

# Combining Low-Cost Eye Trackers for Dual Monitor Eye Tracking

Sebastian Balthasar<sup>1</sup>, Manuel Martin<sup>2</sup>, Florian van de Camp<sup>2(✉)</sup>,  
Jutta Hild<sup>2</sup>, and Jürgen Beyerer<sup>2</sup>

<sup>1</sup> University of Applied Sciences Karlsruhe, Karlsruhe, Germany  
<sup>2</sup> Fraunhofer IOSB, Karlsruhe, Germany  
florian.vandecamp@iosb.fraunhofer.de

**Abstract.** The increasing use of multiple screens in everyday use creates a demand for multi-monitor eye tracking. Current solutions are complex and for many use cases prohibitively expensive. By combining two, low-cost single monitor eye trackers, we have created a dual monitor eye tracker requiring only minor software modifications from the single monitor version. The results of a user study, which compares the same eye trackers in a single monitor and a dual monitor setup, show that the combined system can accurately estimate the user's gaze across two screens. The presented approach gives insight into low-cost alternatives for multi-monitor eye tracking and provides a basis for more complex setups, incorporating even more screens.

## 1 Introduction and Related Work

Today, typical input devices for computer systems are mouse and keyboard. Over the years, several techniques for user-interface enhancement have been investigated. Eye gaze tracking systems are such an enhancement [ZJ04], which provide input based on the current eye gaze of the user. In situations where manual input is challenging, e.g. due to physical limitations, gaze-based interaction provides a powerful alternative [HGB14]. Multi-monitor setups are becoming the norm and double-monitor systems are widespread in professional environments. Hence, it is worth considering gaze-based interaction also for multi-monitor setups. To our best knowledge, there are only sparse contributions on double monitor eye tracking [CXS+12], whether on how to build such a system or even on what performance such a system could provide. The few commercial solutions are prohibitively expensive for most use cases and require a complex setup.

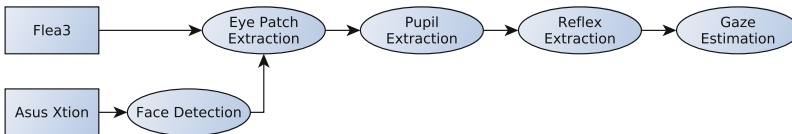
We present an eye tracking system for a horizontal double-monitor setup. The system uses two self-designed remote single-monitor eye tracking devices using the pupil-corneal reflection method to determine the gaze position [QWLY13, GEVC04, HRF14]. We show how the system can be made robust against depth changes and how the occurring interference between the eyetrackers can be compensated without degrading the performance.



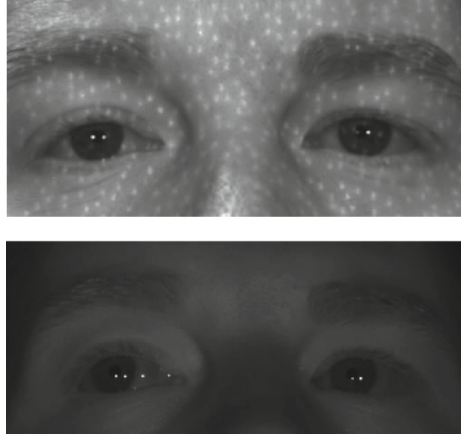
**Fig. 1.** Eye tracker setup with high resolution Flea3 camera, Asus Xtion, IR Led clusters and processing unit.

## 2 Implementation

The eye tracking device consists of a Point Grey Flea3 camera with an IR band pass filter, one Asus Xtion PRO Live camera system, two IR-LED clusters, and one processing unit (Fig. 1). The detection pipeline consists of modules for face detection, pupil detection, corneal reflection detection, and gaze point determination. The basic approach is as follows: The RGB camera of the Asus Xtion is used to detect the face of the user. The bounding box is transferred to the camera image of the Flea3 by means of coordinate transformation between the two calibrated cameras. The rough areas around the eyes are then extracted from the high-resolution Flea3 image using basic facial geometric assumptions. These eye patches are then used to detect the pupil and the two corneal reflections caused by the IR-LED clusters. Afterwards, the pupil position and the corneal reflections are used to calibrate the system and to estimate the gaze (Fig. 2). The IR band pass filter is necessary because the Asus Xtion projects a pattern of IR dots into the scene (Fig. 3 top) which are hard to discern from the corneal reflections. The IR filter causes a darker image but removes any interference with the Asus Xtion (Fig. 3 bottom). In the following, we describe each of the steps in detail.



**Fig. 2.** The eye tracker processing pipeline, using the two image sources in parallel.



**Fig. 3.** IR pattern projected by the Asus Xtion (top), image with IR band pass filter (bottom)

## 2.1 Eye Patch Extraction

The first step in eye tracking is to find the eyes. To do this efficiently, a face detector is used on the color image of Asus Xtion. Based on the face detection the rough positions of the eyes  $(u_a, v_a)$  are determined. In addition to the position of the eyes the depth image of the Asus Xtion is used to extract the distance of the eyes  $(d_a)$  to the camera. Using this information and a device specific stereo calibration between the two cameras ( $C$ : Camera Matrix,  $T$ : Extrinsic Transformation) the eye positions in the Flea3 camera Image  $(u_f, v_f)$  can be computed by the following formula:

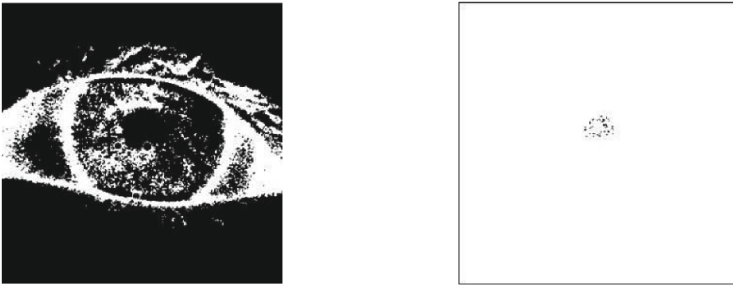
$$\begin{pmatrix} u_f \cdot w \\ v_f \cdot w \\ w \end{pmatrix} = C_f \cdot T \cdot C_a^{-1} \begin{pmatrix} u_a \cdot d_a \\ v_a \cdot d_a \\ d_a \end{pmatrix}, \quad \text{with } C = \begin{pmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{pmatrix} \quad (1)$$

The rough positions of the eyes in the Flea's Image are then used to extract the eyepatches shown in Fig. 6.

## 2.2 Pupil Detection

The detection of the pupil is based on the assumption that it is the darkest part of the image. As we do not work on the whole image at this stage of the processing pipeline, this assumption holds true in all cases we have come across. An implication of this assumption is that the pupil can be extracted from the image using a simple threshold. Depending on the head-pose of the user, height and position as well as eye color the best threshold is not only different from person to person but also changes while using the eye tracker. We therefore use an automatism to constantly adjust the threshold. The basic idea is that a

low threshold will cause many foreground blobs while a high threshold will only leave few foreground pixels. Figure 4 shows these two cases. For every frame, the foreground pixels are analysed and if more than one foreground blob exists after basic cleanup using morphological operators, the current threshold is increased. If there is no foreground blob found, the current threshold is decreased. The constraints of what number of connected foreground pixels constitutes a blob can be configured by providing a minimum width and height of the bounding box of such a blob. A typical image for a good threshold can be seen on the very left of Fig. 5. The next image shows the resulting blob after erosion. The area of the resulting foreground blob is then analysed to find the center and minimal enclosing circle which are then used as the current estimate of the pupil position.



**Fig. 4.** A low threshold will cause many foreground pixels and blobs (left), a high threshold will only leave few foreground pixels (right)



**Fig. 5.** From left to right: Initial extraction of the pupil, erosion to remove scattered foreground pixels, analysis of blob area, center and minimum enclosing circle

### 2.3 Corneal Reflection Detection

The detection of the corneal reflections is similar to the detection of the pupil. However in case of the corneal reflections we assume they are the brightest parts of the image. The small size of the reflections cause an additional challenge as there might be additional reflections caused by tear fluid on the eyes (Fig. 3) which are hard to discern as the reflections just consist of a few bright foreground pixels. For this reason, in a first step, all reflections are detected using

a threshold, in a second step the relative position of the reflections, as well as the position in relation to the position of the pupil, are taken into account to decide which reflections are the actual, direct corneal reflections. While in the case of the pupil a single blob is desirable, for the corneal reflections we need two for a single eye tracker and expect up to four in a dual monitor setup. The number of reflections is therefore used to adjust the threshold as less reflections than required are an indication for a high threshold and too many reflections are an indication for a low threshold.

## 2.4 Calibration and Gaze Estimation

For the estimate of the gaze point on the screen a calibration procedure is necessary. There are a wide variety of mapping functions employed by different research groups. We use a second order polynomial and a 9 point calibration pattern to map the vector between the center of the two corneal reflections and the center of the pupil  $(v_x, v_y)$  to screen coordinates  $(s_x, s_y)$ . The polynomial is defined as:

$$\begin{aligned} s_x &= a_0 + a_1v_x + a_2v_y + a_3v_xv_y + a_4v_x^2 + a_5v_y^2, \\ s_y &= a_6 + a_7v_x + a_8v_y + a_9v_xv_y + a_{10}v_x^2 + a_{11}v_y^2, \end{aligned} \quad (2)$$

The parameters  $a_0 - a_{11}$  are unknowns. Each of the 9 calibration points results in two equations resulting in an over-constrained system with 12 unknowns and 18 equations which is solved using least squares. To make the calibration more robust against outliers we collect, for each calibration point, a number of samples and take the pupil corneal reflection vector with the median length of all measures for calibration. As discussed in [MM05] this gaze mapping approach is robust against all rotations and translations in front of the eye tracker except for back and forth movement. This movement causes a length change of the pupil-corneal reflection vector without a real change of the gaze direction. The result is a deviation in the measurement which quickly degrades the performance of the results if the eye tracker is used without some kind of fixation for the head. A way to improve this is to normalize the length of the pupil-corneal reflection vector to a length that is independent of depth changes. The algorithm already has the distance of the head to the eye tracker from the depth image of the depth camera. The following function can be used to normalize the length of the vector  $v$ :

$$\hat{v} = (m \cdot d_a + b) \cdot v \quad (3)$$

where  $m$  and  $b$  are unknown and need to be calibrated once per device. The calibration procedure is as follows. A test person sits in front of the uncalibrated eye tracker fixates a single point on the screen from different distances and takes some samples at each distance. Solving following equation results in the missing parameters:

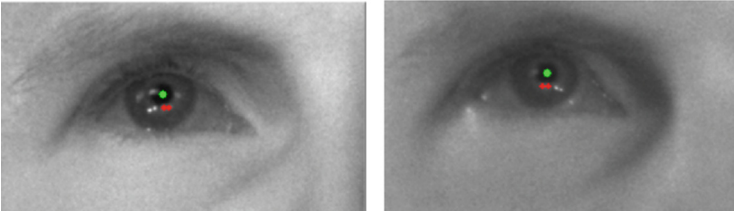
$$m \cdot d_a + b = \frac{1}{\|v\|} \quad (4)$$

where  $d_a$  is the distance of the head to the eye tracker and  $v$  is the vector between pupil and corneal reflection. Solving the equation needs, at least, two

measurements in different distances. After the calibration, the normalized vector will have a length of 1 at the point used for calibration independent of the distance to the eye tracker.

## 2.5 Extensions for the Dual Monitor Setup

To extend the single-monitor eye tracker to a dual-monitor setup, two eye trackers one below each monitor are used. The use of two eye trackers causes four corneal reflections instead of just two in the image of each eye tracker. The placement of the eye trackers, however, causes two distinctive pairs of reflections which can be separated and correlated to either eye tracker, by using their relative location to each other: the eye tracker to the right of the user will cause the pair of reflections on the left (looking at the eye) and vice versa. Figure 6 shows the right eye as seen from the left eye tracker (Fig. 6 left image) and from the right eye tracker (Fig. 6 right image). While this setup allows for eye tracking on two screens, an important question is how the additional corneal reflections affect the detection robustness and therefore, the accuracy of the whole system.

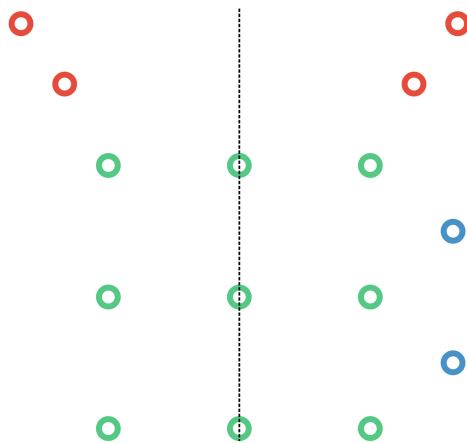


**Fig. 6.** Detected cornea reflections of the left eye tracker (left image) and the right eye tracker (right image)

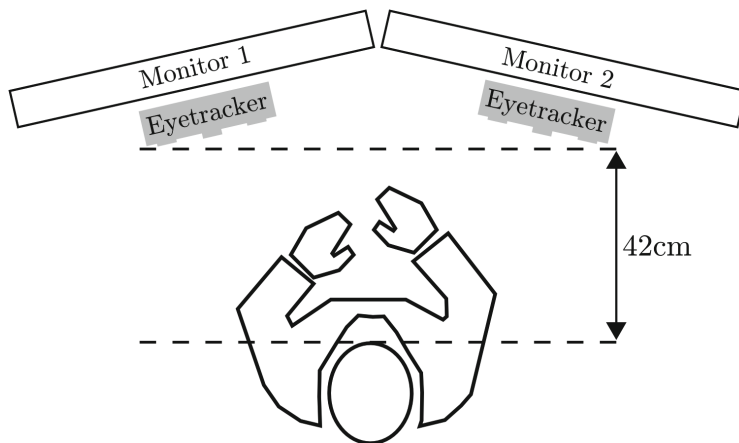
## 3 Evaluation

For evaluation of the eye tracking system, 13 students volunteered in a user study (7 male, 6 female, average age 32.5). None of them wore glasses, one wore contact lenses. The apparatus consisted of two eye tracking devices, each placed in front of a monitor with a resolution of 1920x1200 pixels. The two monitors stood side by side, slightly turned towards the user (Fig. 8). The participants sat in the center. We did not use a chinrest in our evaluation as this would not be accepted by our target users. For evaluation, the participants first calibrated each eye tracker with a 9-point calibration and then had to fixate fifteen points presented on each monitor. The points can be grouped into three sets: two sets followed the design provided by Tobii [Tob11] on their website, one containing points located within  $30^\circ$  of visual angle (main), the other set contained points laying at the upper corners of the monitors (top). The third group consists of two points located at the border where the monitors meet (border) (Fig. 7).

The procedure is as follows: The points are displayed in order from the top left to the bottom right on the left screen and top right to bottom left on the right screen, row by row across screens. The press of a button triggers the collection of evaluation samples and the display of the next point afterwards. This evaluation was done on both screens with both eye trackers running and on the left screen with just the left eye tracker running.



**Fig. 7.** The pattern used for evaluation with the areas top, main and border. The pattern for the right monitor is mirrored on the dashed line.



**Fig. 8.** Setup of two monitors and two eye trackers for dual monitor eye tracking.

### 3.1 Results

The heatmaps in Fig. 9 give a first impression of the accuracy of the double monitor system compared to the single monitor system. The hotspots are comparable showing less consistency for the top and bottom targets in both cases. An important aspect besides accuracy is the extraction quality, which describes the percentage of frames in which both the pupil and the corneal reflections could be found. The eye tracker run at an average of 90 frames per second. Figure 10 shows the percentage of frames for the different systems in which a gaze point could be determined. Here the dual monitor setup slightly outperforms the single monitor setup. The same is true for the accuracy, which is presented in Table 1. These results not only show that the additional corneal reflections do not cause any problems with the detection of reflections, the multi-eye tracker setup even outperforms the single eye tracker in terms of accuracy as well. Our best guess for this is the additional IR light sources, which improve the overall image quality and therefore, aid the accurate detection of the pupil and corneal reflections.

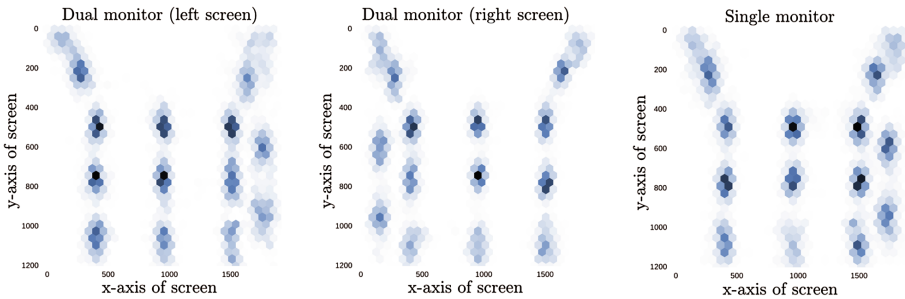


Fig. 9. Heatmaps of the gaze positions over the course of the experiments.

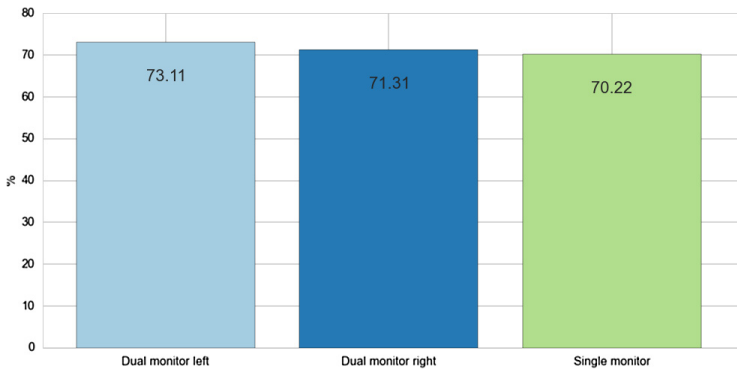


Fig. 10. Extraction quality in percent, with the value representing the percentage of frames in which a gaze point could be determined.



**Table 1.** Average error in degree for the different setups and target sets.

Point set	Main	Top	Border
Dual right	1 : 27° ( $\sigma = 1 : 61^\circ$ )	1 : 63° ( $\sigma = 3 : 07^\circ$ )	1 : 68° ( $\sigma = 1 : 79^\circ$ )
Dual left	1 : 51° ( $\sigma = 2 : 09^\circ$ )	2 : 09° ( $\sigma = 3 : 44^\circ$ )	1 : 82° ( $\sigma = 1 : 77^\circ$ )
Single	1 : 61° ( $\sigma = 3 : 05^\circ$ )	2 : 23° ( $\sigma = 4 : 72^\circ$ )	1 : 72° ( $\sigma = 1 : 41^\circ$ )

## 4 Conclusion and Future Work

We presented a low-cost, remote eye tracker setup capable of accurately detecting the gaze position on a dual monitor setup by using one eye tracker per screen. We have shown how depth information can be used to create a mapping function for estimating screen coordinates robust to distance changes of the user. We have also shown how simple duplication of a single monitor system can lead to a working double monitor system by handling the occurring interferences of the used infrared emitters. Our evaluation shows that the dual monitor setup is equally accurate compared to our single screen system despite the interferences. In the future, we hope to improve this approach to scale to an arbitrary number of screens enabling accurate large scale eye tracking for analysis but also interaction purposes.

**Acknowledgement.** The underlying projects to this article are funded by the WTD 81 of the German Federal Ministry of Defense as well as by Fraunhofer IOSB in-house funding. The authors are responsible for the content of this article.

## References

- [CXS+12] Coddington, J., Xu, J., Sridharan, S., Rege, M., Bailey, R.: Gaze-based image retrieval system using dual eye-trackers. In: Proceedings of the IEEE International Conference on Emerging Signal Processing Applications (ESPA), pp. 37–40 (2012)
- [GEVC04] Goni, S., Echeto, J., Villanueva, A., Cabeza, R.: Robust algorithm for pupil-glint vector detection in a video-oculography eyetracking system. In: Proceedings of the 17th International Conference on Pattern Recognition (ICPR), vol. 4, pp. 941–944 (2004)
- [HGB14] Hild, J., Gill, D., Beyerer, J.: Comparing mouse and magic pointing for moving target acquisition. In: Eye Tracking Research and Applications Symposium (ETRA), pp. 131–134 (2014)
- [HRF14] Hansen, D.W., Roholm, L., Ferreiros, I.G.: Robust glint detection through homography normalization. In: Proceedings of the Symposium on Eye Tracking Research and Applications (ETRA), pp. 91–94 (2014)
- [MM05] Morimoto, C.H., Mimica, M.R.: Eye gaze tracking techniques for interactive applications. *Comput. Vis. Image Underst.* **98**(1), 4–24 (2005). special Issue on Eye Detection and Tracking

- [QWLY13] Qin, H., Wang, X., Liang, M., Yan, W.: A novel pupil detection algorithm for infrared eye image. In: Proceedings of the IEEE International Conference on Signal Processing, Communication and Computing (ICSPCC), pp. 1–5 (2013)
- [Tob11] Tobii Technology AB, “Accuracy and precision test method for remote eye trackers,” Tobii, Technical Report, 02 2011
- [ZJ04] Zhu, Z., Ji, Q.: Eye and gaze tracking for interactive graphic display. *Mach. Vis. Appl.* **15**(3), 139–148 (2004)