An open-source low-cost eye-tracking system for portable real-time and offline tracking

Nicolas Schneider
Institute for Neuro- and Bioinformatics
University of Lübeck
Ratzeburger Allee 160
23562 Lübeck, Germany
schneider@inb.uni-luebeck.de

Peter Bex
Schepens Eye Research Institute
Harvard Medical School
20 Staniford Street
Boston, MA 02114, USA
peter.bex@schepens.harvard.edu

Erhardt Barth
Institute for Neuro- and Bioinformatics
University of Lübeck
Ratzeburger Allee 160
23562 Lübeck, Germany
barth@inb.uni-luebeck.de

Michael Dorr
Schepens Eye Research Institute
Harvard Medical School
20 Staniford Street
Boston, MA 02114, USA
michael.dorr@schepens.harvard.edu

ABSTRACT

Open-source eye trackers have the potential to bring gaze-controlled applications to a wider audience or even the mass market due to their low cost, and their flexibility and tracking quality are continuously improving. We here present a new portable low-cost head-mounted eye-tracking system based on the open-source ITU Gaze Tracker software. The setup consists of a pair of self-built tracking glasses with attached cameras for eye and scene recording. The software was significantly extended and functionality was added for calibration in space, scene recording, synchronization for eye and scene videos, and offline tracking. Results of indoor and outdoor evaluations show that our system provides a useful tool for low-cost portable eye tracking; the software is publicly available.

Categories and Subject Descriptors

H.5.2 [**Information Interfaces and Presentation**]: User Interfaces – Interaction styles; J.4 [**Computer Applications**]: Social and Behavioral Sciences – Psychology; I.4 [**Computing Methodologies**]: Image Processing and Computer Vision – Miscellaneous

Keywords

low-cost mobile eye-tracking, gaze interaction, off-the-shelf components, open source

NGCA '11, May 26-27 2011, Karlskrona, Sweden

1. INTRODUCTION

Eye tracking plays an increasingly important role in human-computer interaction. Gaze position as a sole or an additional input modality has been used in a wide range of applications, including typing [6, 13], games [4, 9], video conferencing [12], and generally hands-free control [7]. This development was partially driven by ever more robust and accurate eye-tracking systems that also have become less intrusive.

However, for a wide-spread adoption of gaze-controlled applications, commercial eye trackers are still prohibitively expensive at a typical cost of several tens of thousands of dollars. Furthermore, both software and hardware design are closed source, so that extensions or modifications are impossible or incur additional cost. As a low-cost alternative, open-source trackers have become available that use cheap commodity hardware such as webcams, and the source code can be flexibly modified for specific needs. These systems might not have reached the maturity of their commercial counterparts yet, but development progresses rapidly and tracking has become good enough for all but the most demanding applications.

Several open-source systems have been presented in recent years. Babcock and Pelz showed how to build low-cost hardware [2] and Li et al. presented openEyes, a more evolved tracking system using different tracking algorithms [5]. Zieliński developed a simple webcam-based tracking software [14], and San Agustin et al. published the ITU Gaze Tracker [10]. The ITU Gaze Tracker is a comprehensive eye-tracking framework for remote and head-mounted tracking that can be used with almost any webcam, and is continually maintained and extended by an active community.

Most eye-tracking systems measure gaze in screen coordinates, which is useful when well-defined stimuli are presented on computer screens, e.g. in psychophysics. However, other applications, such as market research, can bene-

fit from portable systems where gaze is recorded in 3D space (for example for better product placement in supermarkets); with the advent of gesture-based techniques, user interaction might generally be freed from the constraints of a confined (and wired-up) workspace. Our main motivation is the development of a portable eye-tracking system to study attentional processes in natural and potentially safety-critical tasks such as driving.

Until now, only a few systems have offered an integrated solution with a scene camera that records the subject's field of view. As with stationary systems, commercial portable trackers [3] are expensive, but existing low-cost approaches require advanced technical and electronical knowledge and lack functionality, e.g. tracking is performed either offline (without real-time feedback) [8] or online only [5]. We here present modifications to the ITU Gaze Tracker in order to use it for mobile applications, with off-the-shelf hardware components that can be assembled with a minimum of technical knowledge and effort.

The following requirements were set: i) mobile and usable both indoors and outdoors; ii) integrated scene video that shows the subject's field of view; iii) eye and scene video should be recorded for later offline-analysis at full temporal resolution, with a real-time feedback dependent on available computational resources; iv) calibration in 3D space.

2. HARDWARE

The system should offer a high-quality but low-cost solution for mobile eye tracking. Since the software is open source, the hardware is the only cost factor. The necessary parts are two cameras with lenses, spectacle frames, wire, cable tie, USB cables, and (optionally) an infrared LED to illuminate the eye (Fig. 1). The most expensive parts are the cameras. Depending on resolution, framerate, weight and size the price can vary widely. We optimized our system for weight and size so that cameras cost around 270 EUR each (high spatial, low temporal resolution colour camera for the scene image, relatively low spatial, high temporal resolution grayscale camera for the eye image); in principle, any cheap webcam could be used as well. All other parts range inbetween 1–10 EUR each and together cost not more than 50 EUR. The glasses can be assembled in half a day, a construction manual with an illustrated step-by-step guide and advice concerning radiation intensity of the LED and camera/diode placement is available online at `http://gazetrackinglib.svn.sourceforge.net/viewvc/gazetrackinglib/H.E.A.D/`.

For mobile applications, a laptop is necessary. In order to utilize the full bandwidth of both cameras, two separate USB controllers are needed, which unfortunately is an uncommon feature in (cheaper) laptops. Commercial systems are now available that use a much less bulky embedded device instead of a laptop, but it should be noted that these are not only significantly more expensive, but also offer less spatio-temporal resolution than the system presented here.

One problem that is intrinsic to this hardware setup is the parallax error between the optical axes of scene camera and recorded eye. The glasses are designed in order to be as lightweight and balanced as possible to increase comfort. Moving the scene camera closer to the recorded eye would reduce the parallax error, but might require a more elaborate fixation of the glasses to the head. Even though the glasses are not slip-resistant and thus minor slippage might occur, we could not observe a noticeable impact on the tracking quality over time.

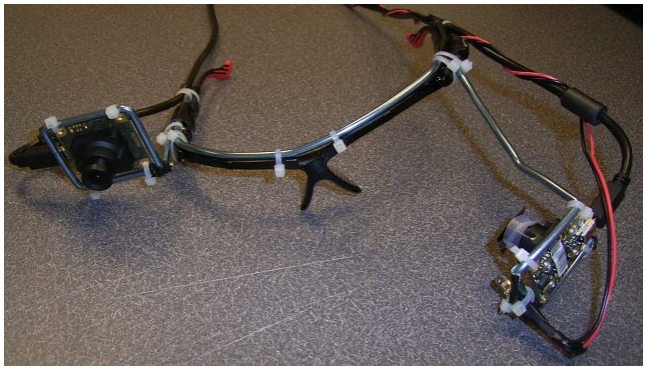

Figure 1: The low-cost prototype with scene camera (left) and eye camera and IR LED (right).

3. SOFTWARE

Video-oculographic eye trackers estimate gaze based on the position of eye features, usually the pupil and an additional set of corneal reflections. In order to calculate the intersection of gaze with a plane in space, e.g. the computer screen, eye feature positions in the eye image need to be set in relation to screen coordinates. Therefore, a calibration routine is typically performed before or after the tracking session to obtain coefficients which can be used to compute gaze position on the screen. The ITU Gaze Tracker uses an overdetermined linear equation system to calculate these coefficients after nine-, twelf-, or sixteen-point calibration.

3.1 Modifications and Extension

The ITU Gaze Tracker provides remote and head-mounted eye tracking for screen-based applications. The system can be used with most webcams, the source code is available online, and an active user community is contributing to the project. Tracking is based on the dark pupil technique and an additional illumination with an infrared LED can be used in the remote tracking mode. Gaze positions are output in a log file or over network and can be forwarded to and used in other applications, for example in eye-typing programs [11].

We implemented several modifications and extensions in order to allow portable eye tracking. Detaching the system from computer screens made it necessary to integrate a scene camera that records the subject's field of view, and a novel calibration routine based on points in the scene image was implemented. For later offline analysis, video streams from both eye and scene camera needed to be stored to disk, and measures had to be taken to ensure synchronicity of these streams even for long recording sessions.

3.1.1 Combined Online and Offline Tracking

For the offline functionality, it was necessary to add video recording and replay, and calibration and tracking on previously recorded data. The video export itself is a trivial function, but the synchronization of the two video files is more challenging since running both cameras (and the laptop) at their limit led to occasionally dropped frames (see below). During calibration, position and timing of all calibration points are saved to disk with their corresponding eye

frames, so that the system can be calibrated offline based on recorded videos without any manual intervention.

The main remaining difference between online and offline tracking then lies in the time constraints during tracking. All frames from the eye camera are buffered in a queue and encoded to disk in a separate background thread. However, the more critical and time-consuming feature extraction, e.g. pupil and CR detection, is always only performed on the most recent image in the queue at a potentially lower rate. During offline tracking, each frame of the eye video can be processed without real-time constraints, potentially with more complex algorithms to determine thresholds for feature detection. The exact reconstruction of calibration sequence and eye video can also be used to compare different tracking algorithms under identical conditions.

3.1.2 *Scene Video Integration*

A scene camera was integrated so that the system can be calibrated based on the scene video and gaze positions can be overlaid onto the subject's field of view. Adding a second camera to the setup required a synchronization mechanism that is not offered natively by the DirectShow framework the ITU Tracker uses. We thus implemented a new interface to the cameras, which currently requires (some) additional source code for each new camera type.

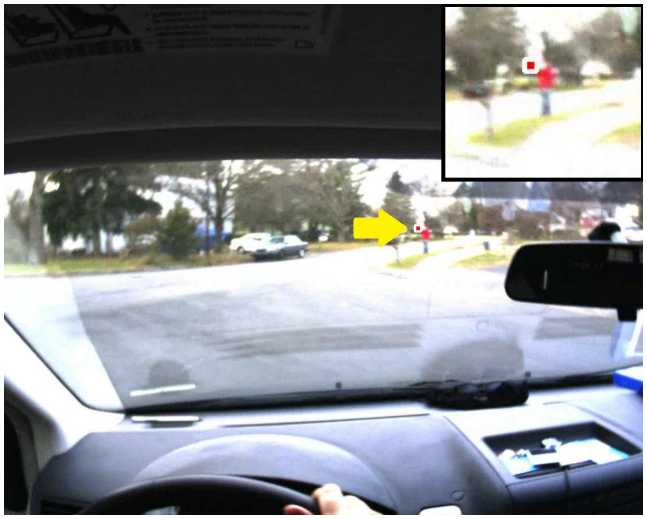

Figure 2: Stillshot from gaze overlay on scene video. Top right: enlarged area with gaze cursor on target.

The new communication layer is based on several concurrent threads that handle frame grabbing and a performance-dependent processing. While recording, it buffers the frames in queues, encodes them into video files, and collects and saves additional data for later synchronization of the two videos. During offline tracking the calculated gaze points will be assigned to their corresponding scene frames with tolerance against dropped frames, transfer errors, and rounded frame rates during playback.

3.1.3 *Calibration in Space*

For mobile tracking, calibration should be possible at varying distances and in 3D space. As a calibration target, we use an A4-sized black and white bullseye that is unnatural and has high contrast, so that it can be reliably detected in the scene image using template matching. We match the template at different sizes and on several scales of a Gaussian multiresolution pyramid to achieve a smooth range of calibration distances from approx. 15 cm to 4 m. This target can be placed anywhere in the scene and the subject then simply needs to fixate the target at different head orientations; for an optimal distribution of calibration points in the scene, an experimenter can also move the target around manually.

Subject	Valid. error	s. d.	Accuracy	Fix. dist.
A	0.31°	0.17°	1.28°	3 m
B	0.18°	0.14°	0.86°	2 m
C	0.18°	0.15°	1.2°	3 m

Table 1: Results of indoor accuracy evaluation.

4. EVALUATION

The system was evaluated for both indoor and outdoor use. We tested the accuracy of the system and its performance under "real" conditions, and how well the requirements set out at the beginning were actually met.

For indoor tests, three subjects were placed on a chair and sequentially fixated points on a 10×7 grid on a wall at a fixation distance of 2 m or 3 m. Table 1 shows the results of the accuracy evaluation which are between 0.86° and 1.28°. This is not as good as the accuracy of commercial systems (which usually claim an accuracy between 0.25° and 1° visual angle), but roughly comparable to other open-source, low-cost systems (e.g. openEyes: $0.68° - 1.37°$).

Spatio-temporal resolution was constrained by USB 2.0 bandwidth and limits were determined with a high-end desktop computer (with two USB controllers) and a laptop. The scene video resolution was set to 1280×1024 pixels; for the eye video, an ROI of 450×340 pixels was selected around the pupil (full eye camera resolution 752×480). On the desktop computer, up to 150 Hz eye video and 10 Hz scene video could reliably be recorded. These numbers were lower for the laptop with 40 and 7 Hz, respectively; higher rates could be obtained by sacrificing scene video resolution.

Outdoor tests in the car mainly evaluated the influence of varying illumination and "real" conditions on the tracking quality. As expected, several adjustments had to be made to calibrate and track successfully outdoors. For example, the stark white parts of the bullseye calibration target led to blooming artefacts in bright sunshine and a failing template matching, so that the target was replaced with a black and gray bullseye. A more severe problem arose with under- and overexposure of the scene video. Extreme illumination changes, e.g. going through an underpass, led to short unusable segments in the scene video despite automatic exposure adjustments. More importantly, the high contrast between outdoor scene and parts of the car interior that were visible in the scene image (see e.g. dashboard, ceiling in Fig. 2) prevented the whole dynamic range of the camera from being used for the relevant outdoor scene only. Feature extraction thresholds were also affected by unstable light conditions. This effect could be reduced by simply placing a small piece of exposed film as an IR-pass filter in front of the eye camera, which also reduced other reflections on the cornea. Because of the offline functionality, tracking could also be achieved by manually adjusting thresholds even for recording segments

where pupil and CR were temporarily lost completely.

5. CONCLUSIONS AND FUTURE WORK

Here we described our modifications to an existing open-source eye tracker to use it as a low-cost mobile eye-tracking system with combined online and offline tracking. A scene camera was integrated and an automatic calibration routine in space based on multi-scale template matching was implemented so that the system can be used in indoor and outdoor scenarios. An initial evaluation shows that the design requirements were met and tracking accuracy is comparable to other open-source systems.

Fully functional offline tracking may seem trivial in principle, but has some pitfalls in practice. In particular, synchronization of scene and eye camera previously has been achieved only by using (expensive) hardware-locked cameras (with identical frame rates) [5] or by workarounds such as manual alignment to an event visible in both videos such as a flash [2, 8]. For longer recordings, minor camera clock inaccuracies or occasionally dropped frames may lead to accumulating offsets, and we have solved this problem by storing frame-exact time stamps to a separate synchronization file. Offline functionality is highly useful for the comparative evaluation of different tracking algorithms or parameters because these can be tested under exactly identical conditions. Furthermore, to our knowledge, we have implemented the first portable tracker that makes optimal use of available resources by providing both online measurements – at potentially low temporal resolution – and high-resolution measurements after offline postprocessing.

Clearly, several issues remain for future work. In order to better utilize the full dynamic range of the scene video in automotive applications, the automatic exposure adjustments of the camera should be restricted to the outdoor (windshield) part of the scene. Because e.g. the dashboard's position in the scene video depends on head position, a suppression mask would need to actively track these regions. For video synchronization to work, it was necessary to use camera SDK-specific functionality instead of DirectShow. This currently limits the range of supported cameras, but adding new cameras (e.g. with USB 3.0 interface) should typically involve only small modifications to the source code.

Despite these limitations, we hope that our work will be already of use to the eye-tracking community. All our modifications and extensions are publicly available in the Gaze Tracker repository [1], as well as an assembly manual that describes the exact hardware setup. A video of the system can be found at `http://www.youtube.com/watch?v=zreWEScirlI`. Changes were originally made only against version 1.6 of the ITU Gaze Tracker, but will soon be merged into the current version, which had not been published yet when our project started. In initial tests, version 2 of the ITU Gaze Tracker delivers an improved accuracy of 0.75° visual angle (depending on application and setup, up to 0.3°–0.6°). The great advantage of collaborative open-source software is that these improvements will directly benefit our system, as well as low-cost gaze-controlled applications in general.

6. ACKNOWLEDGMENTS

Supported by a DAAD scholarship to N.S. and NIH grants EY018664 and EY019281.

7. REFERENCES

[1] ITU Gaze Tracker development page. http://www.gazegroup.org/develop/.

[2] J. S. Babcock and J. B. Pelz. Building a lightweight eyetracking headgear. In *ETRA '04: Proceedings of the 2004 Symposium on Eye tracking Research & Applications*, pages 109–114, New York, NY, USA, 2004. ACM.

[3] A. Bulling and H. Gellersen. Toward mobile eye-based human-computer interaction. *IEEE Pervasive Computing*, 9:8–12, 2010.

[4] M. Dorr, M. Böhme, T. Martinetz, and E. Barth. Gaze beats mouse: a case study. In *The 3rd Conference on Communication by Gaze Interaction - COGAIN 2007, Leicester, UK*, pages 16–19, 2007.

[5] D. Li, J. Babcock, and D. J. Parkhurst. openEyes: A low-cost head-mounted eye-tracking solution. In *ETRA '06: Proceedings of the 2006 Eye Tracking Research & Application Symposium*, pages 95–100, New York, NY, USA, 2006. ACM.

[6] P. Majaranta and K.-J. Räihä. Twenty years of eye typing: systems and design issues. In *Proceedings of the 2002 symposium on Eye tracking research & applications*, ETRA '02, pages 15–22, New York, NY, USA, 2002. ACM.

[7] J. C. Mateo, J. San Agustin, and J. P. Hansen. Gaze beats mouse: hands-free selection by combining gaze and EMG. In *CHI '08 extended abstracts on Human factors in computing systems*, CHI EA '08, pages 3039–3044, New York, NY, USA, 2008. ACM.

[8] W. J. Ryan, A. T. Duchowski, E. A. Vincent, and D. Battisto. Match-moving for area-based analysis of eye movements in natural tasks. In *ETRA '10: Proceedings of the 2010 Symposium on Eye-Tracking Research & Applications*, pages 235–242, New York, NY, USA, 2010. ACM.

[9] J. San Agustin, J. P. Hansen, A. Villanueva, and J. C. Mateo. Evaluation of the potential of gaze input for game interaction. *PsychNology*, 7(2):213–236, 2009.

[10] J. San Agustin, H. Skovsgaard, J. P. Hansen, and D. W. Hansen. Low-cost gaze interaction: ready to deliver the promises. In *CHI '09: Proceedings of the 27th international conference extended abstracts on Human factors in computing systems*, pages 4453–4458, New York, NY, USA, 2009. ACM.

[11] J. San Agustin, H. Skovsgaard, E. Mollenbach, M. Barret, M. Tall, D. W. Hansen, and J. P. Hansen. Evaluation of a low-cost open-source gaze tracker. In *ETRA '10: Proceedings of the 2010 Symposium on Eye-Tracking Research & Applications*, pages 77–80, New York, NY, USA, 2010. ACM.

[12] R. Vertegaal. The GAZE Groupware System: Mediating Joint Attention in Multiparty Communication and Collaboration. In *Human Factors in Computer Systems, CHI 1999 Proceedings*, pages 294–301, 1999.

[13] D. J. Ward and D. J. C. MacKay. Fast hands-free writing by gaze direction. *Nature*, 418(6900):838, 2002.

[14] P. Zieliński. Opengazer: open-source gaze tracker for ordinary webcams (Piotr Zieliński), 2007. http://www.inference.phy.cam.ac.uk/opengazer/.