Towards a Head Movement-based system for multi-layer digital content exploration

Alessandro Bruno[1*]
abruno@brounemouth.ac.uk

Morgan Moore[1*]
27morganmoore@gmail.com

Jinglu Zhang[1]
zhangj@bournemouth.ac.uk

Stéphane Lancette[1]
slancette@bournemouth.ac.uk

Ville P. Ward[2]
peter.ward@shopparapp.com

Jian Chang[1]
jchang@bournemouth.ac.uk

[1]National Centre for Computer Animation. Bournemouth University. Poole. UK
[2]Shoppar Ltd, Plexal, 14 East Bay Lane, Stratford, London. E20 3BS. UK

Abstract

In this paper, we propose a novel technique based on Head Movement tracking to explore multi-layer digital content. We extend an existing method by Kazemi et al. dealing with the extraction of facial landmarks to define the 'head-gaze' of the user. We use the 'head-gaze' to calculate the users' on-screen coordinates. Hovering the cursor over an interactive area for a given time threshold allows users to explore the next layer contents. Our experimental sessions allowed us to measure the technique's level of control and usability. Our results were promising, and users were able to interact with considerably small regions. Furthermore, our lightweight method can be used with a low-cost camera or webcam and a wide range of screen sizes and distances.

Keywords: Head Movements, Digital Content Exploration, Interaction.

1 Introduction

Over the last decades, there has been growing interest in touchless interfaces. A wide variety of fields such as healthcare, customer retail, and gaming, demand interfaces to be accessible, easily ported over different platforms and user friendly. The current COVID-19 pandemic presents many challenges to the world of digital media and some significant changes will need to be implemented. Public touch-based interfaces are now under major scrutiny and this offers an opportunity to consider new innovative ways of interacting with visual content. A growing number of public interfaces are now required to be touchless in order to prevent the spread of disease on surfaces. Touchless techniques such as head tracking and eye-tracking are also becoming increasingly important in improving user control in fields such as gaming and assistive technology [1]. Several eye-tracking and head movement-based approaches allow users to trigger various commands without using hands or changing gaze direction [2], [3]. As highlighted by Ju et al. [4] eye-tracking can be a reliable tool for human-computer interaction. Many of the most accurate techniques use thermal infrared sensors and require the user to complete

[*]Alessandro Bruno and Morgan Moore should be considered joint first authors. Alessandro Bruno has coordinated the research activities and paper writing. Morgan Moore has coded the main algorithm and implemented the experiment.

a calibration step. Other eye-tracking solutions [5] base their functionality on wearable equipment (eye-glasses). While these techniques are highly effective their hardware and calibration requirements may limit their accessibility to the general public.

Recent scientific findings from the neuroscience community[6] reveal that the vestibular system in mammals broadcasts head pose signals to areas throughout the brain and up to the visual cortex to be processed. This indicates the significance of head pose in visual attention analysis.

In our work, we modify and extend an existing architecture focused on the optimisation of face alignment [7]. Our technique can provide a low cost, lightweight and user friendly method of 'Headgaze' capture. The program can accurately detect facial landmarks up to 2.5 meters making it a reliable solution for various environments and screen sizes. Before using the program, it can be configured to the user's camera specifications, camera location and screen size. Using both 2D and 3D facial key points we define the head gaze and head position of the user. We then use these values to calculate their on-screen coordinate position. We adopt a multilayer digital content architecture that can be navigated through using buttons/icons (see figure 1). The remainder of the paper includes the following sections: Related Techniques; Proposed Method of Headgaze; Experimental Results; Conclusions and Future Works.

2 Related Techniques

The fast development of touchless HCI (Human Computer Interaction) techniques [8] such as eye-tracking, hand gesture recognition, time of flight sensors, head pose detection, etc. provide users with a more natural and intuitive way of communicating with machines, which largely improve the user experience. Eye-tracking plays an important role in HCI applications due to the fact that eye movements and the point of gaze can provide valuable insights about user preferences [9]. Some systems take advantage of the electrooculography (EOG) [10] to measure electrical potential differences between the front and back of the human eye. In [11], authors combine the top-down underlying eye dynamics with the bottom-up gaze measurements from a static gaze estimation network to improve eye gaze predictions. Hotrakool et al. [12] provide a real-time eye-tracking solution based on gradient orientation pattern matching and automatic template update. Some eye-tracking methods are the mixture of different technologies. For example, in [13], Coetzer and Hancke use an Infrared (IR) camera and IR LEDs to capture the bright and dark pupil images. As the result, substracted images from different illumination conditions are feed into support vector machine (SVM) and addaptive boosting (AdaBoost) classification network. However, most of the eye-tracking techniques either require additional sensors or a calibration process, which are cost and computationally inefficient. Instead, our proposed head pose tracking can be regarded as a low-cost and efficient substitution of eye-tracking, because it relies only on a single webcam.

From another perspective, hand posture and gesture recognition have observed an increasing number of studies in HCI. Hand gesture recognition focuses on the combination of static and dynamic palm and finger position and shape. The static hand gesture refers to the stable hand shape, while the dynamic hand gesture is the sequence of hand movements such as waving hand [14]. It is a highly feasible nonverbal communication strategy because the hand is the most effective general-purpose interaction tool [15]. Pisharady et al. [16] implement human visual attention by extracting high level (shape, shading) and low level (colour) features to recognize hand postures against a complex background. In [17], the authors provide a two-stage Kinect based human posture recognition system for sign language recognition. Colour and depth information are processed to detect and track the hand. Then they apply a convolutional neural network (CNN) to identify features from the hand gesture images automatically. Nevertheless, training a DNN (Deep Neural Network) requires a huge amount of data and is highly reliant on powerful hardware.

Head pose estimation refers to capturing small movements of a person's head to infer its orientation. As well as determining the head orientation head pose can also offer valuable in-

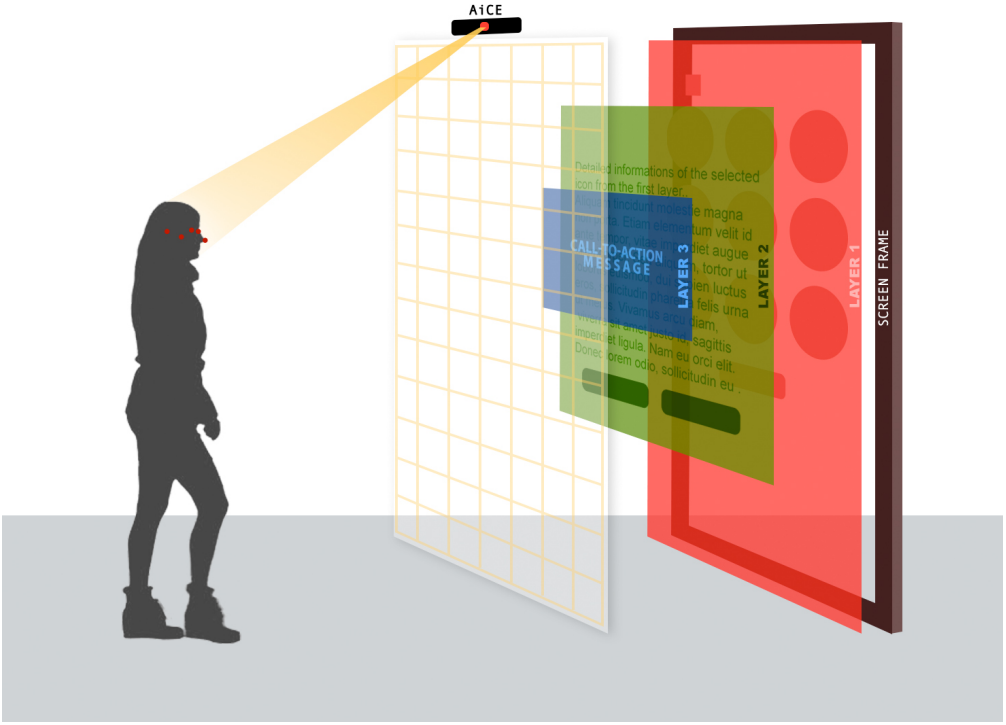

Figure 1: The overall graphical description of our architecture for multilayer digital content exploration is given above. A camera device installed on the top of the screen records everything happening in its field of view. Our proposed method processes each frame out of the camera, detects head movements, and projects them onto a screen as a cursor overlaid with a not visible regular grid (yellow grid). Each layer (the blue, the red and the green one) has got its interactive regions (buttons, icons). As soon as users spend a certain amount of time interacting with a region of interest on the current layout, they are shown a more detailed layer with contents related to the same area which they were interacting.

formation about the environment and its context; for instance, a quick head movement might be a sign of alarm [18]. Head pose estimation techniques can be roughly divided into two categories: wearable sensors and computer vision techniques. In our case, we only deal with touchless human-computer interaction, so we only focus on computer vision techniques such as pattern recognition and image processing. Ruiz et al. [19] propose a fine-grained head pose Euler angles prediction method with a multi loss network trained on a large synthetic dataset. In [20], the authors use a CNN with adaptive gradient methods to estimate the head pose in the wild. In addition, some researches combine the eye-tracking and head movement techniques. Yingbo et al. [21] propose a hybrid eye-tracking (g a modified version of the open source ITU Gaze Tracker) and head movement (Microsoft Kinect depth sensor) technique to reconstruct the realist eye movement avatars. Again, most of the head pose prediction methods suffer from high computational software and hardware requirements. Although our solution is based on traditional computer vision technology, the results are promising, and it is lightweight, thus can be easily integrated into edge devices.

3 Proposed Method of Headgaze

The technique we describe in this section is an extension of an off-the-shelf solution that dealt with predicting facial landmarks from inputted images/videos [7]. We propose a method of using these facial landmarks to define the user's head gaze and to interact with visual content. The graphical description of the

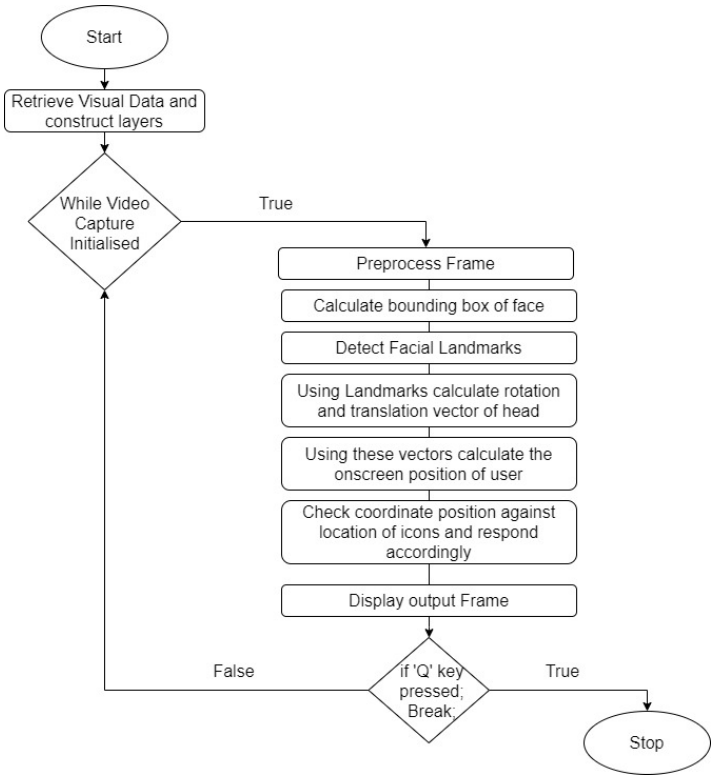

Figure 2: Above we provide a flowchart of the main steps of the algorithm behind Headgaze proposed method

algorithm behind the proposed method is given in figure 2. In order to achieve a high level of accuracy in 'Head Gaze' detection, proper camera calibration is required. This includes calculating the camera matrix and distortion coefficients. Smoothing filters [22] were also used to smooth results and to improve the users experience. The program begins by retrieving the visual content. This content is then used to create the necessary image layers for the program. For layer 1, each icon is cropped into the background image. To create the layer 2 images, each icon's corresponding layer 2 content is cropped into copies of the completed layer 1 image. The main event loop of the program is then entered, which is called for every frame. The frame firstly goes through some pre-processing steps where it is flipped and converted to grayscale. In the next stage the bounding box of the face is calculated using a frontal face detector function from the Dlib library [23]. This information can then be used to identify the precise landmarks of the face using a shape predictor model [7]. This model is trained over i-bug 300-W datasets [24] [25] [26] to identify the facial landmarks of each frame. The detected 2D landmarks, the corresponding 3D co-ordinates, camera matrix and distortion coefficients are then used as inputs to the opencv [27] function which returns the rotation vector and translation vector of the head.

The rotation and translation vectors can then be used to determine the 'cursor' location. The rotation vector x-value represents the pitch of the head, and the y-value represents the yaw of the head. Using these two values, the screen size, and the distance relative to the camera,the screen position can be calculated and converted to screen coordinates (see equation 1 and figure 3).

$$\begin{aligned} sxp &= t_z * tan(-r_y) - t_x \\ syp &= t_z * tan(r_x) - t_y \\ x_c &= (sxp)/(sw/2) * w/2 + w/2 \\ y_c &= (syp)/(sh/2) * h/2 + h/2 \end{aligned} \tag{1}$$

Where x_c and y_c are the screen coordinates

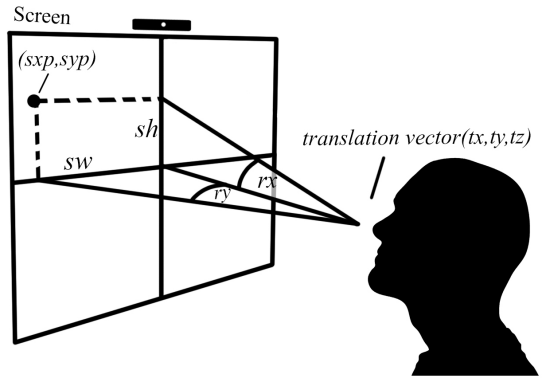

Figure 3: The diagram above shows the current scenario with a screen sized sh x sw

of the cursor, h and w are the height and the width of the screen in pixels, sh and sw are the height and the width of the screen in cm, and r_x is the pitch and r_y is the yaw of the user's head. sxp and syp represent the on-screen position in cm and t_x t_y t_z represent the distance relative to the camera. The values of sxp and syp must be adjusted based on the camera position in order to correspond with the centre of the screen. The cursor is then drawn to the screen and its location is checked against each icon/button. If there is an overlap, the loading animation for the given icon is triggered and continues while the cursor remains within the icon's bounding box. Once the predefined time limit is met, action is initialised e.g change layer. Choices of interactive regions from viewers are considered simple tasks by neurocognitive studies [28].

Neuroscience maps them out to compositional acts which lay within the order of magnitude of 2 seconds. Due to the reasons above, we fixed to two seconds the time limit spent over an interactive region enabling access to further layer. Throughout interactive sessions, the area of interest data is collected. Each visual component e.g. icons, has an associated 'counter' parameter which is incremented for every frame the user is continuously within the icon bounding box. Upon ending the program using the 'q' key this data is then used to give a complete percentage breakdown of user focus for each visual component.

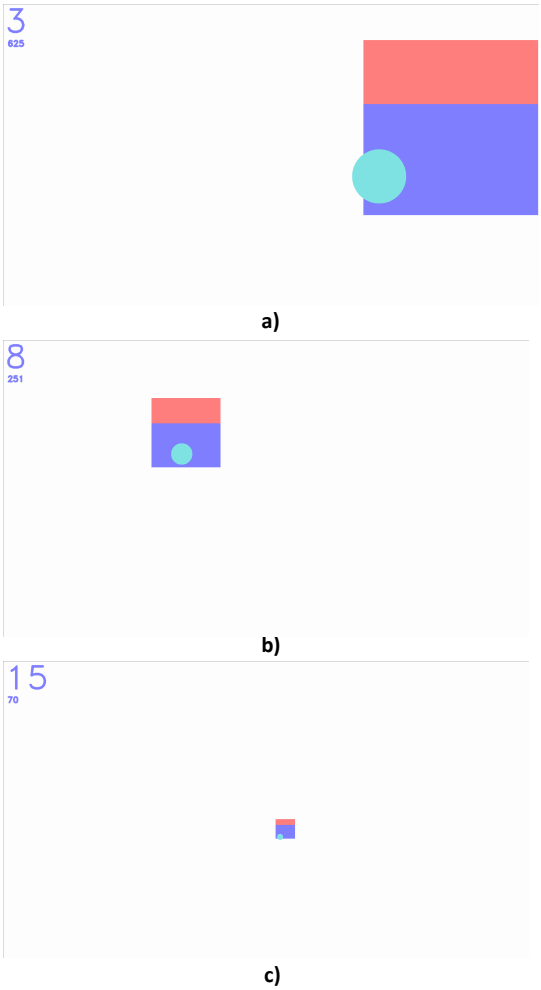

Figure 4: The subjects who took part in the experiments were asked to hover the cursor over the red square region. The main purpose was to assess on the easiness of control of the cursor on increasingly smaller regions. You can see a big-size (a), a medium-size (b) and a small-size interactive region (c) corresponding to different levels of test. The level is printed out on the upper left corner of the screen.

4 Experimental Results

The experimental sessions have been conducted on a 15.6-inch Dell Inspiron 5570 with 8 GB of RAM, Quad-Core Intel(R) Core(TM) i5-8250U CPU @ 1.60GHZ and Intel(R) UHD Graphics 620 and a screen resolution of 1920 x 1080. The purpose of this experiment was to test the usability and control of our program. We created a test in which participants would have to direct

Subject	Level	Region Size	Inches
1	18	41	0.28
2	23	16	0.11
3	20	28	0.19
4	20	28	0.19
5	19	34	0.24
6	21	23	0.16
7	20	28	0.19
8	22	20	0.14
9	19	34	0.24
10	19	34	0.24
11	21	23	0.16
12	18	41	0.28
13	20	28	0.19
14	21	23	0.16
15	19	34	0.24
16	20	28	0.19
17	20	28	0.19
18	21	23	0.16
19	18	41	0.28
20	20	28	0.19
21	22	20	0.17
22	17	49	0.44
23	22	20	0.17
24	19	34	0.30
25	17	49	0.44
26	16	58	0.52
27	17	49	0.44
28	22	20	0.17
Average	19.67	31.5	0.24

Table 1: The table shows the number of subjects taking part into the experimental session to assess the stability of Headgaze. The size of the red square region (in pixel units and inches) each subject is able to interact by using Headgaze returns a measure of the accuracy and stability of the method.

their Headgaze at a square region and hover the 'cursor' over it for a two second time period. If the participant is successful they will progress to the next 'level' containing a smaller region to select. The region decreases in size by a factor of 1.2 for each new level (see figure 4). This continues until the user can no longer select the given region within our specified time limit of 30 seconds. By observing the results of participants we will be able to assess the usability and control of our technique. For example, if the the average region is 'small', this would suggest that the program offers a high level of control, or vice versa.

Due to COVID-19 pandemic restriction, we had to limit our trials to team members and households (28).The average age of participants was 36 and the gender distribution was 15 women to 13 men. During the experiment the participant sat in a chair approximately 1.3m away from the laptop webcam. For the experiment we kept the location and hardware as constants. The results from the table 1 show that subjects on average are able to control the cursor with Headgaze up to a region of 31 by 31 pixels. We adopted this value to benchmark the level of stability and accuracy of the method. Furthermore, we calculated the average fps (frames per second) in the experimental sessions to be 21 fps. For further speed improvements the program could potentially be optimised using a compiled programming language.

After assessing the stability and easiness in the cursor control of Headgaze, we conducted other experiments where subjects may interact with multilayered architecture contents such as the ones in figure 5. Participants in the experimental session (see figure 5a,d) are shown the first layer with a layout characterized by interactive icons. They can make access to the second layer of the digital content architecture by hovering over an icon of interest for two seconds (see figure 5b,e). As soon as the time range of two seconds is met, the program triggers the access to the second layer (see figure 5c,f) showing more detailed contents related to the icon hovered in the previous step. Using the same mechanism described above, a participant can move back and forth across different levels of a given multilayered architecture.

5 Conclusions and future works

In our work, we focus our attention on the development of a low-cost and user-friendly solution to digital content interaction. The technique does not require any wearable devices or additional hardware. The method defines a new spatial coordinate system for a cursor overlay-

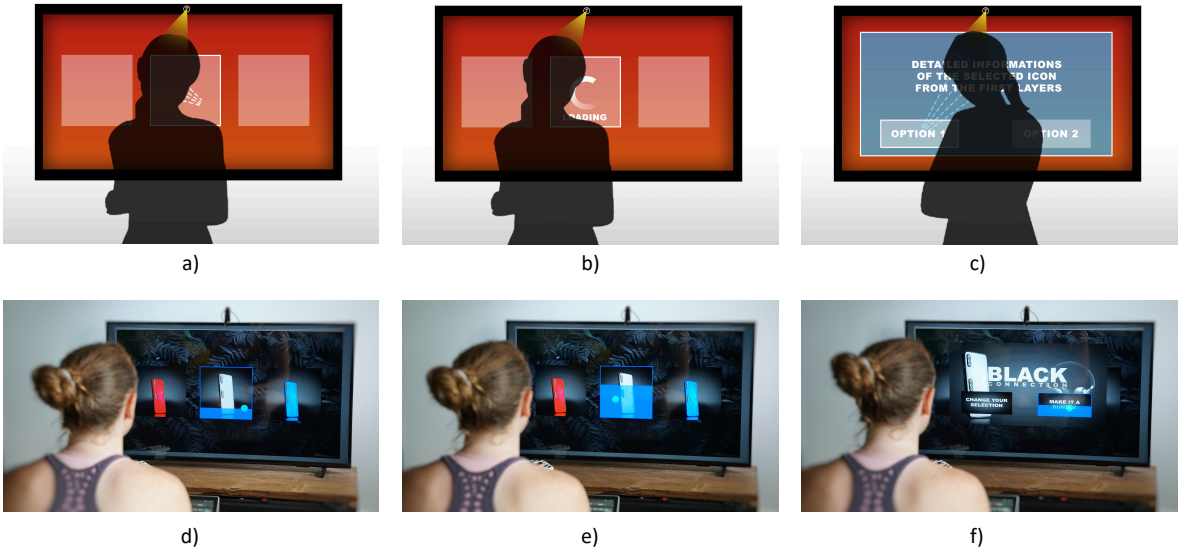

Figure 5: A subject focuses her 'head gaze' over the middle icon (a,d), a two-second loads animation (b, e) enables the transition to the corresponding second layer contents (c, f).

ing the digital content projected onto the screen. As soon as users' head movements are tracked down over a particular interactive icon, a loading bar or an animation highlights the loading of new contents letting users make access to the next layer. Our main objective is to provide users with a user-friendly and lightweight system. From our experiments we found that Headgaze allows users to interact with regions of size 31 by 31 pixels and above, at a distance of 1.3m and on a screen size of 15.6-inch (1920,1080). With the given setup of the experiment it would recommended to have all interactive content larger or equal to level 15 (70 by 70 pixels) to avoid difficulty for users. Optimal distances for our screen size appear to be in the range of 0.5-2m however further studies would be required in order to clarify this. The extensions proposed in our paper widen the usability of Headgaze over different application domains such as customer retail, human computer interaction, computer-based rehabilitation processes. Furthermore, we aim to integrate the current head pose project into a broader architecture involving other interaction tools such as webcam-based eye-tracking and hand-gestures detection. On the top of the integration of different interaction tools, we plan to build on a new visual library that utilises several gestures and expressions.

Acknowledgement

This research was supported by Innovate UK. Smart Grants (39012) - Shoppar: Dynamically Optimised Digital Content.

References

[1] J Vivek Veeriah and PL Swaminathan. Robust hand gesture recognition algorithm for simple mouse control. *International Journal of Computer and Communication Engineering*, 2(2):219–221, 2013.

[2] Raffaele Giaffreda, Radu-Laurentiu Vieriu, Edna Pasher, Gabriel Bendersky, Antonio J Jara, Joel JPC Rodrigues, Eliezer Dekel, and Benny Mandler. *Internet of Things. User-Centric IoT: First International Summit, IoT360 2014, Rome, Italy, October 27-28, 2014, Revised Selected Papers*, volume 150. Springer, 2015.

[3] Ahmad F Klaib, Nawaf O Alsrehin, Wasen Y Melhem, and Haneen O Bashtawi. Iot smart home using eye

tracking and voice interfaces for elderly and special needs people. *J. Commun*, 14:614–621, 2019.

[4] Qinjie Ju, René Chalon, and Stéphane Derrode. Assisted music score reading using fixed-gaze head movement: Empirical experiment and design implications. *Proceedings of the ACM on Human-Computer Interaction*, 3(EICS):1–29, 2019.

[5] Robert D Horning, Thomas Ohnstein, and Bernard Fritz. Wearable eye tracking system, March 19 2013. US Patent 8,398,239.

[6] Guy Bouvier, Yuta Senzai, and Massimo Scanziani. Head movements control the activity of primary visual cortex in a luminance dependent manner. *bioRxiv*, 2020.

[7] Vahid Kazemi and Josephine Sullivan. One millisecond face alignment with an ensemble of regression trees. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1867–1874, 2014.

[8] René de la Barré, Paul Chojecki, Ulrich Leiner, Lothar Mühlbach, and Detlef Ruschin. Touchless interaction-novel chances and challenges. In *International Conference on Human-Computer Interaction*, pages 161–169. Springer, 2009.

[9] Amer Al-Rahayfeh and Miad Faezipour. Eye tracking and head movement detection: A state-of-art survey. *IEEE journal of translational engineering in health and medicine*, 1:2100212–2100212, 2013.

[10] Jana Annina Müller, Dorothea Wendt, Birger Kollmeier, and Thomas Brand. Comparing eye tracking with electrooculography for measuring individual sentence comprehension duration. *PloS one*, 11(10), 2016.

[11] Kang Wang, Hui Su, and Qiang Ji. Neuro-inspired eye tracking with eye movement dynamics. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 9831–9840, 2019.

[12] Wattanit Hotrakool, Prarinya Siritanawan, and Toshiaki Kondo. A real-time eye-tracking method using time-varying gradient orientation patterns. In *ECTI-CON2010: The 2010 ECTI International Confernce on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology*, pages 492–496. IEEE, 2010.

[13] Reinier C Coetzer and Gerhard P Hancke. Eye detection for a real-time vehicle driver fatigue monitoring system. In *2011 IEEE Intelligent Vehicles Symposium (IV)*, pages 66–71. IEEE, 2011.

[14] Munir Oudah, Ali Al-Naji, and Javaan Chahl. Hand gesture recognition based on computer vision: A review of techniques. *Journal of Imaging*, 6(8):73, 2020.

[15] K Martin Sagayam and D Jude Hemanth. Hand posture and gesture recognition techniques for virtual reality applications: a survey. *Virtual Reality*, 21(2):91–107, 2017.

[16] Pramod Kumar Pisharady, Prahlad Vadakkepat, and Ai Poh Loh. Attention based detection and recognition of hand postures against complex backgrounds. *International Journal of Computer Vision*, 101(3):403–419, 2013.

[17] Ao Tang, Ke Lu, Yufei Wang, Jie Huang, and Houqiang Li. A real-time hand posture recognition system using deep neural networks. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 6(2):1–23, 2015.

[18] Stephen RH Langton and Vicki Bruce. You must see the point: automatic processing of cues to the direction of social attention. *Journal of Experimental Psychology: Human Perception and Performance*, 26(2):747, 2000.

[19] Nataniel Ruiz, Eunji Chong, and James M Rehg. Fine-grained head pose estimation without keypoints. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, pages 2074–2083, 2018.

[20] Massimiliano Patacchiola and Angelo Cangelosi. Head pose estimation in the wild using convolutional neural networks and adaptive gradient methods. *Pattern Recognition*, 71:132–143, 2017.

[21] Yingbo Li, Haolin Wei, David S Monaghan, and Noel E O'Connor. A low-cost head and eye tracking system for realistic eye movements in virtual avatars. In *International Conference on Multimedia Modeling*, pages 461–472. Springer, 2014.

[22] Garry Einicke. *Smoothing, filtering and prediction: Estimating the past, present and future*. BoD–Books on Demand, 2012.

[23] Nataliya Boyko, Oleg Basystiuk, and Nataliya Shakhovska. Performance evaluation and comparison of software for face recognition, based on dlib and opencv library. In *2018 IEEE Second International Conference on Data Stream Mining & Processing (DSMP)*, pages 478–482. IEEE, 2018.

[24] Christos Sagonas, Epameinondas Antonakos, Georgios Tzimiropoulos, Stefanos Zafeiriou, and Maja Pantic. 300 faces in-the-wild challenge: Database and results. *Image and vision computing*, 47:3–18, 2016.

[25] Christos Sagonas, Georgios Tzimiropoulos, Stefanos Zafeiriou, and Maja Pantic. 300 faces in-the-wild challenge: The first facial landmark localization challenge. In *Proceedings of the IEEE International Conference on Computer Vision Workshops*, pages 397–403, 2013.

[26] Christos Sagonas, Georgios Tzimiropoulos, Stefanos Zafeiriou, and Maja Pantic. A semi-automatic methodology for facial landmark annotation. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, pages 896–903, 2013.

[27] Joseph Howse. *OpenCV computer vision with python*. Packt Publishing Ltd, 2013.

[28] Richard L Lewis. Cognitive theory, soar. *International Encyclopedia of the Social and Behavioural Sciences*, 2001.