

# LOOK NO HANDS

Connecting eye motion to computer control



**Jonathan Chetwynd**

Guest Writer

## The Picameral eye-tracker

**SKILL LEVEL : ADVANCED**

The Picameral eye-tracker is hosted at <http://www.gnote.org> and is part of a larger project studying how we understand complex systems, my current visualisation is <http://www.peepo.com>.

The Raspberry Pi computer enables, encourages and empowers the individual to explore and conduct independent research; so why make an eye tracker? The eye is a challenge as it is easily the fastest human organ, tracking its motion produces masses of data, and there are useful rules or heuristics to be derived from the various motions including saccades, glissades and fixations, or jerk, track and dwell.

Early eye-tracking research took place in a laboratory in a strictly controlled environment, perhaps using a forehead and chin rest to study reading skills or more recently web usability. It has only very recently become possible to make a cheap portable eyetracker. The total cost for this project should be ~£100 which includes 2x RaspberryPi with cameras, SD cards, cases, leads, etc. though you may already have some of this kit.

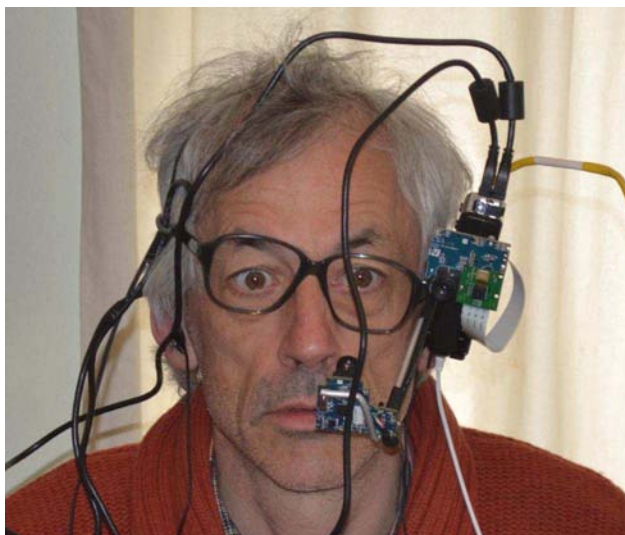
A portable eyetracker greatly broadens possible applications, perhaps recording and modelling social interactions at a party? whether side and rear-view mirrors are being observed regularly whilst driving? Is a soldier in the field using appropriate IED search

techniques? You can easily imagine others.

There are some excellent resources available, the approach used builds from the excellent work of John Paulin Hansen and the team at ITU in Denmark who developed the Windows based GazeTracker; and the book Eye Tracking: by Holmqvist & Nyström 2011 provides a thorough introduction, without being overly academic.



Creating an eye-tracker is not itself a large problem, the difficulty is in considering the purposes for which you wish to use your device, and hence the peripheral devices and software required. For instance an 850ms dwell could be set to take a high quality still image using a second camera, here a cut down PS3 eyecam is connected with an RPi viewcam to a single RPi. The setup was tested and found to capture 200 images mostly of my choice, while cycling from Vauxhall to Kingston in about 45 minutes.



An earlier prototype had some load balancing problems that were improved in the final design.

Cameras have been improving fast, and the Raspberry Pi camera has many excellent qualities which we rely on. It can easily be adapted as a macro, which happens to focus at a useful distance with good quality video images. When mounted perpendicular to the camera board, the sensor is only a small obstruction and James Hughes implementation of Region Of Interest (ROI) enables digital cropping.



For more serious research, frame rates faster than 30 are desirable, and it is hoped that for formats considerably smaller than 1080HD, 60-90fps may be available shortly. The sensor is capable of 500fps for some formats, and this would allow one to estimate accelerations as well as velocities and locations.

As a project for one person, implementing an eyetracker requires a variety of skills, however the tasks might be split amongst a class of enthusiastic and bright teenagers!

## Outline

The physical construction is relatively simple, and evident, in addition you will need: a database to store the data, a graphical tool for calibration, communications including a web server for ethernet and uart for serial port, perhaps audio feedback, and possibly a portable display.

Portable displays are ubiquitous, even HD is common, but HDMI inputs are rare to non-existent. If your project needs a portable interface the Kindle is cheap, and can use ethernet over usb to communicate.

Assuming you intend to visualise any results, which is not a requirement, I recommend html5 and NodeJS, a simple and powerful web server, with an up-to-date version of a web browser such as Mozilla Firefox, Chrome, Safari or Opera, or possibly IE11.

It may seem counter-intuitive, but whilst the RPi is capable of serving the eyetracking data, including video, it cannot currently run an html5 browser efficiently so you may need another box. I use a 5 year old mac-mini running Debian Wheezy for convenience.

During construction and development, communication relies on ssh over ethernet. Once we need to be portable, start applications in Screen and detach, to leave our application running while we unplug ethernet, and with care, later reconnect and continue; magic indeed!

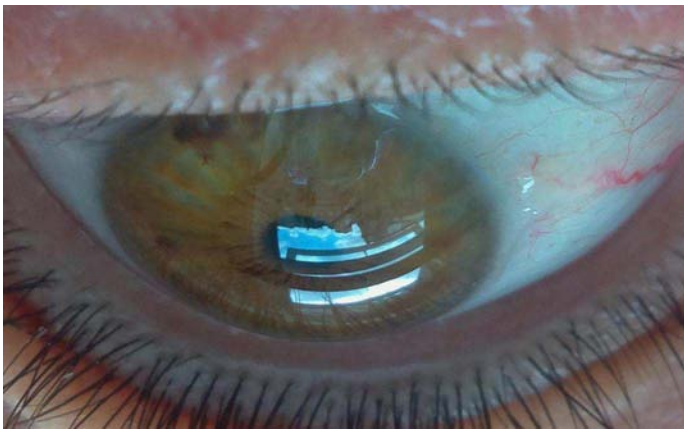


## Software

In addition to various easy to use applications such as raspivid, an acquaintance with a few software languages is helpful to those wishing to develop their own application; some javascript, html5 and svg will be useful and perhaps also C and bash.

When bug hunting, and in case of difficulties, create a reduced test case to test each part independently.

The actual iris tracking software is simple enough, we assume that the iris covers about 30% of the image, and is the darkest part of the image, similar to that illustrated:



Current camera sensors adjust very fast to changes in illumination. A first run assesses the overall tone of each frame. In tests, the upper bound for this darkest 30% ranged from 27 to 200 out of a total possible range of 0-255, one would need to check each frame to evaluate further. We then calculate the centre of this darkness, our supposed iris.

So a few lines of code need to be added to the excellent rpi-mmal camera software from tasanakorn based on his mmal studies:

<http://github.com/raspberrypi/userland/tree/master/interface/mmal>

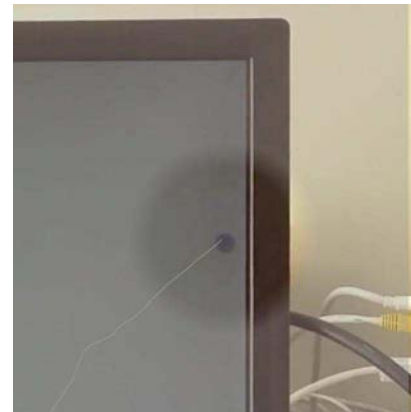
Then the Raspberry Pi with eyecam can calculate and display the output in real time at 30fps.

This x,y data is stored in our Redis database for later visualisation, though it may also be sent via ethernet to a website, for realtime interactive gameplay, or via serial uart to our second Raspberry Pi with video camera to mark the user's gaze in real-time.

Uart over serial is used to sync the eyecam and viewcam, as a clapperboard is used to sync audio and video. Drift is bound to arise, there is no editor, nor cuts, and it is almost essential to have a video timestamp, though one is not currently available.

## Callibration

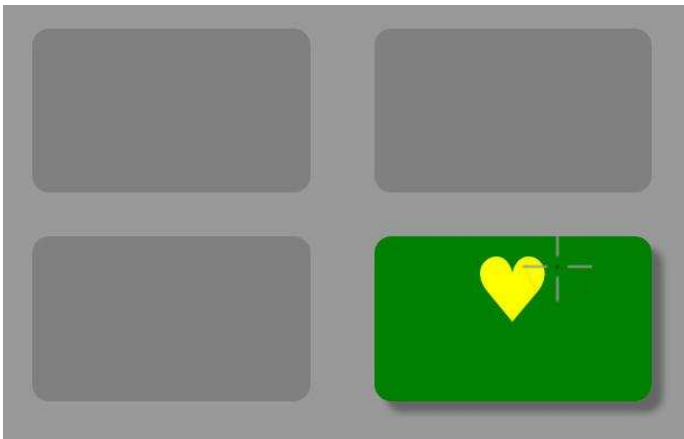
You will need to callibrate the eyecam's eyetracking data with the Raspberry Pi viewcam's sensor's Field Of View. That is, we need to align them in respect of time, and to estimate the user's gaze position on the video frame.



The user should be directly in front of a 1080HD screen which should cover perhaps 20% of their total visual field. The user follows a moving point with their eyes, we use a 4x4 grid which covers the whole screen and is ordered randomly, to prevent the user guessing the next point. This takes about a minute to complete, for the moment assume callibration is complete and successful.

Ready to play my eyetracking game 'Take the Box' and hone your skills.

When designing a game that provides visual feedback one needs to be careful; it can get in the way. The aim of this game is to find which one of 16 boxes contains the hyperlink. The eye moves the cursor in this case a target, which is surprisingly difficult to see, for it moves ever so slightly out of sync.. As you hold your gaze over a box the hue changes slowly from grey to green, and after 2 seconds when a golden heart appears, it's time to try another box.



Should your gaze wander sooner, you start again. Each time you play the game the hyperlink is in a different box, so no cheating.

For now this is a single player html5 game, just plug your new eyecam into your browser box and play. If you've been following, you'll remember the Raspberry Pi cannot run an html5 browser, so for now maybe use a PS3 eyecam and given demand an Raspberry Pi version using x,y coordinates will appear.

## For the future

Browser standards are being developed with the natural assumption that a browser is in use. However input devices do not generally require or need a browser. For this reason it may be helpful to develop a range of independent input-device standards and heuristics which whilst not constrained by browser standards, may be considered.

Social media heatmaps that indicate where people are looking during sports events, films, social interaction and other activities.

A suitable cheap configurable eyetracker may provide the affordances that meet medical needs and thus help us better understand the underlying social models of disability.

A results page is under development to help the user evaluate and reject faulty callibration.

The mk2 has the PS3 eyecam in the middle of field of view (FOV), which obstructs the view, but callibration is simple. The Picameral eye-tracker needs a new matrix transformation, which will improve accuracy of

the iris centre. So unfortunately, for now the eyetracks illustrated are from the mk2 data, overlaying video from the RPi viewcam.

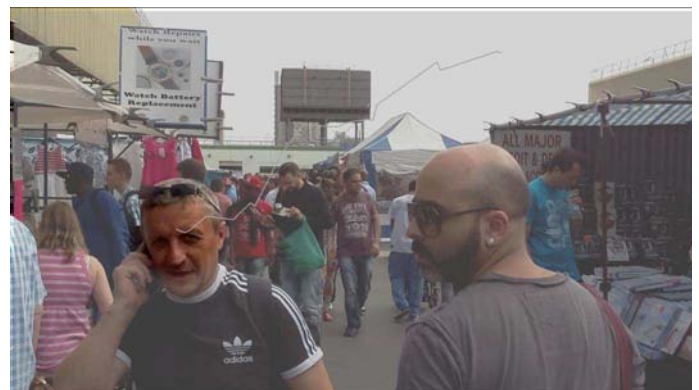
There are many theories of colour and perception. The Ecological approach to visual perception by James J Gibson's describes the sighted animal moving in the natural environment. We follow his belief in this strategy guided by the principal of Gestalt, and the RPi with camera may afford the opportunity to test and develop his hypothesis.

Cartography involves visualising complex data sets, that may include finance, economics, ecology, game theory and is not limited to geography.

My introductory visualisation is maintained at <http://www.peepo.com>, where you can play Go, the ancient oriental game of strategy that is played on a 19x19 board. Go is easy to learn but hard to master, and whilst more complex than chess, computers using Monte Carlo Tree Search are making progress.

The eye-tracker described is intended to be used to contrast a study into how different users understand and communicate using maps, with Go as the metaphor, against another study of eye use in vivo. I encourage others to contribute their own ideas.

For now the Picameral eye-tracker is a proof of concept.



More details and videos on the website <http://www.gnote.org>  
contact: Jonathan Chetwynd [jay@peepo.com](mailto:jay@peepo.com)