

# OGTA: Open Gaze Tracker and Analyzer

## A Remote Low Cost System Based on Off-the-Shelf Components and Open Source Modular Software

Daniele Cipriani ([daniele.cipriani@di.uniroma1.it](mailto:daniele.cipriani@di.uniroma1.it))<sup>1</sup>, Daniele Caligiore ([daniele.caligiore@istc.cnr.it](mailto:daniele.caligiore@istc.cnr.it))<sup>2</sup>, Gianluca Baldassarre ([gianluca.baldassarre@istc.cnr.it](mailto:gianluca.baldassarre@istc.cnr.it))<sup>2</sup>, Constantin A. Rothkopf ([rothkopf@fias.uni-frankfurt.de](mailto:rothkopf@fias.uni-frankfurt.de))<sup>3</sup>, Jochen Triesch ([triesch@fias.uni-frankfurt.de](mailto:triesch@fias.uni-frankfurt.de))<sup>3</sup>, Maria De Marsico ([demarsico@di.uniroma1.it](mailto:demarsico@di.uniroma1.it))<sup>1</sup>

<sup>1</sup>Dipartimento di Informatica, Università di Roma “Sapienza” of Rome, Italy

<sup>2</sup>Laboratory of Computational Embodied Neuroscience, Istituto di Scienze e Tecnologie della Cognizione, Consiglio Nazionale delle Ricerche (LOCEN-ISTC-CNR) of Rome, Italy

<sup>3</sup>Frankfurt Institute for Advanced Studies, Goethe University of Frankfurt, Germany

### Abstract

Several academic and commercial eye tracking systems have evolved to the point that they can operate without contact with the user. In addition, they also permit free head movement (within reasonable limits) without losing tracking and maintaining a good accuracy (errors below 1 degree). However, there are still several aspects which require further improvement before these systems can be extensively used. These include price, accuracy, robustness, and ease of set-up and use. This work proposes a preliminary version of a remote eye tracking system which starts to deal with some of those critical points. To drastically reduce the costs, the system has been built by assembling low cost off-the-shelf components, and the cross-platform software has been developed based on an open source philosophy. Second, the accuracy in gaze detection has been improved through the Starburst algorithm. Last and importantly, the *plug-in organization of the software architecture*, which crucially distinguishes the proposed system from similar ones previously proposed in literature. This facilitates the addition of dedicated software modules designed to improve specific features according to the particular application at hand. Here we present the architecture of the system and preliminary results on the functioning and accuracy of the system.

### Method

In this section we describe the hardware and software components of the proposed eye tracker. The hardware is formed by low cost off-the-shelf components: a camera (Thorlabs DCC1545M), positioned under the monitor, which mounts a lens Vivotek 8-16mm F1.6, and one infrared light (CCTV IR) positioned next to the camera (the total cost of the system is about 450€).

The gaze tracking application is an open-source software, developed in C++ language, which uses the Qt Library (<http://qt-project.org/> last accessed on February 26, 2013), the OpenEyes Toolkit (<http://thirtysixthspan.com/openEyes/> last accessed on February 26, 2013) and the FaceTracker API (<http://jsaragih.org/> last accessed on February 26, 2013). The proposed software has a composite, plugin-based architecture based on the Qt Plugin API, that facilitates the design of rich, flexible and easy to maintain desktop applications. Moreover, Qt Library is a cross-platform application and user interface framework allowing to write code once and to target multiple platforms.

As shown in the Fig. 1, the composite architecture is partitioned into a number of semi-independent, loosely coupled components, that can be easily integrated in the overall application, can evolve independently, and can be possibly substituted. Each module, shown within the dashed square, encapsulates a portion of the functionality of the application and typically addresses a set of related activities. The Face Tracker Module implements the Haar-Cascade Classifier and the Active Shape Model, which are exploited respectively for face detection and tracking. The Eye Tracker Module detects pupil center and glint within the eye regions. The

Mapping module implements a second order polynomial mapping to bind the eye vector to the corresponding screen vector. When a module is needed to perform its specific task, it is loaded by the QPluginLoader and is interfaced to the application. The Graphical User Interface (GUI) of the application shows a visual feedback to the user concerning the tracking operations, the control widgets for parametrization, and the information output, all suitably organized into child dockable window widgets. The adopted software organization promotes the reuse of code and the clean separation of concerns, by applying design patterns, e.g. the Model View design pattern to separate data from views in widgets that handle data sets.

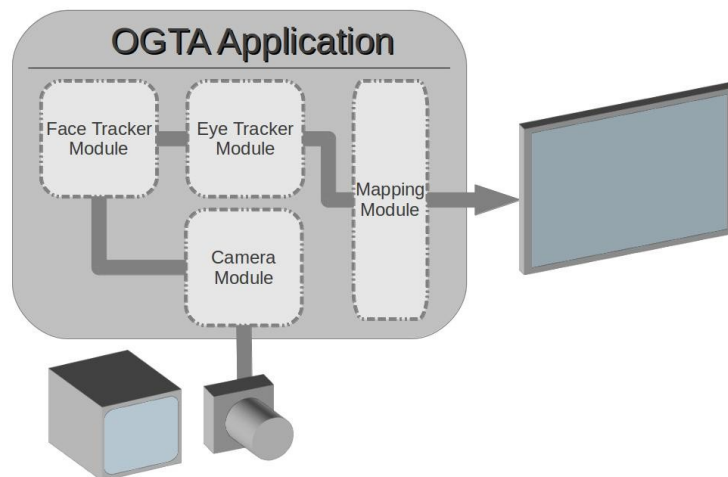


Figure 1: The hardware and software components of the OGTA system.

**OGTA functioning.** Regarding the adopted tracking strategy, OGTA platform is a PCCR (Pupil Center Corneal Reflection). The developed software first finds the pupil and corneal reflections centers for the gaze estimation. Then the Haar-Cascade detection [Bradski, G. et al (2008). Learning OpenCV: Computer Vision with the OpenCV Library. O'Reilly] and the Active Shape Model tracking of face [Saragih, J. et al (2011). Deformable model fitting by regularized landmark mean-shift. International Journal of Computer Vision, 91, p. 200], pupil and glints are detected by double thresholding the image and applying the blob analysis. The center of the pupil and the center of the glint correspond to the center of gravity of the respective blobs. The pupil and glint centers obtained in this way are two good starting positions to apply the Starburst algorithm [Li, D. et al (2006). A single-camera remote eye tracker. Perception and Interactive Technologies, 4021 of Lecture Notes in Artificial Intelligence, p. 208], which further improves the accuracy of the detection and reduces the jitter of the target indicating the gaze point.

## Results

In order to assess the validity of the proposed system we tested the gaze tracker on 10 participants. The experiment consisted in executing a calibration procedure to calculate the accuracy with an angle of view in degrees. While calibrating, the user was sitting at a desk with his head still at 60 cm from the screen. The calibration procedure shows 9 calibration points in a random order. When a calibration point is shown, it remains blocked for a while to prevent the anticipatory behavior of the eye, and then it shrinks to collect data. At the end of the calibration procedure the resulting accuracy is shown in a child window. Only a subset of points can be possibly re-calibrated if needed, without restarting from scratch. The accuracy values obtained by the subjects in the experiment were in between 0.5 and 1 degree of view angle. The high frame rate reached by exploiting the ROI camera mode is over 40 processed frame per second.

## Conclusion

This work presented and evaluated a low cost, open source gaze tracker. The system offers a high flexibility thanks to the modularity of software and to the off-the-shelf, quite inexpensive hardware components, which can include different cameras and light sources. The resulting gaze tracking system is accurate enough to be employed for research, e.g., in Psychology, Human Computer Interaction and in the Biomedical Field. At present, the system implements a partial head compensation, which allows small movements [Duchowski, A. T. (2007). Eye tracking methodology: Theory and Practise. Springer]. Our main objective in the next future is to improve this aspect. To this aim, we are working on an updated version of our system which employs the face detection features of the Active Shape Model to deal with free head motion. This will allow to obtain a more usable eye tracker, that will be able to compensate the error introduced in the gaze estimation by the head movement.

### Acknowledgements

This work was funded by the European Commission 7<sup>th</sup> Framework Programme (FP7/2007-2013), *Challenge 2 – Cognitive Systems, Interactions, Robotics, Grant Agreement No. ICT-IP-231722, Project IM-CLeVeR – Intrinsically Motivated Cumulative Learning Versatile Robots*. This work was also partially supported by Italian MIUR under the FIRB project "VISEL - E-learning, deafness, written language: a bridge of letters and signs towards knowledge society".