Automated Monitoring of Content Demand in Distance Learning

Viktor I. Shynkarenko, Valentyn V. Raznosilin and Yuliia Snihur

Dnipro National University of Railway Transport named after Academician V. Lazaryan, 2 Lazaryana str., Dnipro, 49010, Ukraine

Abstract

In this paper the research of means and the development of software for matching the student's gaze focus with the structure of information on the computer monitor during distance learning is presented. Widespread hardware is envisaged to be used. Primary processing of the face image, eye regions separation is performed by means of the OpenCV library. An appropriate algorithm to calculate the center of the eye's pupil has been developed. The influence of the system calibration process with different schemes of calibration point display, its delay time on the screen and location of the additional camera according the accuracy of the calculation the coordinates of the gaze focus is investigated. Based on the performed experiments, it was defined that the error of gaze focus recognition with using two cameras can be reduced to 4-10%. The proposed approach makes it possible for objective measurement the working time of each student with one or another part of content. The lecturer will have the opportunity to improve the content by highlighting significant parts that receive little attention and simplifying those elements that students process for an unreasonable amount of time. It is planned to integrate the developed software with the LMS Moodle in the future.

Keywords

Distance learning, educational content, program tools, oculography, gaze focus

1. Introduction

Distance learning is becoming more popular as higher education develops and modernizes. This is made possible by improved technical capabilities (computer public and private networks, digital knowledge bases, and so on) and the emergence of specialties that can be mastered remotely, without direct contact between the student and the lecturer. IT, finance, management, and other specialties are among them.

Aside from the obvious benefits of distance learning methods, there are some significant drawbacks:

- the learning process monitoring is absent or seriously difficult (attention, interest, understanding of the material by the student);

ICTERI'21: 17th International Conference on ICT in Education, Research and Industrial Applications, September 28 – October 02, 2021, Kherson, Ukraine

Shynkarenko.vi@gmail.com (V. I. Shynkarenko); valentin.raznosilin@gmail.com (V. V. Raznosilin); snigurjulia150498@gmail.com (Y. Snihur)

0000-0001-8738-7225 (V. I. Shynkarenko); 0000-0002-4463-4588 (V. V. Raznosilin); 0000-0001-7294-6821 (Y. Snihur)

- the possibility of an objective knowledge assessment has been complicated (person authentication, self-fulfillment of tasks, use of only permitted materials).

The first drawback is that modern students have difficulty concentrating when they master the material on their own. Clip thinking, caused by an abundance of unstructured information and its ease of access, has a negative impact.

The second relates to the evolution of communication tools, high-speed information retrieval, which can be used in the testing and examination process in the absence of a lecturer.

A method and corresponding software toolkit for tracking the gaze focus on the training material elements of the training course are presented in this article. A necessary requirement for development is the solution of the problem using the bare minimum of technical means to ensure mass application.

As a result, it is possible to objectively measure what the student pays more attention to, what is ignored, how much time is spent studying this or that section of the teaching material, and so on.

Based on the data obtained, it is planned to develop tools for analyzing the process of students' work with teaching materials using big data methods. In this case, both individual and group work patterns of various user groups can be considered (based on filtering by age, gender, level of initial training and other criteria). Finally, this will allow for a reasonable correction of training materials in order to improve the efficiency of their perception.

Assumptions are made as follows:

- a student looks at the content on a laptop screen with a resolution of 1366x768 pixels and physical dimensions of 340x200 mm;
- the built-in laptop video camera is located at the top of the screen and in the center horizontally, with the option of using an additional video camera;
- the distance from the screen to the bridge of the nose is 500 mm;
- the student sits almost motionless, centered on the screen;
- the average distance between the pupils is taken equal to 64 mm;
- there are no significant (more than 10-15 degrees) turns and tilts of the head.

Under the conditions described above, the user's face contour occupies about 50% of the frame's height and about 25-30% of the frame's width. It's allowing for fairly accurate recognition of the face contour as a whole and its individual parts (eye contours, pupils, temples, bridge of the nose, etc.).

2. Related works

To determine the focus point of the user's gaze on the monitor screen oculography or eye-tracking [1, 2] was used – a technology that allows you to observe and record eye movements: pupil dilation, its movement, etc.

Eye tracking is an effective research tool in various fields of education. Using this method, it is possible to analyze how the individual characteristics of students in combination with various properties of the teaching material can influence the learning process [3]. Eye focus tracking, in particular, enables to reveal [3]:

- student behavior features when selecting information and solving problems;
- differences in teaching strategies among different students;
- model of social interaction between lecturer and student;
- the effectiveness of training materials;
- what content elements attract and hold the student's attention.

Industrial gaze capture systems exist [4, 5]. Contact lenses with built-in mirrors, infrared illuminators that are reflected by the eyeballs and recorded by a video camera, and so on are examples. However, all of these methods necessitate complex and costly hardware and software, and they are inapplicable in the context of widespread use of distance learning.

The feasibility of developing a system for determining the gaze focus with a bare minimum of hardware (for example, a standard video camera on a modern laptop) was investigated [6, 7]. It should be noted that the required measurement accuracy is not achieved in these works, and there is no binding to the elements of the training content.

3. Research and project implementation

Computer vision is a set of software and hardware tools that read images in digital form and process them in real time. Various specialized software libraries are used in the developing computer vision systems [8, 9]. We chose OpenCV, which is a library of algorithms for computer vision [10, 11], image processing, and numerous general purpose open source algorithms. OpenCV is one of the most well-known libraries for dealing with computer vision issues. It is written in C++, but it can also be used in Python, Java, Ruby, Matlab, and Lua, and it is supported by a variety of platforms, including MS Windows, Linux, Mac OS, Android, and iOS. The library includes more than 500 functions. In particular, there are many optimized algorithms associated with the processing and analysis of video frames. Filtering, contour search, geometric transformations, motion analysis, object detection, observation, and many other functions have been implemented. It is possible to work with xml files.

The main stages of the gaze focus tracking process are highlighted:

1. Receiving video footage from a webcam that broadcasts a live image of a student's face;
2. Pre-processing – preparation of the image (translation into black and white, increasing contrast, etc.);
3. Facial feature recognition to detect the area of the eye on the video frame;
4. Eye recognition – tracking the contours of the eyes and the position of the pupils;
5. Calibration – comparison of the pupils position with the coordinates of a known point on the screen, which corresponds to the gaze focus;
6. Calculation of the gaze focus coordinates on the screen based on the dependences obtained in the previous stage.

The first three stages and partly the fourth are performed using the functions of the OpenCV library. Stages one and two are engineering and are not of scientific interest. Let us go over the third stage in greater depth.

Active appearance models (AAM), which are provided by the OpenCV computer vision library, were chosen to recognize facial features because they are designed to accurately locate anthropometric points on a facial image. AAM are statistical models of images that can be adjusted to the real image by various transformations. The toolkit, which is based on an AAM, is pre-trained on a set of pre-marked images.

There are two types of parameters used: shape-related (shape parameters) and statistical image model or texture-related parameters (appearance parameters).

The shape is defined as a set of landmark points on the face [12]. They describe certain facial features. Each landmark point defines a face morphological property and has its own number. Figure 1 shows a similar markup.

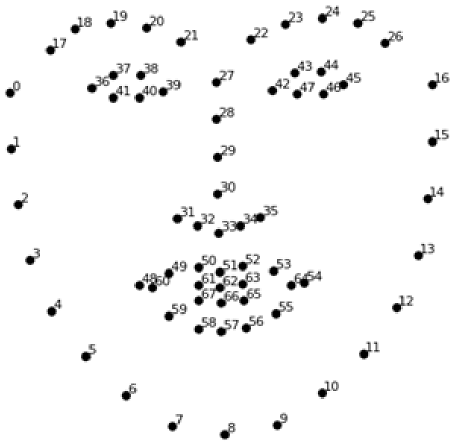

Figure 1: Landmark points of the human face recognized by the OpenCV library

The image in the presented example shows 68 landmark points that form the shape of the active appearance model. This shape represents the outer face contours, including the contours of the mouth, eyes, nose, and brows. This markup allows to determine the various parameters of the face in the image, which can be used for further processing.

For initializing the facial recognition method, the system uses a trained AAM that can be used to predict facial features. This model is presented in the form of a faces database [13]. This database includes variations of facial representation: different poses, lighting, etc.

3.1. Calculation of position of eye center and pupil

The fourth stage of the eye focus tracking system is the calculation of the coordinates of the centers of the eye and pupil. The position of the eye center depends on the tilt and / or rotation of the user's head. The position of the pupil center depends on the rotation of the eyeball and can change relative to the eye center, which will remain unchanged. At this stage, only the region of the video frame containing the image of the eyes is considered. Each eye is processed individually (fig. 2).

Let p1, .., p6 (points 36 −48 on Fig. 1) be the anthropometric points of the eye area. Each

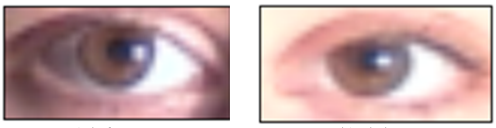

Figure 2: An example of video frame parts that are used to recognize the contour of the pupil (a – left eye; b – right eye)

point is given by a pair of x and y coordinates. Figure 3 and (1) demonstrate the calculation of the coordinates of the eye's center.

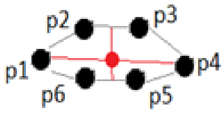

Figure 3: Eye scheme for calculating its center

The coordinates of the center of the eye (CoE) are calculated as following:

$$CoE_x = \frac{p1_x + p4_x}{2}, CoE_y = \frac{p2_y + p3_y + p5_y + p6_y}{4}, \quad (1)$$

where $p1_x, p4_x, p2_y, p3_y, p5_y, p6_y$ – the coordinates of the corresponding points.

One of the most difficult steps in tracking the gaze focus is determining the position of the eye pupil. The final result is significantly dependent on the values obtained at this stage, since the coordinate of the gaze focus can be calculated by analyzing the displacement of the pupil relative to the eye center. The process of tracking the eye pupil consists of several steps.

At the first step, the video frame is processed. The pupil is the part of the eye that is the darkest. To recognize the pupil, need to increase the intensity of the colors and then convert the frame to bi-tone.

To reduce the noise level median filtering is used. Depending on the aperture size (which are square, central) of the median filter, the various sizes and the positions of the pupil can be obtained (fig. 4).

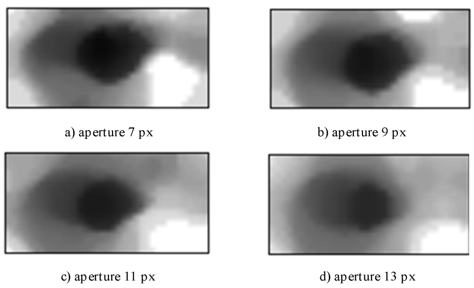

Figure 4: The impact of aperture size on the final image

A threshold algorithm is used to highlight the darkest areas and change the image color model from grayscale to bi-tone [14, 15].

The fig. 5 shows the dependence of the result from the threshold value. The successful application of the threshold algorithm in this example is the use of T = 40, since the binary image clearly shows the area of the eye pupil.

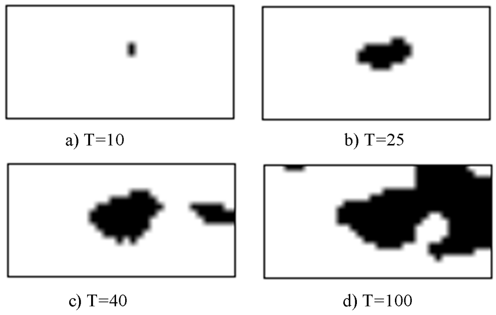

Figure 5: Dependence of the algorithm result from the selected threshold value (T)

The next step of eye pupil tracking is to find the contours of the iris on the binary image. Contour – a curve that connects all points (along the border), that have the same color. The Satoshi Suzuki algorithm [16], provided by the OpenCV computer vision library, is used to recognize contours on a binary image. If there are several contours, the largest one is selected because it corresponds to the iris of the eye. In fig. 6, examples of recognizing the elements contours are shown.

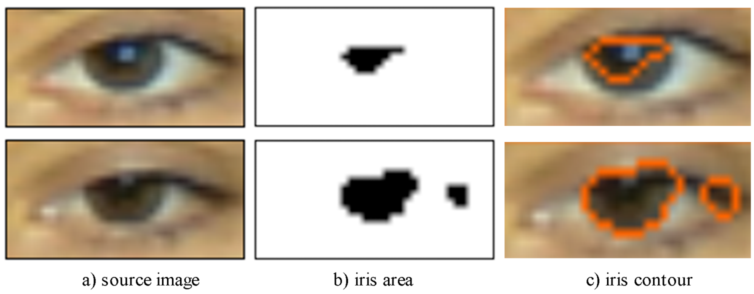

Figure 6: Examples of iris contours recognition

The iris has the shape of a circle. To calculate the position of the eye pupil, need to find a circle that fits and coincides with the contour of the iris obtained at the previous step. The OpenCV library's algorithm "The smallest enclosing disk" [17] is used for this purpose. Figure 7 shows examples of the iris circle and the eye pupil that corresponds to them.

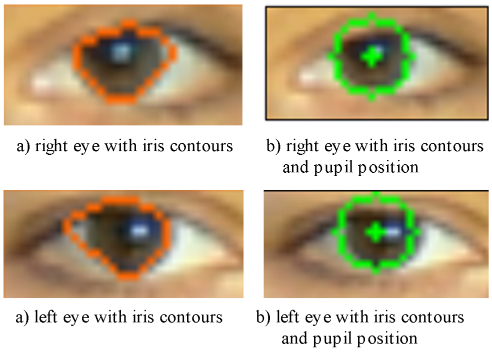

Figure 7: Example of finding the eye pupil position

3.2. Calibration of the gaze focus coordinate on the monitor with the pupil position

In the fifth stage, correspondence is established between the position of the pupil and a known point on the monitor. This requires calibration. Calibration is a sequential demonstration on the monitor screen of a point with known coordinates with simultaneous registration of the user's gaze directed at it. The movement of the calibration point on the monitor is depicted in fig. 8.

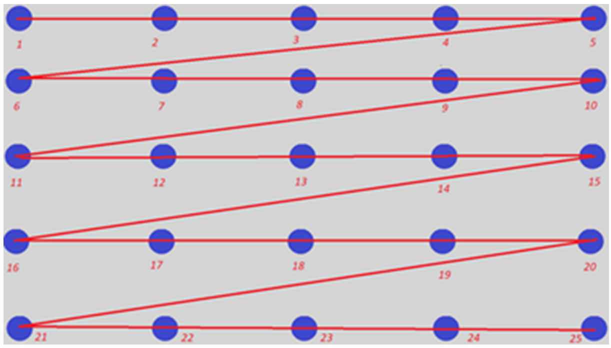

Figure 8: The movement of the calibration point

The program changes the position of the point after a certain period of time and calculates the corresponding eye pupils coordinates. During calibrating, the student must keep a close eye on the point on the monitor. To get rid of incorrect data obtained during blinking, the system filters it out. The relationship between the width and height of the open eye is used to achieve this. According to the study [18], the following is can be used to calculate the ratio of the sides of the eye:

$$EAR = \frac{|p1_y - p6_y| + |p3_y - p5_y|}{2\,|p1_y - p4_y|}, \tag{2}$$

where EAR – the ratio of the eye sides.

The ratio of the eye sides is almost constant as long as the eye is open and close to zero when the person is blinking. Figure 9 depicts eye's sides ratio graph. The EAR threshold is set at 0.15. If the EAR is less than or equal to the threshold value, the system detects blinking and ignores the corresponding frames.

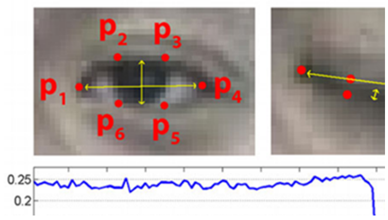

Figure 9: Dependence of EAR on the state of the eye

The result of the calibration is an eye pupils coordinates array corresponds to the positions of the calibration point. The next step is to process the obtained data – to find the linear regression dependence of the pupil coordinates on the calibration point position on the screen:

$$L = \alpha \cdot C + \beta, \tag{3}$$

where $L = CoE - CoI, CoE, CoI$ – coordinates of the eye and pupil centers, C – is the coordinate of the calibration point position, α and β – are unknown coefficients. The linear dependence is calculated separately for the horizontal and vertical axes (x and y coordinates, respectively). The least squares method is used to calculate the parameters of the linear regression:

$$\alpha = \frac{1}{N}\sum_{i=1}^{N} c_i - \frac{\beta}{N}\sum_{i=1}^{N} l_i, \qquad \beta = \frac{N\sum_{i=1}^{N} l_i c_i - \sum_{i=1}^{N} l_i \cdot \sum_{i=1}^{N} c_i}{N\sum_{i=1}^{N} l_i^2 - (\sum_{i=1}^{N} l_i)^2}, \tag{4}$$

where N – data sample size (y_i, x_i).

3.3. Eye focus calculation

Calculations are performed for each frame of the video stream except for frames where are blinking detected. The coordinates of CoE, CoI and their difference L for the left and right eye on both axes are determined ($L_{left,X}, L_{left,Y}, L_{right,X}$ and $L_{right,Y}$). The average (for two eyes) displacement value is calculated:

$$L_averX = \frac{L_{left,X} + L_{right,X}}{2}, L_AverY = \frac{L_{left,Y} + L_{right,Y}}{2}, \tag{5}$$

where L_AverX – the average displacement along the x axis, L_AverY – the average displacement along the y axis.

The coordinates of the gaze focus are calculated as next:

$$PoG_X = \frac{L_averX - \alpha_X}{\beta_X}, PoG_Y = \frac{L_averY - \alpha_Y}{\beta_Y}, \tag{6}$$

where α_X, β_X – linear regression coefficients obtained at the calibration stage by the x axis and α_Y, β_Y – by the y axis using (4). If the obtained gaze focus coordinates exceed the monitor boundaries, they are given the nominal value of the monitor coordinate limits. When using an additional camera, calibration is simultaneously performed separately for each camera and dependencies (3) and (4) are determined. When processing a video image, the position of the gaze focus (6) is calculated separately with subsequent averaging.

3.4. Matching eye focus with a content

A method for matching the user's gaze focus with the information on the monitor is proposed. Educational content (such as a tutorial chapter etc) is converted to a fixed-width bitmap. The height of the image depends on the size of the content. The resulting image is divided into regions that describe the structure of the content.

The student can view the content by scrolling the image up or down. For each video frame, the program calculates the gaze focus point. The computation frequency depends on the parameters of the video stream. The gaze focus is averaged over a specified time interval (for example, within 1-2 seconds).

Figure 10 depicts an example of content splitting into structural elements.

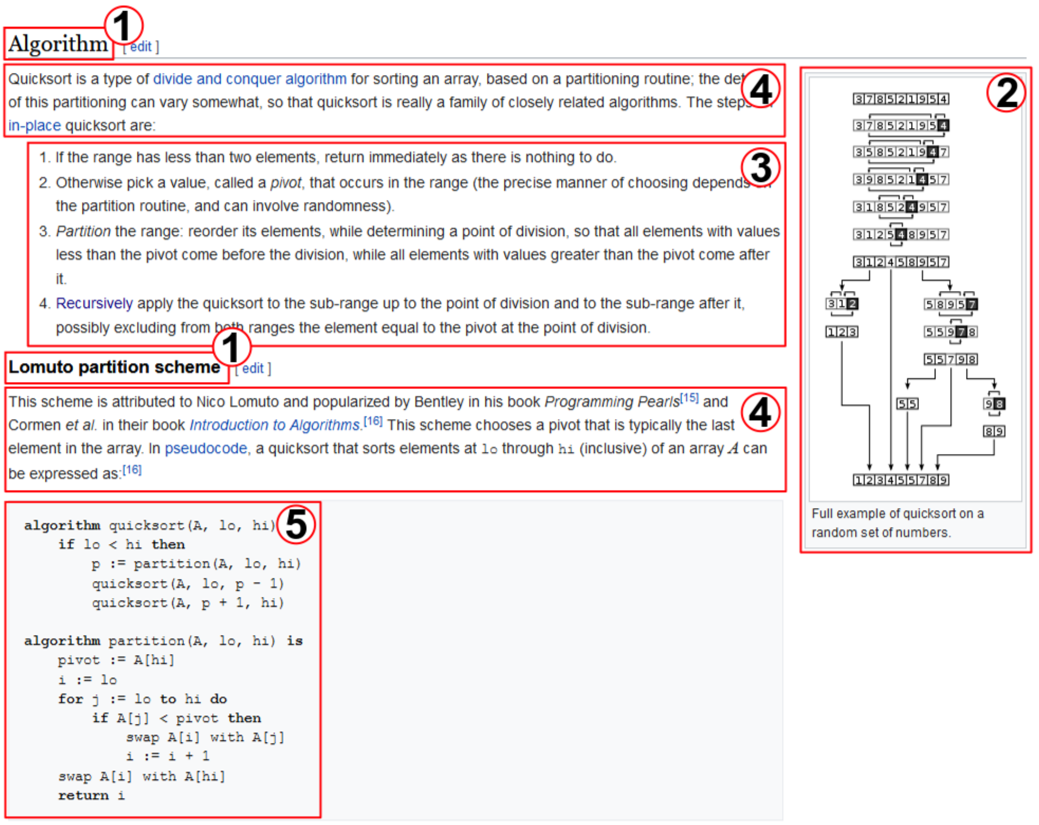

Figure 10: An example of splitting content into separate elements

In figure 10 circles are used to highlight various types of elements:

1. title of the article;
2. drawing;
3. list;
4. a paragraph of the text;
5. software source code.

Next, a search is performed for the content element on which the gaze is focused. The corresponding item identifier saves into the list. The resulting data can be used to visualize how users interact with the content. For example, the program can reconstruct the sequence of the user's gaze movement; highlight content elements on which the gaze was focused for the minimum or maximum time, and so on. In addition, it can calculate various statistical parameters that characterize the users experience with this content.

The structure of the content is described in the form of a JSON file, which has the following format:

```
content_structure ::=
{
  "source": {
   "title": "<content title>",
   "type" : "< book | monograph | article | manual >",
   "media": "< img | pdf | docx | web >",
   "link" : "<web link>",
   "image": "<image filename>" },

  "content": {
   "heading" : <items array>,
   "text"    : <items array>,
   "picture" : <items array>,
   "formula" : <items array>,
   "code"    : <items array>,
   "list"    : <list-items array> }
}
<items array> ::=
 [
  { "id":<int>, "pos" : [<top>,<left>,<W>,<H>], "info":"<descr>"},
  ...
  { "id":<int>, "pos" : [<top>,<left>,<W>,<H>], "info":"<descr>"}
 ]
<list-items array> ::=
 [
  { "id":<int>, "pos" : [<top>,<left>,<W>,<H>], "info":"<descr>",
    "items" : [<vert_offset_0>,..,<vert_offset_N>]},
```

```
  ...
  { "id":<int>, "pos" : [<top>,<left>,<W>,<H>], "info":"<descr>",
    "items" : [<vert_offset_0>,..,<vert_offset_N>]}
 ]
```

The description of an individual element consists of information about the frame enclosing it: coordinates of the upper left corner, height and width. In addition, a free-form text label is provided for each element. An example, for a drawing, the label can contain the number and title of the drawing, for a paragraph, it can be empty or briefly describe its content, etc.

Additional information is provided to identify the position (vertical) of the list's individual items when describing an element in the form of a list (for example, references).

4. Results and discussion

Several experiments were performed to assess the impact of the number of points and the delay in their demonstration during calibration, an additional camera, and slight head movements on the accuracy of the results.

In the performed experiments:

1. uses a laptop with a screen resolution of 1366x768, a built-in camcorder with a resolution of 1280x720. Experimental conditions: the timer of calibration point position change is equal to 15 sec; changing the position of the calibration point is crosswise in the center of the monitor, horizontally 9 points, vertically 5; the static position of the head;
2. unlike the previous one, an additional external video camera with a resolution of 1280x1024 is used, which is located at the bottom and in the middle of the monitor screen;
3. equipment is as second experiment. Experimental conditions: the timer of calibration point position change is equal to 10 sec; changing the position of the calibration point has the sawtoothed form (like a Fig. 6) horizontally 3 points, vertically 3; the static position of the head;
4. equipment is as in the 2nd and the 3rd experiment. Experimental conditions: the timer of calibration point position change is equal to 6 sec; changing the position of the calibration point has sawtoothed form, horizontally 6 points, vertically 6; the static position of the head;
5. equipment and conditions are as in the experiment 4, except for changing the position of the head (rotations up to 10-15 degrees) when calibrating and calculating the gaze focus;
6. equipment and conditions are as in the experiment 4, except for changing the position of the additional camera, which is located on the top and in the middle of the monitor, screen.

Table 1 displays the results of the software's accuracy under different conditions of use.

The results of the experiments demonstrated that using two cameras produce the best results for tracking the gaze focus along the y-axis. It was also found that adjusting the calibration process has a significant impact on the final result. The advantages of using two cameras are clearly demonstrated in experiment 2. The error of the results for each camera is more than if these two results are generalized (Table 2).

Table 1
Errors of gaze focus tracking in performed experiments

№	Error by the horizontal axis (px)	Error by the horizontal axis (%)	Error by the vertical axis (px)	Error by the vertical axis (%)
1	51	3,8	214	27,9
2	58	4,2	71	9,3
3	84	6,2	94	12,3
4	80	5,8	73	9
5	223	16	127	16
6	193	14	109	14

Table 2
Deviation of the calculated gaze focus coordinates from the coordinates of the calibration point

№	Error by the horizontal axis (px)	Error by the horizontal axis (%)	Error by the vertical axis (px)	Error by the vertical axis (%)
Camera (internal)	90	6,6	90	11,7
Camera (external)	67	4,9	87	11,4
For both cameras	58	4,2	71	9,3

As an example, the graphs of the calibration point coordinates on the screen (PoS) and the calculated coordinates of the gaze focus (PoG) when tracking the user's gaze using both cameras (experiment 2) by the x-axis are presented in fig. 11 and fig. 12.

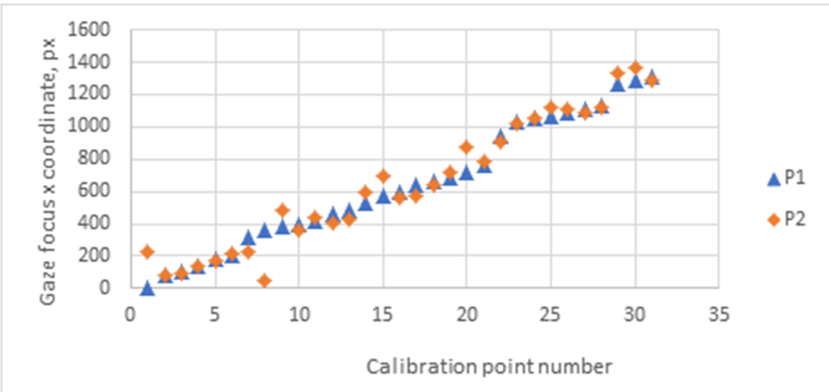

Figure 11: Coordinates of the calibration point and gaze focus

When the results of experiments where the user holds the head in a static or dynamic state were compared, it was found that tracking of the gaze focus is better if the head is held in a static state. It because the tracking system better recognizes pupil displacement under these conditions.

Existing approaches are based on direct observation of the trainee or use costly equipment. This presupposes conducting separate experiments to assess attention on structural elements of

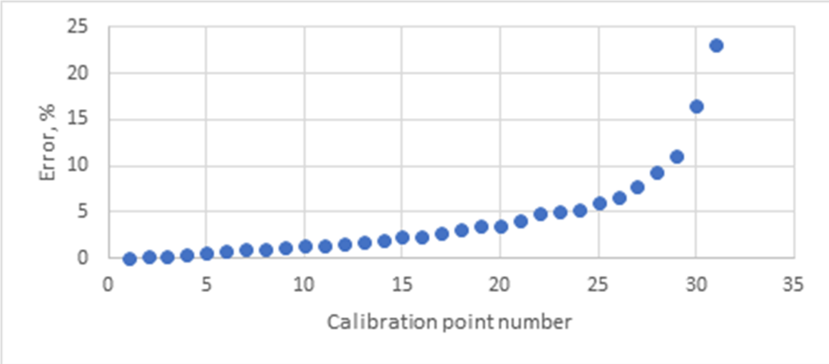

Figure 12: The relative error of calculation the gaze focus coordinates

the studied material in each specific case.

The method presented in this work involves a massive study of the behavioral reactions of students who are either in the classroom or at home and have standard devices like a laptop. Unlike other works, the goal here was not to achieve pixel (or close to it) accuracy in determining the gaze focus. For the purposes specified, an accuracy of 50-100 pixels along each axis of the monitor is adequate. The specified accuracy corresponds to the size of the content (paragraph, drawing, formula, etc.) with which the student interacts. Furthermore, an additional USB video camera can be used, which on the one hand significantly improves accuracy (especially along the vertical axis) while not reduce the target audience too much because such cameras are widely available and relatively inexpensive.

5. Conclusions

The approach to the development of tools for tracking the gaze focus compliance with the structure of information on a computer monitor is studied in this paper. Using hardware that is available to everyone to improve the assimilation of information by students during distance learning proposed.

Implemented software environment that is able to investigate the correspondence of the gaze focus with the structure of information on the monitor using one or two cameras.

Several experiments were carried out under various conditions, including calibration process settings, changing the position of the additional video camera, and the student's head state during the experiment. Based on the results of the experiments, it was concluded that the error of gaze focus recognition is up to 10

The widely used in Ukrainian universities LMS Moodle distance learning system lacks tools for assessing the significance, and intelligibility of teaching materials and their elements. Comprehensibility is assessed indirectly through student testing.

The proposed method allows for the objective measurement of each student's work time with one or another element of content. The lecturer has the opportunity to highlight significant parts that receive little attention and to simplify elements that students process for an unreasonable amount of time.

This work was carried out in continuation of the research project of the tracking and analysis of learning processes, in particular, to programming [19, 20, 21].

It is planned to integrate the developed software with the LMS Moodle in the future, as well as to refine the algorithm for identifying the coordinates of the pupil and flare on the eyeball in order to improve the accuracy of recognizing the gaze focus along the vertical axis in different lighting and other conditions (for example, the presence of glasses, overlapping hairstyles).

References

[1] L. R. Young, D. Sheena, Survey of eye movement recording methods, Behavior research methods & instrumentation 7 (1975) 397–429.

[2] F. Wadehn, T. Weber, D. J. Mack, T. Heldt, H.-A. Loeliger, Model-based separation, detection, and classification of eye movements, IEEE Transactions on Biomedical Engineering 67 (2019) 588–600. doi:10.1109/TBME.2019.2918986.

[3] Understanding different aspects of learning, 2020. URL: https://www.tobiipro.com/applications/scientific-research/education/, accessed 07 September 2020.

[4] S. Y. Gwon, C. W. Cho, H. C. Lee, W. O. Lee, K. R. Park, Robust eye and pupil detection method for gaze tracking, International Journal of Advanced Robotic Systems 10 (2013) 98. doi:10.5772/55520.

[5] C. W. Cho, J. W. Lee, K. Y. Shin, E. C. Lee, K. R. Park, H. Lee, J. Cha, Gaze detection by wearable eye-tracking and nir led-based head-tracking device based on svr, Etri Journal 34 (2012) 542–552. doi:10.4218/etrij.12.0111.0193.

[6] E. Skodras, V. G. Kanas, N. Fakotakis, On visual gaze tracking based on a single low cost camera, Signal Processing: Image Communication 36 (2015) 29–42. doi:10.1016/j.image.2015.05.007.

[7] O. Ferhat, F. Vilariño, Low cost eye tracking: The current panorama, Computational intelligence and neuroscience 2016 (2016). doi:10.1016/j.image.2015.05.007.

[8] Y. V. Bulatnikov, A. A. Goeva, Sravnenie bibliotek kompyuternogo zreniya dlya primeneniya v prilozhenii, ispolzuyushchem tekhnologiyu raspoznavaniya ploskikh izobrazheniy (comparison of computer vision libraries for use in an application using flat image recognition technology), Vestnik Moskovskogo gosudarstvennogo universiteta pechati (2015) 85–91.

[9] G. Shakhin, Sravnitelnyy analiz bibliotek kompyuternogo zreniya (comparative analysis of computer vision libraries), in: Colloquium-journal, 24 (48), 2019, pp. 53–55. doi:10.24411/2520-6990-2019-10812.

[10] Y. Ji, S. Wang, Y. Lu, J. Wei, Y. Zhao, Eye and mouth state detection algorithm based on contour feature extraction, Journal of Electronic Imaging 27 (2018) 051205 1–8. doi:10.1117/1.JEI.27.5.051205.

[11] D. Chandrappa, G. Akshay, M. Ravishankar, Face detection using a boosted cascade of features using opencv, in: International Conference on Information Processing, Springer, 2012, pp. 399–404. doi:10.1007/978-3-642-31686-9_46.

[12] Facial point annotations, 2020. URL: https://ibug.doc.ic.ac.uk/resources/facial-point-annotations/, accessed 07 September 2020.

[13] Shape predictor 68 face landmarks, 2020. URL: https://github.com/davisking/dlib-models/blob/master/shape_predictor_68_face_landmarks.dat.bz2, accessed 07 September 2020.
[14] Tracking your eyes with python, 2020. URL: https://medium.com/@stepanfilonov/tracking-your-eyes-with-python-3952e66194a6, accessed 07 September 2020.
[15] Obrobka rastrovykh zobrazhen (raster image processing), 2020. URL: https://www.tobiipro.com/applications/scientific-research/education/, accessed 07 September 2020.
[16] S. Suzuki, et al., Topological structural analysis of digitized binary images by border following, Computer vision, graphics, and image processing 30 (1985) 32–46.
[17] E. Welzl, Smallest enclosing disks (balls and ellipsoids, in: New results and new trends in computer science, Springer, 1991, pp. 359–370.
[18] J. Cech, T. Soukupova, Real-time eye blink detection using facial landmarks, Cent. Mach. Perception, Dep. Cybern. Fac. Electr. Eng. Czech Tech. Univ. Prague (2016) 1–8.
[19] V. Shynkarenko, O. Zhevago, Visualization of program development process, in: 2019 IEEE 14th International Conference on Computer Sciences and Information Technologies (CSIT), volume 2, IEEE, 2019, pp. 142–145. doi:10.1109/STC-CSIT.2019.8929774.
[20] V. Shynkarenko, O. Zhevago, Development of a toolkit for analyzing software debugging processes using the constructive approach, Eastern-European Journal of Enterprise Technologies 5/2 (2020) 29–38. doi:10.15587/1729-4061.2020.215090.
[21] V. Shynkarenko, O. Zhevaho, Constructive modeling of the software development process for modern code review, in: 2020 IEEE 15th International Conference on Computer Sciences and Information Technologies (CSIT), volume 1, IEEE, 2020, pp. 392–395. doi:10.1109/CSIT49958.2020.9322002.