

FACIAL FEATURES TRACKING SYSTEM FOR ADAPTED HUMAN-COMPUTER INTERFACE

Rafael Santos, Arnaldo Abrantes, Pedro M. Jorge

Electronics, Telecommunication and Computer Engineering Department, Instituto Superior de Engenharia de Lisboa, Portugal

a37178@alunos.isel.pt aabrantes@deetc.isel.pt pjorge@deetc.isel.pt

Abstract — This work describes the implementation of an eye gaze tracking system for a natural user interface, based only on non-intrusive devices, like Intel RealSense camera. This camera has depth and infrared information which make the extraction of face features more robust. Through image processing, the implemented system is able to convert face elements to the corresponding focus point on the computer screen. Two approaches were taken, one using the eye gaze and the other using nose tracking to move the cursor. Preliminary tests show promising results.

Keywords: Eye gaze; nose tracking; image processing; human-computer interaction; 3D camera.

I. INTRODUCTION

The advance of technology has been motivating the development of new human-computer interfaces (HCI) [4]. The act of looking at a screen is part of most natural interaction processes. However, the information that the eye gaze can provide is still not entirely exploited on current HCI applications. Gathering and processing the eye gaze from a user to interact with the computer is a topic already studied [3,4], but mostly based on specific technologies that are not available in mass market devices, such as, laptops or tablets. Currently, most devices are equipped with a webcam, which can be used to collect visual information and provide feedback. However, this technology is not specific to acquire the information needed for eye gaze detection and tracking, missing in quality and sample rate. Even if those requirements were met, light condition can limit eye gaze accuracy. To avoid some detection problems, a camera with infra-red light and depth information is used. This work describes a system to detect eye gaze based on Intel RealSense F200 camera, enabling a more natural form of human-computer interaction. The manufacturer expects that this camera can replace the generic webcam and there are already some laptops that include it.

RealSense SDK not only allow us to access some useful camera functions but with a faster and better quality. To help on image processing tasks, OpenCV [5] is used with some useful functionalities, like, Haar cascade, Hough transform or Kalman filter.

The implemented system also includes other face elements, like the nose, that could replace eye movements in the proposed HCI.

This paper is organized as follows: section II presents the state of art of eye gaze systems. The implemented system is described in section III. Eye gaze results are evaluated in section IV and section V concludes this paper.

II. STATE OF THE ART

Eye gaze is a natural form of interaction, and identifying where a person is looking allows a computer to interact in a more human way [1,6]. However, replicating this procedure automatically in terms of human-computer interaction is not simple. Eye gaze has been the subject of several studies over the past years [1,2]. Rayner and Pollatsek [7] presented the first work that used electro oculography to measure retina movements to monitoring the user while reading. In 2003, Duchowski [8] proposed a similar method, where it was used a metallic hoop on contact lens to measure the electromagnetic field variations created by retina movement. Morimoto and Mimica [9], presented in 2004, a study where they used several techniques for eye gaze detection, where the main purpose was to develop interactive applications, based on eye structure and corneal reflex. However, none of these systems achieved good results, and the best results are obtained using specific and expensive hardware/equipment. New systems are being developed that use non-intrusive and more affordable devices. Among these systems, the best performances are obtained with a source of infrared light [3]. However, distrust of infrared light exposure motivated the development of eye gaze detection and tracking systems that use current technology, such as webcams [2]. These systems still have limitations associated with head movement compensation [2,3]. It is also necessary to improve real time processing algorithms and hardware [3]. At the end of 2014, Intel presented a new camera with depth, infrared and color information. The Intel RealSense is meant to work on tablets, computer and even smartphones. With this technology, the work proposed here is meant to update the one presented in [5].

III. EYE GAZE DETECTION SYSTEM

The main goal of the proposed work is to implement a new interaction system between human and computer with a state of art equipment. This method uses facial metrics to perform eye gaze tracking. It is desirable that the system comply some requirements like the use of commercial hardware, real time execution, enough precision for daily use and easy and fast calibration method.

The implemented system can be described by the block diagram of figure 1. The first block, Image Acquisition, is where the system acquires information from the camera; Acquire Information block is where the system processes the acquired data; based on the features points extracted from the face, the system estimate the cursor position on the screen display, performed by Cursor positioning block; finally, in

the last block, Perform Actions, the system will perform some actions requested by the user, besides cursor positioning, like clicking or moving a window.

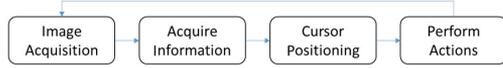


Figure 1- System block diagram.

A. Image Acquisition

The system starts with image acquisition from Intel RealSense camera (Image Acquisition block). It can be used two streams: Color and Depth images (see figure 2).

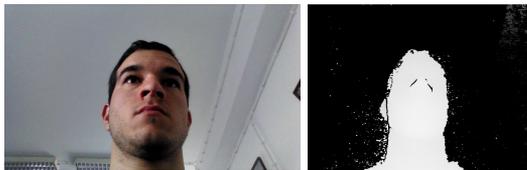


Figure 2 – RGB Image and Depth image.

B. Acquire Information

Based on the acquired image, the system processes the informations to extract the required features from the face of the user, such as, head orientation, eye, pupil and nose positions. This block is implemented in two different operation modes: (i) as a common webcam and (ii) as a 3D camera. In mode (i) it was used only the Color stream from the RealSense camera with some image processing algorithms [3,5]. Those algorithms consist on using Haar cascade to find out the face position and eyes position. It also uses Hough transform to find pupil circle and Kalman filter to smooth the eye movements. Mode (ii) use the infrared, depth and color information stream from the same camera. The camera SDK fuses those information to provide a more accurate and robust face features.

However, the results obtained with the webcam mode were not very different from the other works presented in the previous section. Figure 3 presents two frames where is possible to conclude that pupil tracking with traditional methods has not enough accuracy.

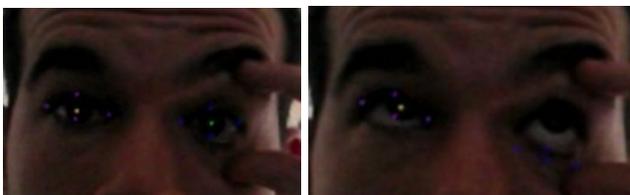


Figure 3 - Looking forward and looking up right.

Since RealSense camera has depth, infrared and color information, it is expected that better eye gaze tracking results can be obtain. For that propose this system uses the Real Sense SDK to acquire the required features. For face pose, Euler angles were used. On figure 4 an example is shown.

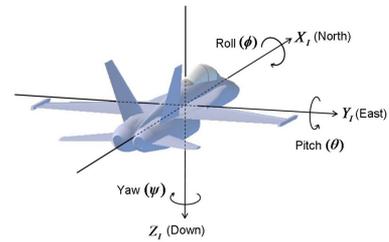


Figure 4 - Euler Angle applied to aeronautics.

After face pose information, the reference points were acquired. The system uses points from both eyes and nose (see figure 5 and 6).



Figure 5 - Eyes points.



Figure 6 - Nose point.

Although the SDK allows to acquire the reference points, their detection over time is very noisy. Even with the eye without movement the reference points have some variations on the position. To reduce the noise, a Kalman filter was used.

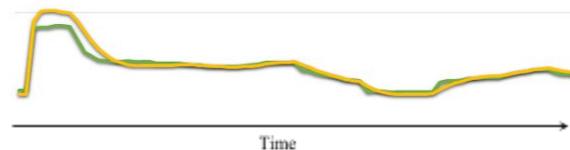


Figure 7 - Kalman filter applied to eye points. (X is the time and Y is the pupil horizontal coordinate)

Figure 7 shows the results of the kalman filter apply to the pupil position (green line) along eye movements around the screen. It is possible to see that, with Kalman filter, the detection is smoother (yellow line). However, Kalman filter prediction causes some delay. Nevertheless, the Kalman prediction allows the system to keep eye tracking even with eye's natural blinking.

C. Cursor Positioning

With face features, the system can process the information and execute actions. At this step, the system uses the previous information to determine the focus of attention point and position the cursor on the screen. The main objective is to use the eye as a mouse (illustrated in figure 9).



Figure 9 - Cursor movement.

For eye positioning several methods were tested:

- Eye as a mouse working with incremental position based on eye movements (up, down, left and right). See illustration in figure 10. The results were not as good as expected since we cannot make eye stop like a mouse. This approach has not enough precision;
- The cursor position is controlled by the pupil centre related with the eye. However, this method requires a calibration procedure to adjust the pupil position to the screen dimensions. For this propose a calibration pattern composed by 9 points is shown to the user (see figure 11). He should look at them in a predefined order and those eye positions are acquired. Finally, this information is used to relate eye and cursor positions. As the pupil movements are too small compared with screen dimensions, this approach has some limitations;
- The position of the pupil relative to the limits of the eye defines the position of the cursor relative to the screen (see figure 12). For this method the minimum and the maximum value for pupil centre variation are calculated. Those values define screen boundaries. Good results are achieved but the method needs to be improved due to small pupil movements and sensibility.

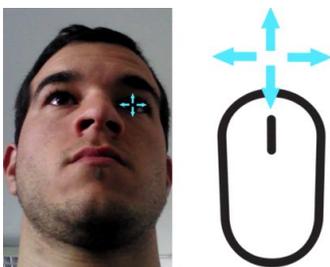


Figure 10 - Cursor movement as mouse differential.



Figure 11 - Calibration screen.

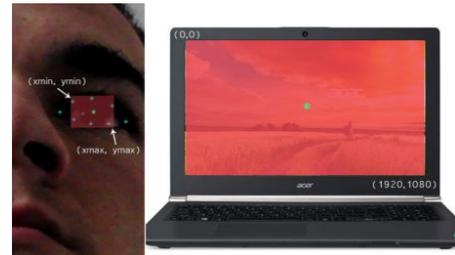


Figure 12 - Eye with screen reference.

However, the results are not as good as expected. So, to define the position of the cursor on the screen, another approach was implemented. Last approach is used, but now the cursor position is controlled by the nose position relative to a predefined window. Figure 13 illustrate this approach.



Figure 13 - Nose with screen reference.

This method obtains the best results, allowing the user to control the cursor position on the screen with good accuracy.

D. Perform Actions

Once the cursor is controlled with enough precision, some actions can be implemented, allowing the user to interact with the computer. The implemented actions are:

1. Eye blink to simulate left and right mouse click;
2. Mouth open to simulate left mouse click.

The first action (1) needs the eye circular reference points to determine if the eye is open or closed, as shown in figure 14. With those points, a minimum and maximum distances are calculated. The minimum distance would means that the eye is closed, and maximum that the eye is open (see figure 14).

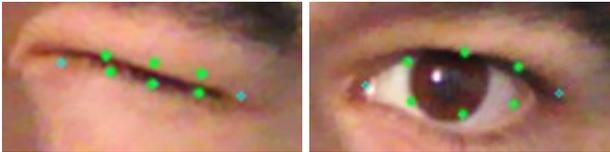


Figure 14 - Eye closed minimal distance and eye opened maximum distance.

Good results were achieved as the system is able to detect 7 clicks of 10 tries. With some training, the user is able to get 9 clicks of 10 tries, from the system.

The second action (2) was implemented to complement nose cursor positioning. As the user is moving the head/nose to move the cursor, eye blinking can not be used to perform actions. Alternatively, mouth is used to simulate mouse click (see figure 15). The results are better with this approach, being the system able to detect 9 click of 10 tries, without user training.



Figure 15 - Mouth open to determine the maximum distance.

IV. TESTING AND ANALYSIS

To evaluate system accuracy, several tests were carried out. Based on usability tests, the approach using eye movement had not enough accuracy. The user can actually move the cursor horizontally, but vertically no movement is practically noticed, since the pupil movement is minimal.



Figure 16 - Cursor movement looking at the top left side of the screen.



Figure 17 – Cursor movement looking at the top of the screen.

Usability tests were also performed with nose tracking approach. The user is able to move the cursor to the desired position, just with a short training (see figure 18). To perform mouse click with the eye blink, a few tries are needed for the user to get the desired result (see figure 19).



Figure 18 - Cursor positioning with nose movement.

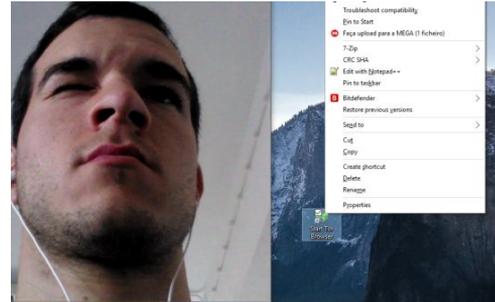


Figure 19 – Right eye blink to perform click over desktop icon.

With mouth open and close, the user is not only able to click over an icon, as he is able to move it to another place, doing drag and drop action (see figure 20).

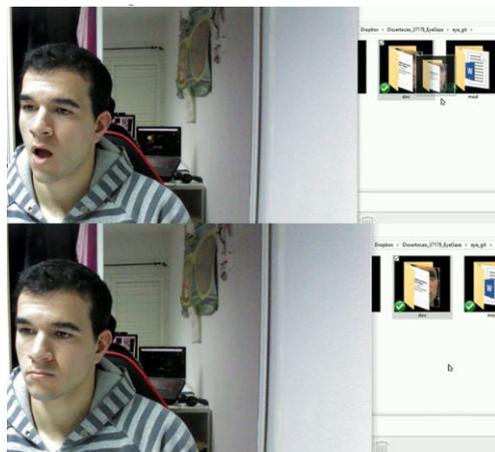


Figure 21 - Mouth open/close click and drag and drop.

Precision tests were performed with two screen dimensions, 24” and 15”, with two resolutions, FullHD and HD. To have some comparison between tests, a visual pattern is displayed on the screen, like a calibration screen. The user should look to different points from this pattern and the necessary information is acquired. Tables 1 and 2 show the errors between pattern and cursor points, for eye gaze and nose tracking, respectively.

		Cursor positioning - Eye gaze					Total	
		Test points						
		center	bottom left	bottom right	top left	top right		
E R R O R	FHD15	X	154	572	669	599	510	501
		Y	153	402	379	614	641	438
	FHD24	X	130	456	645	462	550	448
		Y	129	294	349	516	515	361
	HD15	X	27	479	399	307	343	311
		Y	62	262	279	344	351	260
	HD24	X	42	454	492	449	436	375
		Y	7	367	339	344	370	285
	Total	X	88	491	551	454	459	409
		Y	88	331	336	455	469	336

Table 1 - Cursor positioning error through eye gaze.

From table 1, eye gaze presents a mean error of 409 pixels for X coordinate (horizontal) and 336 for Y coordinate (vertical). The same test was performed for nose tracking and cursor positioning, the results are shown on table 2.

		Cursor positioning - Nose gaze					Total	
		Test points						
		center	bottom left	bottom right	top left	top right		
E R R O R	FHD15	X	54	366	23	3	115	112
		Y	31	21	25	0	140	43
	FHD24	X	22	5	16	0	66	22
		Y	25	16	17	9	9	15
	Total	X	25	123	13	1	60	45
		Y	19	12	14	3	50	20

Table 2 - Cursor positioning error through nose tracking.

From table 2, nose tracking represents a much smaller mean error, with 45 pixels for X coordinate (horizontal) and 20 pixels for Y coordinate (vertical). Tests show a big variability in the results. Mean error is lower with nose tracking than with eye gaze, due to the information that can be acquired with the nose and with the eyes.

V. CONCLUSION

This project implements a system that uses non-intrusive technology to develop a more natural human-computer interface. The ambient light is one of the main factors [5] that causes errors in eye gaze tracking. The use of Intel RealSense camera, that has three sources of information (infrared, depth and color), decreases the impact of the previous problem since the fusion of those information improves the robustness and the accuracy of tracking face features. Other aspect is user distance related with the camera. Camera's distance to the eye will influence pupil's detection since, eye resolution decreases and less information exist to process and acquire the pupil with precision.

With those requirements meet the system could get promising results, being the user be able to perform actions. Tests for cursor movement with eye show that work needs to be done in order to have better accuracy. One possibility is to develop head movement compensation for eye gaze, this way the eye estimated position can have more accuracy. In another way, nose tracking obtained enough precision to perform actions with good accuracy. Actions like drag & drop or mouse clicking to minimize a windows are possible to execute. Mouse clicking with mouth open/close showed good results as it complemented the nose tracking and cursor movement. A commercial solution for a human-computer

interface might be possible to implement if: (i) a Intel RealSense camera is available (or other device with the same information); (ii) the cursor movement is controlled with nose movements and (iii) mouse clicking is performed with mouth open/close.

REFERENCES

- [1] Poole, A. e Ball, L. J. (2005). Eye Tracking in Human-Computer Interaction and Usability Research: Current Status and Future Prospects. In C. Ghaoui (Ed.) *Encyclopedia of Human-Computer Interaction*, Pennsylvania: Idea Group, Inc.
- [2] Tunhua, B. B. W., Changle, L. S. Z. e Kunhui, L. (2010). Real-time Non-intrusive Eye Tracking for Human-Computer Interaction. *Proceedings of the 5th International Conference on Computer Science and Education (ICCSSE)*.1092-1096.
- [3] Wild, D. J. (2012). Gaze tracking using a regular web camera.
- [4] Drewes, H. (2010). Eye Gaze Tracking For Human Computer Interaction.
- [5] Santos, R., Santos, N., Jorge, P. e Abrantes, A. (2013) Eye Tracking System With Common Webcam. Lisboa, ISEL. Elsevier.
- [6] Jaimes, A. and Sebe, N.(2007). Multimodal human computer interaction: A survey. *Computer Vision and Image Understanding*, 1-2 (108):116-134.
- [7] Rayner, K. and Pollatsek, A. (1989). The psychology of reading. Prentice Hall, NJ.
- [8] Duchowski, A. T. (2003). Eye tracking methodology: Theory and practice. Springer-Verlag Ltd, London.
- [9] Morimoto, C. H. and Mimica, M. R. (2004) Eye Gaze tracking techniques for interactive applications. Elsevier Inc.