BULLETIN OF THE POLISH ACADEMY OF SCIENCES
TECHNICAL SCIENCES, Vol. 63, No. 4, 2015
DOI: 10.1515/bpasts-2015-0100

Single web camera robust interactive eye-gaze tracking method

A. WOJCIECHOWSKI* and K. FORNALCZYK

Institute of Information Technology, Lodz University of Technology, 215 Wólczańska St., 90-924 Łódź, Poland

Abstract. Eye-gaze tracking is an aspect of human-computer interaction still growing in popularity,. Tracking human gaze point can help control user interfaces and may help evaluate graphical user interfaces. At the same time professional eye-trackers are very expensive and thus unavailable for most of user interface researchers and small companies. The paper presents very effective, low cost, computer vision based, interactive eye-gaze tracking method. On contrary to other authors results the method achieves very high precision (about 1.5 deg horizontally and 2.5 deg vertically) at 20 fps performance, exploiting a simple HD web camera with reasonable environment restrictions. The paper describes the algorithms used in the eye-gaze tracking method and results of experimental tests, both static absolute point of interest estimation, and dynamic functional gaze controlled cursor steering.

Key words: eye-gaze tracking, computer vision, human computer interaction, real-time tracking.

1. Introduction

Among eye tracking systems two main branches of their functionality can be distinguished: diagnostic and interactive [1]. As the first group is mainly dedicated to precise medical [2–4] ergonomic [5, 6] biometric [7] or marketing [8, 9] researches, interactive eye-gaze tracking systems provide solutions mainly for selective human computer interaction and gaze-contingent solutions [10–29]. Interactive eye-gaze tracking methods, exploited in human computer interaction field, can be also complemented by other human senses related modalities like, voice communication [30–33], hand movements [34–36] or even brain waves analysis [37], satisfying broad natural communication with computer.

Though interactive gaze tracking systems have relatively lower precision and accuracy, than diagnostic one, such systems assure, from the application point of view, fully functional, robust, real-time response and system interactive reaction. There were several attempts, where interactive solutions stability and precision improvement was considerably enhanced by system intrusiveness. Due to special devices attached directly to the skin or enforcing users to wear uncomfortable glasses or helmets [38-40], systems reduce correlation costs between eyes related and eye tracking device related, coordinating systems.

Fortunately, due to computer vision methods development and digital cameras increasing performance, non-intrusive, appearance based methods have been still growing in popularity [41–45]. Using computer vision and geometrical properties of the eyes, gaze direction can be estimated without the need of any kind of physical contact with the user. Great amount of non-intrusive gaze tracking methods exploited dedicated IR light sources [43–50]. Among them there are professional video-based eye trackers which still are relatively expensive (i.e.: Tobii eye-tracker).

Meanwhile authors claim that, especially for interactive eye-tracking system, non-professional equipment without dedicated light sources can work sufficiently, on condition overcoming several barriers. Simple video cameras are widely available, and when equipped with appropriately elaborated gaze tracking software, a powerful interactive eye-tracking system can be constructed.

Though Hansen [47] review has shown that appearance based methods, considering daylight environment, are confined to a precision of about 5 deg, authors have elaborated a method of considerably higher precision.

Presented paper describes a simple, non-intrusive, interactive and effective method for tracking eye movements and eye blinking detection. Method did not enforce any sophisticated lighting condition, besides moderately uniform illumination assuring perceptible contrast between skin color and colors of eyelid and iris. Method was not verified against people with albinism and other skin and iris color extinction diseases.

Additionally, the method requires HD web camera and limited head movement – head orientation compensation was necessary for the whole gaze-tracking system but it was not discussed within this paper. Nevertheless image based approach, the method belongs to, may fail under strongly uncomfortable lighting (strong shadows encompassing eyes) and color conditions or if a user has been wearing glasses occluding crucial part of eye region. However satisfying mentioned, moderate lighting and skin color assumptions, after simple calibration, method has provided gaze point estimation high accuracy. Moreover newly proposed geometric eyes' features vectors can be very effectively calculated for low resolution eyes' images.

Very promising tests, performed with the implemented method, were also described. Method was tested for both static gaze-point estimation and dynamic, gaze-controlled cursor steering.

---

*e-mail: adam.wojciechowski@p.lodz.pl

2. Related work

Among different interactive gaze tracking systems, several non-intrusive appearance based selective and gaze-contingent solutions can be found.

In selective interactive systems, eyes actively control the system by means of gaze direction. A possible use of eye trackers typically employ gaze as a pointing modality, i.e. in similar manner to a mouse pointer. It can enable a user interface to be sensitive to the attention of a user. Sibert and Jacob [51] showed that eye tracking interfaces are both usable and superior to mouse driven interfaces for some metrics. Possible applications involve selection of interface items as well as selection of objects or areas of virtual environment. Jacob [10] demonstrated an intelligent gaze-based informational display. The "What You Look At Is What You Get" idea let to scroll the window to show information on visually selected items. Starker and Bolt [11] presented gaze-controlled navigation in a three dimensional environment. Tanriverdi and Jacob [12] elaborated an eye-based interactive system with gaze acting as a selective mechanism in VR. Another prototypical application of interactive eye tracking was screen displayed keyboard typing [13, 14], screen gaze controlled drawing [15] or blink controlled web browsing [19], and such systems were dedicated mainly for handicapped people. Santalla et al. [16] adaptively cropped photographs basing on averaged region of interest.

Interactive systems exploiting gaze direction for screen display direct changes are called gaze-contingent systems. Having user's point of regard, a system can tailor the display or scene content so that the details are displayed at the point (region) of gaze and degraded in the periphery. The idea behind such systems is minimizing bandwidth requirements. Within peripheral region of image-based approach, authors suggested to diminish quality of scene rendering by decreasing visual details of displayed objects [20–23] or to change the level of image codec compression [24, 25]. For object-based approach authors suggested reducing resolution by affecting objects' geometry prior to rendering [26–29]. O'Sullivan [30, 31] developed also gaze-contingent collision handling system.

Within quoted applications, powered by non-intrusive, single camera, interactive, gaze tracking methods, two main technical approaches were distinguished. First, the most popular, considered light corneal reflections or glints on corneal surface caused by dedicated light sources [43, 46, 47]. Another approach tended to evaluate geometric characteristics of eye components (pupil, iris, sclera, limbus) interrelations and analyzed them in order to estimate gaze direction [47, 52–55].

Nguyen [53] and Williams [54] have localized eyes in the camera derived calibration images and then calibration samples based, Gaussian process was used to calculate the predictive distribution, transforming eyes images into gaze point on the screen. For training the distribution function, neural networks were used. Also Sewell [56] has used eyes image trained neural network for gaze estimation.

Betke [55], Liu [57] and Kao [58] suggested tracking characteristic features of the eye regions throughout all subsequent camera frames. Once recorded, fragment of eye ball was located in subsequent frames basing on the best correlation coefficients [55], mean-shift algorithm [57] or pattern voting scheme [58]. Authors reported that direct mapping of eye center onto mouse cursor position, had resulted in a miserable precision.

Magee [59] has evaluated gaze direction basing on adjustment of eyes' images. Subtraction of left and right eyes images (one of the eye images was mirrored) let evaluate gaze direction. Nevertheless method's tests have limited its functionality to moving cursor horizontally, to the left or to the right respectively.

Yamazoe [60] and Ishikawa [61] have proposed eye model based gaze estimation exploiting iris center, eye ball center and eye corners interrelations. However, the obtained precision was only about 5 deg. horizontally and 7 deg. vertically.

Method presented in the paper, neither imposes severe lighting conditions nor requires long training. Assuming that user, being interested in some point of the screen, is enforced to freeze his head for at least several frames, the head and eyes were tracked on frames by direct highly optimized Haar-like classifier and upgraded between similar frames. Thus method does not use any sophisticated features tracking algorithm [62]. Similarly to Yamazoe and Magee, gaze estimation was based on eyes images features, but features retrieval was performed only in image space. In reward method provided highly interactive frame rate of about 20 fps and on contrary to Hansen [47] conclusions it featured high accuracy of about 1.5 degree horizontal and 4.5 degrees vertical gaze tracking angular resolution.

3. Gaze point tracking method

The main goal of the method was to determine whether the user is looking at the screen or not, and if so – to determine the point on which the user's eyes are focused (gaze point). The method relies on eyes region color analysis so it assumed that eyelashes (emphasized by open eye eyelid wrinkles shadows) and iris contrast decently with the skin color and no external obstacles like thick border glasses, occluded or competed with eyes appearance. On successful eyes detection, gaze analysis could be preceded. It determined whether the user is looking at certain place, or the gaze point is moving. This task can be divided into five general stages:

1. Finding user's face in the image.
2. Finding (on the face) eyes.
3. Iris and upper eyelid region segmentation.
4. Determining whether the eyes are opened.
5. If eyes are opened, estimating gaze point.

The particular stages are briefly discussed below.

In order to find user's face within the image, the cascade of Haar-like features classifiers (describing face) was used [63]. This solution, though computationally demanding, has been chosen due to its effectiveness and ease of use. Face detection performance was improved by simple optimizations. Assuming, the system is dedicated to one user sitting in front of the camera, the biggest region satisfying Haar-like classifier was chosen as a valid face region. Subsequent iterations, of the face localization, were initiated if the aggregated pixels' values have changed more than 2% in relation to previous frame. Otherwise former face region was set as valid for at maximum 3 iterations.

The next stage was to find the user's eyes. For this purpose the cascade of Haar-like features classifiers (describing eyes pair) was used [63, 64]. The eyes searching task was performed only within the face region. Additional boosting assumption was that stripe containing eyes should be localized within upper part of the face region. Within the face region, 20% from the top (forehead) and 40% from the bottom (face below nostrils) can be cropped and rejected from further analysis [65]. Inspected region limitation let to gain several milliseconds for every cycle of the method.

Subsequent stage was to segment iris and upper eyelid region. It was noticed that upper eyelid (with eyelashes) reflects eyeball extent and iris to eyelid relative position may reflect gaze point after adequate calibration. As a consequence eyes region was divided into two parts, comprising right and left eye respectively (Fig. 1a). Consecutive operations were performed for both eyes separately and averaged at the end of the whole process, as the brightness of each eye region may differ.

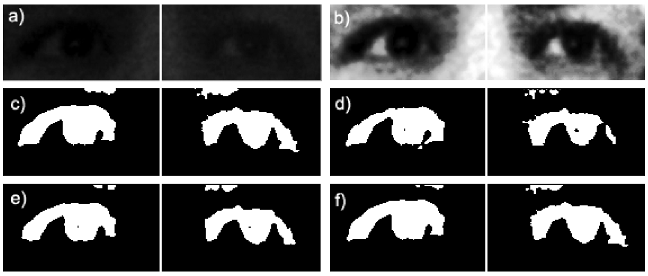

Fig. 1. Eyes inverted binarization with adaptive thresholding; a) original web camera image; b) image after histogram equalization; c) manual thresholding; d) mean value thresholding; e) Otsu method based thresholding Ref [67]; f) p-tile thresholding, after Ref [68]

Introductory step was eye's image histogram equalization (Fig. 1b). Subsequently, inverted binarization (low intensities mapped to '1' and high to '0'), both for manually estimated threshold (Fig. 1c), and several adaptive threshold estimating methods were tested for iris and upper eyelid segmentation (Fig. 1d, 1e, 1f) [66–69]. Appropriate threshold value should let separate iris and upper eyelid region contour from the eyebrow contour and other eye region elements. The most valuable were Otsu [67] (Fig. 1e) and empirically adjusted (19%) percentile thresholding [68] (Fig. 1f) based binarization. OpenCV library default adaptive mean image value thresholding has degraded slightly the contour and finally it was rejected (Fig. 1d).

To delete possible noise, which has remained after binarization, operations of 1 pixel erosion and twice sequentially applied 1 pixel dilatation were used.

The fourth stage was devoted to the eyes opening verification. In fact it was performed twofold in order to find the most appropriate solution. The first approach was based on template matching [19], where cross-correlation between templates acquired during initialization of the system (Fig. 2) and non segmented current eye's images (before stage 3) (Fig. 1b) was calculated, with the 70% correlation threshold.

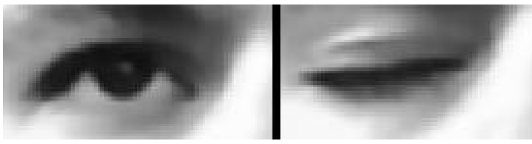

Fig. 2. Exemplary eye images used as templates

Second approach was based on analysis of the eye regions segmented within the 3rd stage of the method. As two considered approaches have not revealed any significant differences, both in precision (about 90%) and performance, the second approach has been chosen as valid, mainly due to consistency with the contour processing pipeline.

Within further processing the largest eye representing region was selected. Possibly disjoint segmented regions were concatenated – all minor contours found between horizontal lines bounding main region from top and bottom (Fig. 3a). Other minor regions beyond two horizontal lines were cropped. In consequence the valid eye region was constructed. Then region area based eye opening verification was possible. The heuristic method was inspired by observation of eye region area and its convex hull area (Fig. 3). It was noticed that closed eye region has similar area to its convex hull area, while opened eye region was significantly smaller than its convex hull area. Conditions describing proportion between eye region area and its convex hull area are formulated as follows:

$$\begin{aligned} \frac{S_{eye\_reg\_area}}{S_{conv\_hull\_area}} &> 85\% \quad - \quad eye\ is\ closed \\ \frac{S_{eye\_reg\_area}}{S_{conv\_hull\_area}} &\leq 85\% \quad - \quad eye\ is\ opened. \end{aligned} \tag{1}$$

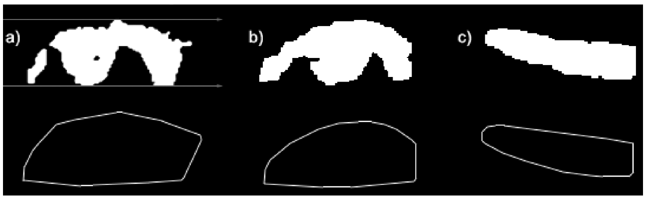

Fig. 3. Eye segmented regions and their convex hulls a) opened eye disjoint region and its convex hull; b) opened eye joint region and its convex hull; c) closed eye region and its convex hull

If eye region area ($S_{eye\_reg\_area}$) occupies more than 85% (usually more than 90%) of the eye region's convex hull area ($S_{conv\_hull\_area}$) the eye is closed, if less – eye is opened (usually between 60% and 80%).

The last stage was to determine the gaze point through analysis of resulting eye region. This was done in several steps:

- calculating a window coordinating system axis aligned bounding rectangle for segmented contour;

- closing morphologically the segmented eyes regions (Fig. 4a);
- calculating their distance transform and normalizing results with MIN-MAX type normalization (Fig. 4c);
- calculating the resulting image geometric moments: bounding rectangle centroid (RC) and contour area first order moments (center of gravity – CG) (Fig. 4d);
- calculating the vector v connecting the above points (Eq. (2)) within image coordinate system;

$$\overrightarrow{v} = \langle v_x; v_y \rangle = \langle CG_x - RC_x; CG_y - RC_y \rangle; \tag{2}$$

- normalizing vector v by dividing the v_x coordinate by the width and v_y coordinate by the height of the contour bounding rectangle (Fig. 4d).

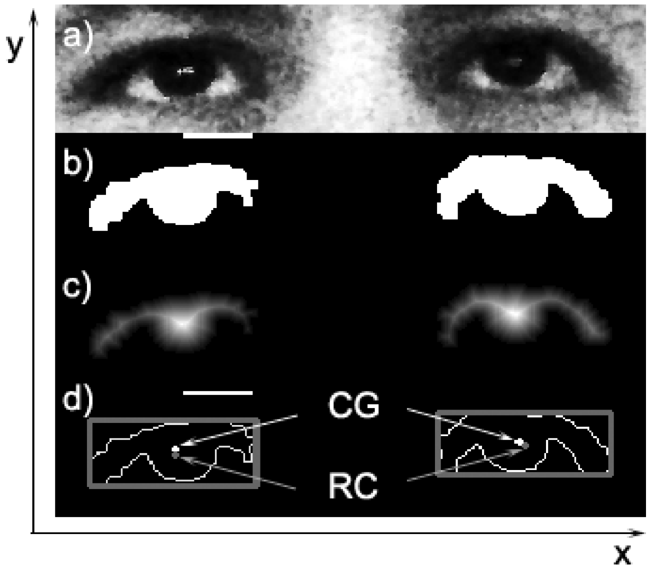

Fig. 4. Eye contours, theirs bounding rectangles, centers of gravity (white points – CG) and contour bounding rectangles centroids (grey points – RC)

As it is presented in Fig. 5, relation between the contour center of gravity (CG) and its bounding volume centroid (RC) reflects changes in gaze direction. Analysis of the vector v, upon previous calibration, allowed to determine user gaze point.

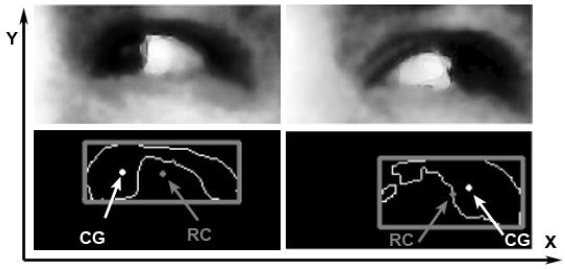

Fig. 5. Eye contours, theirs bounding rectangles, centers of gravity (white points – CG) and contour bounding rectangles centroids (grey points – RC)

The goal of the calibration was to find linear conversion coefficients transforming the v vector components into screen gaze point coordinates. During calibration user was asked to record a specific number of vector v samples for each corner of the screen. Then, for each individual corner, the arithmetic mean of all samples was calculated. In this way 4 points A, B, C, D representing vector v mean values, corresponding to screen corners, were obtained (Fig. 6).

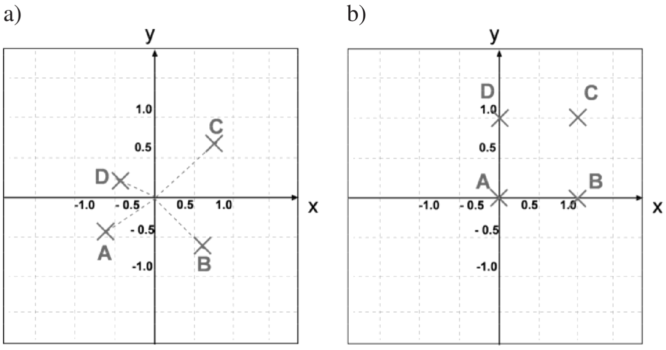

Fig. 6. Set of four points A, B, C, D and their coordinates representing averaged screen corners: a) averaged vector v values collected for each screen corner; b) averaged vector v values after transformation according to algorithm 1

In theory, the points (A, B, C, D) should create a rectangle, but due to eye movement interpersonal variability, the points usually form a quadrangle (Fig. 6a). They should be transformed so the resulting vertices were located at (0, 0), (1, 0), (1, 1), (0, 1) (Fig. 6b).

Process of characteristic points (A, B, C, D) transformation, was performed according to Algorithm 1.

Algorithm 1. Algorithm of points $A = \langle A_x; A_y \rangle$, $B = \langle B_x; B_y \rangle$, $C = \langle C_x; C_y \rangle$, $D = \langle D_x; D_y \rangle$ coordinates normalization.

step **I**	coordinates of points A, B, C, D should be translated by vector $\langle -A_x; -A_y \rangle$
step **II**	Coordinates of points B,C,D should be transformed according to equations: $x := x + (y/D_y)\,(-D_x);$ $y := y + (y/D_y)\,(1 - D_y);$
step **III**	coordinates of points B, C should be transformed according to equations: $x := x + (x/B_x)\,(1 - B_x);$ $y := y + (x/B_y)\,(-B_y);$
step **IV**	coordinates of points C should be transformed according to equations: $x := x + (y/C_y)\,(x/C_x)\,(1 - C_x);$ $y := y + (y/C_y)\,(x/C_x)\,(1 - C_y).$

Sequence of operations from Algorithm 1 was recorded and its transformation coefficients were saved as a corresponding set of linear conversion coefficients. These coefficients were used for gaze point estimation during tests of the application.

After completing the calibration, gaze point estimation tests were possible. Current frame gaze point was estimated according to procedure described in Algorithm 1. The frame absolute screen coordinates can be calculated as a multiplication of screen resolution and current resulting calibration coefficients – the lower left corner of the screen is represented by coordinates (0, 0) and the right top by (1, 1). If one of the coordinates was less than 0 or greater than 1 it has meant that the user was not looking at the screen.

As frames derived coefficients provided noisy results, any type of filtering should be applied. Thus certain number of samples was collected and averaged. In result, certain number (discussed later in tests) of previous frames values influenced currently estimated gaze point.

4. Tests

Efficiency of the method was tested on a group of 10 information technology students. The students were between 21 and 23 years old, and they did not have glasses. All the tests were performed using Creative Live! Cam Sync HD 720p web camera. The key parameter of this device was an image resolution of 1280 × 720 pixels at 30 frames per second.

The first group of tests goal was to measure a static position of the gaze point fixation in relation to the mouse cursor position, the eyes were concentrated on. The testers were placed about 75 m away from the monitor and the camera was positioned on the laptop keyboard, midway between the screen and the user. Tests were carried out on ten persons under the same conditions – with unified lighting and possibly not moving head. In each test 25 measurements were made. Achieved results are collected in Table 1.

Table 1
The average results of accuracy tests

No. of calibr. samples	No. of aver. samples during tests	Aver. x coord. diff. [pix]	Stand. dev. x [pix]	Aver. y coord. diff. [pix]	Stand. dev. y [pix]	Abs. distance [pix]	Stand. dev. distance [pix]
50	20	85.72	70.83	83.28	51.92	135.48	58.91
100	30	52.04	29.26	76.96	56.46	101.42	48.16
200	60	64.96	53.95	71.4	56.59	109.68	57.34

As one might notice, during each test, the differences between the x coordinates were usually smaller than between the y coordinates. Apparently presented algorithm was doing worse in determination of correct value of the gaze point for y coordinate. Obtained precision of about 52 pixels (1 deg.[1].) horizontally and 77 pixels (1.5 deg.[3]) vertically (101 pixels for distance – 1.95 deg.[3]) corresponds to metric precision of about 1.31 cm horizontally and 1.96 cm vertically (2.56 cm for distance). Taking into consideration measurements substantial standard deviation of about 29 pixels horizontally (about 0.73 cm) and about 56 pixels vertically (about 1.41 cm) and assumed normal error distribution, the precision of the method falls down (with probability $p > 68\%$) to about 2.04 cm (1.56 deg.[3]) horizontally and 3.37 cm vertically (2.57 deg.[3]). These results sound very reasonably as professional, IR based, eye tracking system accuracy[2], reached about 0.65 cm (0.5 deg.[3]). It must be remarked that collected individual data will be further filtered.

One may also notice, that increasing the number of samples between 2nd and 3rd test had no significant effect on the further improvement in the accuracy. It seems, that current method has probably reached the limits of its accuracy in this aspect and further increasing the number of samples would be pointless. However it requires further method studies.

Second group of tests goal was to verify functional aspects of the eye-tracking method, while dynamic managing gaze controlled cursor. Users were asked to follow the provided path (Fig. 7), from the darkest segment to the brightest one, with a cursor controlled independently by three different input controllers: mouse, touchpad and finally web camera supported with authors' eye-gaze tracking method. The segment was accepted as visited if the cursor spent inside at least 0.5 second.

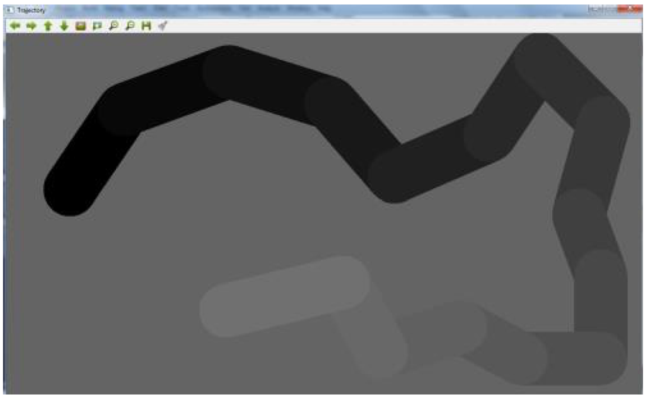

Fig. 7. Trajectory followed by cursor controlled with mouse, touchpad and eye-gaze tracking method

During experiments, the followed path had four different thicknesses: 50, 100, 150 and 200 pixels. Display screen resolution was 1366 × 768 pixels and corresponding physical dimension was 34.5 × 19.5 centimeters. In result 1 centimeter corresponded to about 40 pixels.

Users have started from the darkest segment and were asked to move the cursor with provided input controller along the path. Total time of cursor passage was measured and percentage of time spent inside the path in comparison to total time was collected. The sets of measurements were collected individually for each tested path thickness. For each test 10 measurements were made. Achieved results were collected in a Table 2.

Results of the eye-tracking dynamic precision tests have shown that for paths of 150 and 200 pixels in width, gaze controlled cursor movements were very precise and they were comparable with mouse and touchpad accuracy. Even doubled cursor passage time was not severely perceptible as the experience of controlling cursor with a mouse was not comparable with the gaze-based interaction. Following the path

---

[1] Viewing angle estimated considering on screen metrical precision and the eyes to the screen distance of about 75 cm

[2] Tobii Test specification, Accuracy and precision test method for remote eye trackers, http://www.tobii.com/Global/Analysis/Training/Metrics/, (April 2013).

of 100 pixels in width (about 2.5 cm) became a challenge for the method but it has satisfied the cursor positioning expectations in almost 80%, what made a very reasonable achievement. Almost tripled time of passage in comparison with a mouse was rewarded and justified by new user impressions. Unfortunately the path of 50 pixels in width (about 1.25 cm) was still beyond the scope of the method, not only due to time devoted to cover the path distance but due to unacceptably low percentage of time (about 40%) spent within a path.

Table 2
The average results of path controlling test

Path width [pixels]	50 pixels width	100 pixels width	150 pixels width	200 pixels width
Mouse total time [ms]	8269 ms	7952 ms	7918 ms	8339 ms
Mouse time inside the path [%]	98.09 %	99.73 %	100 %	100 %
Touchpad total time [ms]	12043 ms	10209 ms	9083 ms	8780 ms
Touchpad time inside the path [%]	99.68 %	99.74 %	100 %	100 %
Camera total time [ms]	35006 ms	23605 ms	17331 ms	16894 ms
Camera time inside the path [%]	40.39 %	78.97 %	98.86 %	99.20 %

The tests were also carried out using VerySleepy[3] profiler. Their aim was to determine what percentage of the time was spent on carrying out the various algorithm steps. The results of the test are shown in a Table 3. The value of 100% represents all the time needed to analyze the image (without displaying converted images). The table shows only the operations, which executions took more than 5% of the total time. Speed of whole system during test was about 20 frames per second.

Table 3
The results of CPU speed tests

Activity	% of time devoted to it
Face detection	24
Eyes detection	23
Further transformation of the image section that contains the eyes and calculating the gaze point	20
Printing information on the console	9
Image conversion (RGB -> GRAY)	7.5

Apparently, almost half of the time was consumed on face and eyes detection. It shows, that this part of the algorithm in the first place, should be subjected to a more detailed analysis and optimization. Tests have shown also that the system has a lot of potential and there is still a lot of room for improvements.

5. Conclusions

Provided paper has described simple and very effective eye-gaze tracking method. Results of its accuracy and speed tests have been presented as well. The analysis of the results has shown that method was reasonably precise. Its metric resolution was about 2.04 cm horizontally and 3.37 cm vertically what corresponds to angular resolution of 1.56 degree horizontally and 2.56 degrees vertically respectively.

The gaze tracking system performance was about 20 fps what suited for most of interactive application requirements. System performance has decreased to about 15 fps upon user rapid head movements due to time demanding process of face and eyes searching. In this context using some techniques of optical flow analysis [62], or two correlated cameras, one capturing the face and second searching for eyes within the face, instead of current solution, may give similar results in shorter time.

Method was tested not only statically, for its absolute precision, but also for its dynamic functionality. Very successful rate of 80% for eye-gaze cursor control over the path of 100 pixels in width (about 2.5 cm) and almost 100% success rate for wider paths, let exploit it as a fully professional input device for human computer interaction tasks.

REFERENCES

[1] A. Duchowski, *Eye Tracking Methodology: Theory and Practice*, Springer, Berlin, 2007.

[2] J.M. Henderson, “Human gaze control during real-world scene perception”, *Trends in Cognitive Sciences* 7 (11), 498–504 (2003).

[3] K. Rayner, “Eye movements and the perceptual span in beginning and skilled readers”, *J. Experimental Child Psychology* 41, 211–236 (1986).

[4] D.M. Snodderly, I. Kagan, and M. Gur, “Selective activation of visual cortex neurons by ?xational eye movements: implications for neural coding”, *Visual Neuroscience* 18, 259–277 (2001).

[5] S. Josephson and M.E. Holmes, “Clutter or content? How on-screen enhancements affect how TV viewers scan and what they learn”, *Proc. Eye Tracking Research and Applications ETRA* 1, 155–162 (2006).

[6] G.L. Lohse, “Consumer eye movement patterns on yellow pages advertising”, *J. Advertising* 26 (1), 61–73 (1997).

[7] K. Ślot, *Selected Aspects of Biometry*, WKŁ, Warsaw, 2008, (in Polish).

[8] M.M. Hayhoe, D.H. Ballard, J. Triesch, H. Shinoda, P. Aivar, and B. Sullivan, “Vision in natural and virtual environments”, *Proc. ACM Symp. on Eye Tracking Research and Application* 1, 7–13 (2002).

[9] G. Ho, C.T. Scialfa, J.K. Caird, and T. Graw, “Visual search for traffic signs: the effects of clutter, luminance and aging”, *Human Factors* 43 (3), 194–207 (2001).

[10] R.J. Jacob, “What you look is what you get: eye movement-based interaction techniques”, *ACM Conf. Proc. Human Factors in Computing Systems: CHI* 90, 11–18 (1990).

---

[3] Sleepy CPU profiler for Windows – http://www.codersnotes.com/sleepy.

[11] I. Starker and R.A. Bolt, "A gaze-responsive self-disclosing display", *ACM Conf. Proc. on Human Factors in Computing Systems* 1, 3–9 (1990).

[12] V. Tanriverdi and R.J.K. Jacob, "Interacting with eye movements in virtual environments", *ACM Conf. Proc. on Human Factors in Comp. Sci.* 1, 265–272 (2000).

[13] P. Kotze, M. Eloff, A. Adesina-Ojo, and J. Eloff, "Accessible computer interaction for people with disabilities: the case of quadriplegics", *6th Int. Conf. on Ent. Inf. Sys.* 1, 14–17 (2004).

[14] P. Majaranta and KJ. Raiha, "Twenty years of eye typing: systems and design issues", *Proc. ETRA* 1, 15–22 (2002).

[15] A.J. Hornof and A. Cavender, "EyeDraw: enabling children with severe motor impairments to draw with their eyes", *Proc. SIGCHI Conference on Human Factors in Computing Systems, ACM Press* 1, 161–170 (2005).

[16] A. Santella, M. Agrawala, D. DeCarlo, D. Salesin, and M. Cohen, "Gaze-based interaction for semi-automatic photo cropping", *Proc. SIGCHI Conf. on Human Factors in Computing Systems, ACM Press* 1, 781–790 (2006).

[17] B.H. McCormick, D.A. Batte, and A.T. Duchowski, "A virtual environment: exploring the brain forest", *Computation Sistemas* 1 (1), 5–13 (1997).

[18] B. Watson, N. Walker, L.F. Hodges, and A. Worden, "Managing level of detail through peripheral degradation: effects on search performance with a head-mounted display", *ACM Transactions on CHI* 4 (4), 323–346 (1997).

[19] A. Królak and P. Strumiłło, "Eye-blink detection system for human-computer interaction", *Univers. Access Inf. Soc.* 10, 1–11 (2011).

[20] B. Watson, N. Walker, and L.F. Hodges, "Supra threshold control of peripheral LOD", *Trans. on Graphics* 23 (3), 750–759 (2004).

[21] E.C. Chang, S. Mallat, and C. Yap, "Wavelet foveation", *J. Appl. Comp. Harmonic Analysis* 9 (3), 312–335 (2000).

[22] A.T. Duchowski, "Acuity-matching resolution degradation through wavelet coefficient scaling", *IEEE Trans. on Image Processing* 9 (8), 1437–1440 (2000).

[23] M. Levoy and R. Whitaker, "Gaze-directed volume rendering", *Comp. Grap. (SIGGRAPH'90)* 1, 217–223 (1990).

[24] R. Danforth, A. Duchowski, R. Geist, and E. McAliley, "A platform for gaze-contingent virtual environments", *Smart Graphics (Paper form 2000 AAAI Spring Symp., Technical Report)* SS-00-04, 66–70 (2000).

[25] D. Luebke and C. Erikson, "View-dependant simplification of arbitrary polygonal environments", *Computer Graphics (SIGGRAPH'97)* 1, 199–208 (1997).

[26] H. Murphy and A.T. Duchowski, "Gaze-contingent level of detail", *Eurographics* 1, 1–10 (2001).

[27] S. Wiak and P. Napieralski, "Visualization method of magnetic fields with dynamic particle systems", *Electrotechnical Review* 88 (7B), 56–59 (2012).

[28] C. O'Sullivan and J. Dingliana, "Collisions and perception", *ACM Trans. on Graphics* 20 (3), 151–168 (2001).

[29] C. O'Sullivan, J. Dingliana, and S. Howlett, "Gaze-contingent algorithms for interactive graphics", in eds. J. Hyona, R. Radach, H. Deubel, *The Mind's Eyes: Cognitive and Applied Aspects of Eye Movement Research*, Elsevier Science, Oxford, 2002.

[30] R.J.K. Jacob, A. Girouard, L.M. Hirsfield, M.S. Horn, O. Shaer, E.T. Solovey, and J. Zigelbaum, "Reality-based interaction: a framework for post-WIMP interfaces", *Proc. ACM CHI Conf.* 1, 201–210 (2008).

[31] J. Nielsen, "Noncommand user interfaces", *Communication ACM* 36 (4), 8-99 (1993).

[32] G. Demenko, B. Mobius, and K. Klessa, "Implementation of polish speech synthesis for the BOSS system", *Bull. Pol. Ac.: Tech.* 58 (3), 371–376 (2010).

[33] T. Marciniak, R. Weychan, A. Stankiewicz, and A. Dąbrowski, "A biometric speech signal processing in a system with digital signal processor", *Bull. Pol. Ac.: Tech.* 62 (3), 589–595 (2014).

[34] S. Zhai, C. Morimoto, and S. Ihde, "Manual and gaze input cascaded (MAGIC) pointing", *Proc. ACM SIGCHI Conf. on Human Factors in Computing Systems* 1, 246–253 (1999).

[35] A. Wojciechowski, "Hand's poses recognition as a mean of communication within natural user interfaces", *Bull. Pol. Ac.: Tech.* 60 (2), 331–336 (2012).

[36] G. Glonek and M. Pietruszka, "Natural user interfaces (NUI): review", *J. Applied Comp. Science* 20 (2), 27–46 (2012).

[37] P.J. Durka, R. Kuś, J. Żygierewicz, M. Michalska, P. Milanowski, M. Łabęcki, T. Spustek, D. Laszuk, A. Duszyk, and M. Kruszyński, "User-centered design of brain-computer interfaces: openBCI.pl and BCI appliance", *Bull. Pol. Ac.: Tech.* 60 (3), 427–431 (2012).

[38] C.S. Lin, C.C. Huan, C.N. Chan, M.S. Yeh, and C.C. Chiu, "Design of a computer game using an eye-tracking device for eye's activity rehabilitation", *Optics and Lasers in Engineering* 42 (1), 91–108 (2004).

[39] N. Schneider, E. Barth, P. Bex, and M. Dorr, "An open-source low-cost eye-tracking system for portable real-time and offline tracking", *Proc. 2011 Conf. on Novel Gaze-Controlled Applications* 1, 1–4 (2011).

[40] Y. Ishiguro, A. Mujibiya, T. Miyaki, and J. Rekimoto, "Aided eyes: eye activity sensing for daily life", *Proc. 1st ACM Augmented Human Int. Conf.* 1, 25–31 (2010).

[41] K.H. Tan, D.J. Kriegman, and N. Ahuja, "Appearance-based eye gaze estimation", *Proc. Sixth IEEE Workshop on Applications of Computer Vision* 1, 191–195 (2002).

[42] M.R.M. Mimica and C.H. Morimoto, "A computer vision framework for eye gaze tracking", *Proc. XVI Brazilian Symp. on Comp. Graph. and Image Proc.* 1, 40–412 (2003).

[43] C.H. Morimoto and M.R.M. Mimica, "Eye gaze tracking techniques for interactive applications", *Computer Vision and Image Understanding* 98 (1), 4–24 (2005).

[44] Z. Zhu and Q. Ji, "Eye gaze tracking under natural head movements", *IEEE CS Conf. on Computer Vision and Pattern Recognition* 1, 918–923 (2005).

[45] M. Reale, T. Hung, and L. Yin, "Pointing with the eyes: gaze estimation using a static/active camera system and 3d iris disk model", *IEEE Int. Conf. on Multimedia and Expo (ICME)* 1, 280–285 (2010).

[46] C. Hennessey, B. Noureddin, and P. Lawrence, "A single camera eye-gaze tracking system with free head motion", *Proc. 2006 ACM Symp. on Eye Tracking Research & Applications* 1, 87–94 (2006).

[47] D.W. Hansen and Q. Ji, "In the eye of the beholder: a survey of models for eyes and gaze", *IEEE Trans. on Pattern Analysis and Machine Intelligence* 32 (3), 478–500 (2010).

[48] A. Villanueva and R. Cabeza, "Models for gaze tracking systems", *J. Image and Video Processing* 2007 (3), 4–19 (2007).

[49] D.H. Yoo and M.J. Chung, "Non-intrusive eye gaze estimation without knowledge of eye pose", *Proc. Sixth IEEE Int. Conf. on Automatic Face and Gesture Recognitionn* 1, 785–790 (2004).

[50] Y. Zhou, X. Zhao, S. Zhang, and Y. Zhang, "Design of eye tracking system for real scene", *IEEE Pacific-Asia Workshop*

on Computational Intelligence and Industrial Application 1, 708–711 (2008).

[51] L.E. Sibert and R.J. Jacob, "Evaluation of eye gaze interaction", *Proc. ACM SIGCHI Conf. on Human Factors in Comp. Systems* 1, 281–288 (2000).

[52] K.N. Kim and R.S. Ramakrishna, "Vision-based eye-gaze tracking for human computer interface", *Proc. IEEE Int. Conf. on Systems Man and Cybernetics* 1, 324–329 (1999).

[53] B.L. Nguyen, C. Tijus, F. Jouen, M. Molina, and Y. Chahir, "Eye gaze tracking with free head movements using a single camera", *Proc. 2010 Symp. on Information and Communication Technology* 1, 108–113 (2010).

[54] O.M.C. Williams, A. Blake, and R. Cipolla, "Sparse and semi-supervised visual mapping with the s3Gp", *Proc. IEEE Conf. Comp. Vision and Pattern Rec.* 1, 230–237 (2006).

[55] M. Betke, J. Gips, and P. Fleming, "The camera mouse: visual tracking of body features to provide computer access for people with severe disabilities", *IEEE Neural Systems and Rehabilitation Engineering* 10 (1), 1–10 (2002).

[56] W. Sewell and O. Komogortsev, "Real-time eye gaze tracking with an unmodified commodity webcam employing a neural network", *ACM CHI'10 Extended Abstracts on Human Factors in Computing Systems* 1, 3739–3744 (2010).

[57] T. Liu and C. Pang, "Eye-gaze tracking research based on image processing", *IEEE Congress on Image and Signal Processing* 1, 176–180 (2008).

[58] C.W. Kao, C.W. Yang, Y.W. Chen, K.C. Fan, B.J. Hwang, and C.P. Huang, "Eye gaze tracking based on pattern voting scheme for mobile device", *Proc. IEEE Int. Conf. on Instrumentation, Measurement, Computer, Communication and Control* 1, 337–340 (2011).

[59] J.J. Magee, M.R. Scott, B.N. Waber, and M. Betke, "Eyekeys: a real-time vision interface based on gaze detection from a low-grade video camera", *IEEE Computer Vision and Pattern Recognition Workshop* 1, 159–159 (2004).

[60] H. Yamazoe, A. Utsumi, T. Yonezawa, and S. Abe, "Remote gaze estimation with a single camera based on facial-feature tracking without special calibration actions", *Proc. Symp. Eye Tracking Research and Applications* 1, 140–145 (2008).

[61] T. Ishikawa, S. Baker, I. Matthews, and T. Kanade, "Passive driver gaze tracking with active appearance models", *Proc. 11th World Congress Int. Transport. Syst.* 1, 159–164 (2004).

[62] S.A. Mahmoudi, M. Kierzynka, P. Manneback, and K. Kurowski, „Real-time motion tracking using optical flow on multiple GPUs", *Bull. Pol. Ac.: Tech.* 62 (1), 139–150 (2014).

[63] R. Lienhart and J. Maydt, "An extended set of haar-like features for rapid object detection", *IEEE ICIP 2002* 1, 900-903 (2002).

[64] P.A. Viola and M.J. Jones, "Rapid object detection using a boosted cascade of simple features", *IEEE Computer Society Conf. on Computer Vision and Pattern Recognition* 1, 511–518 (2001).

[65] O. Oguz, "The proportion of the face in younger adults using the thumb rule of Leonardo da Vinci", *J. Surg. Radiol. Anat.* 18 (2), 111–114 (1996).

[66] M. Sezgin and B. Sankur, "Survey over image thresholding techniques and quantitative performance evaluation", *J. Electronic Imaging* 13 (1), 146–165 (2004).

[67] N. Otsu, "A threshold selection method from gray-level histograms", *IEEE Trans. Sys., Man., Cyber.* 9 (1), 62–66 (1979).

[68] M. Taghizadeh and M.R. Mahzoun, "Bidirectional image thresholding algorithm using combined edge detection and p-tile algorithms", *TJMCS* 2 (2), 255–261 (2011).

[69] A. Tomczyk and P.S. Szczepaniak, "Adaptive potential active contours", *Pattern Analysis and Application* 14, 425–440 (2011).