and screen resolution. Therefore, the stimulus may run in either of two modes: full screen or windowed. This seems to be in a more advantageous position over desktop-based eye trackers. Some of them fail to output a record if the stimulus, a game for instance, is displayed in full screen mode. What is more, the mobile eye tracker can be plugged in to an alternative device (PC, tablet, smartphone, etc.) to make the gathering data task less demanding.

According to obtained results, an experienced pilot switches their attention between information sources less. The reason is flight scenario simplicity. It is only sufficient for the pilot to look at the video channel straight so as to acquire necessary information and accomplish flight task. Furthermore, the Euler angles (pitch, roll) are limited by the C-star software enabling the pilot to perform basic flight maneuvering well. Almost all fixations are caught into the second information source, i.e. the video channel. In addition, as few as seven blinks were registered during entire flight.

Our experience in using the Pupil Labs mobile tracker led us to the following conclusions:

1) The presented complex of equipment will let us study the visual attention distribution of UAV operators during training at a flight simulator.

2) Studying peculiarities of UAV operators' visual attention distribution by means of eye-tracker will justify future creation of a science-based methodology for training students in drone control in different conditions and flight modes.

The data analysis derived from the registered fixations, with respect to the three conditionally separated flight stages, shows that:

- The fixation records are qualitative and admit processing and analysis.

- The flight conditional division allows for analysis of visual attention in terms of number and duration of the individual fixations.

- Fixations distributed on information sources are more difficult to register, especially on the instrument panel due to the specifics of the screen of the simulator itself.

A data interpretation derived from the registered fixations follows.

- Approximately identical depicted models of fixations throughout the flight is established for all test operators.

- The first stage – take-off, is characterized by an increased frequency of fixations at constant duration.

- Among all observed areas of interest, the operators focus their attention on the external environment more than the instruments.

- The second stage – cruise flight, is distinguished by a constant and slightly changing nature of the fixations for all operators.

- Landing features much more diverse fixations for individual operators, though, as long as all operators are concerned, an extended fixation time is observable as well as a decreased number of fixed informational sources.

From the obtained results and observation, the following conclusions can be derived, which, due to the limited number of experiments, cannot be presented as conclusions with a certain objectivity and reliability.

- The methodology at this stage for studying the cognitive functions of UAV operators by means of oculographic methods and developing an experimental flight model, including both visual control of the environment and certain parameters for maintaining flight on the available instruments, i.e. objective performance of the task might be considered developed and can be improved for the purposes of the study.

- There is an initial evidence that oculographic research reveals opportunities in studying effectiveness of the distribution of visual attention during training and improving the UAV operator skills.

- It is impressive from the fixations data that the landing stage appears most difficult to perform, i.e. efforts in the preparation and development of reflexes should be concentrated, contributing to fast and accurate perception and quality cognitive functions, which is a prerequisite for safe and effective operator work.

Source code developed to meet the presented research needs might be downloaded for free by following link [10].

## **Acknowledgements**

The current study has been funded by the Scientific Research Fund of the Republic of Bulgaria according to research contract No. KII-06/H27-10, 11<sup>th</sup> of December, 2018. The project title is “Human Factor in Remotely Controlled Aerial Systems – Analysis, Estimation, and Control.”

## References

1. Diaz Tula, A., A. T. N. Kurauchi, F. L. Coutinho, and C. H. Morimoto, Heatmap Explorer: An Interactive Gaze Data Visualization Tool for the Evaluation of Computer Interfaces, Proceedings of the 15<sup>th</sup> Brazilian Symposium on Human Factors in Computing Systems (IHC'16), October, 2016.  
<https://www.ime.usp.br/~kurauchi/publication/heatmap-explorer/>
2. Bradski, G., The OpenCV Library, Dr. Dobb's Journal of Software Tools, article id=2236121, 2000.  
<https://www.drdoobs.com/open-source/the-opencv-library/184404319>
3. Simlat, 2021. <https://www.simlat.com/>
4. Kassner, M., W. Patera, and A. Bulling, Pupil: An Open Source Platform for Pervasive Eye Tracking and Mobile Gaze-Based Interaction, Proceedings of the 2014 ACM international joint conference on pervasive and ubiquitous computing: Adjunct publication, 2014, 1151–60. New York: ACM. DOI:10.1145/2638728.2641695 arXiv:1405.0006. <https://arxiv.org/pdf/1405.0006.pdf>
5. Węglarczyk, S, Kernel Density Estimation and its Application, XLVIII Seminar of Applied Mathematics, ITM Web of Conferences. 2018, 23, 00037.
6. Romero-Ramirez, F. J., R. Muñoz-Salinas, R. Medina-Carnicer, Speeded up Detection of Squared Fiducial Markers, Image and Vision Computing, 2018, 76, 38–47.
7. Garrido-Jurado, S., R. Muñoz Salinas, F. J. Madrid-Cuevas, R. Medina-Carnicer, Generation of Fiducial Marker Dictionaries Using Mixed Integer Linear Programming, Pattern Recognition, 2016, 51, 481–91.
8. Calib3D, OpenCV. [https://docs.opencv.org/2.4/modules/calib3d/doc/camera\\_calibration\\_and\\_3d\\_reconstruction.html](https://docs.opencv.org/2.4/modules/calib3d/doc/camera_calibration_and_3d_reconstruction.html)
9. <https://www.imatest.com/support/docs/pre-5-2/geometric-calibration/distortion-models/>
10. <https://github.com/samolet4e>

## ИЗПОЛЗВАНЕ НА ОКУЛОГРАФ ЗА ИЗСЛЕДВАНЕ НА РАЗПРЕДЕЛЕНИЕ НА ВНИМАНИЕТО НА ОПЕРАТОРИ НА БЕЗПИЛОТНИ ЛЕТАТЕЛНИ АПАРАТИ

З. Хубенова, К. Методиев, С. Димитрова, Л. Алексиев

### Резюме

В настоящата статия се предлага подход за изследване на причини за разпределение на вниманието на оператор на безпилотен летателен апарат. По време на изследването операторът е тестван на полетен симулатор и данни са събирани и визуализирани с помощта на мобилен окулограф. Два етапа на работа са разгледани последователно: построяване на 2D трансформация на регион на интерес (хомография) в рамките на изображение и наслагване на динамична топлинна карта. По време на първия етап се елиминират спонтанни движения на главата на оператора, което позволява мобилният окулограф да се използва като настолен. По време на втория етап се изобразява разпределението на вниманието на оператора.

За да се реализира настоящето изследване бе разработен код на C++ с използване на методи, налични в библиотека OpenCV. Събраните данни от изследването са обработени и обсъдени подробно.