

Gaze Estimation with Pupil Tracking in Real-Time

Midterm Presentation

Tomas Bartipan
Supervisor: Benjamin Busam

Technical University of Munich
Chair of Computer Aided Medical Procedures
Project Management and Software Development for Medical Applications

November 9, 2017

tomas.bartipan@tum.de

The basics

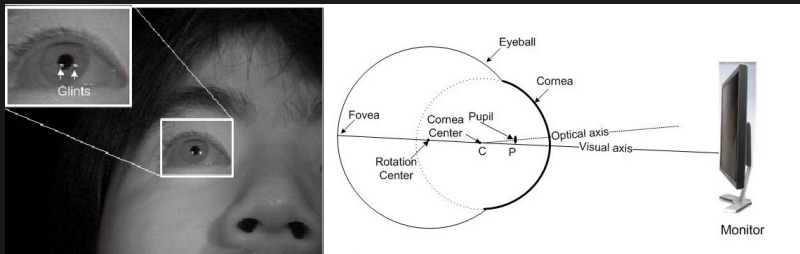


Figure: NIR light reflection and eye schematic

Source:

Real-Time Eye Gaze Tracking Under Natural Head Movements
<https://www.ecse.rpi.edu/~cvrl/zhiwei/gazetracking/gazetracking.html>

Accessed: 9/11/17

The hardware



Figure: Hardware in use

Example input

The problems to solve

- 1 Locate eye region
- 2 Locate iris center
- 3 Locate glint
- 4 Calculate gaze from iris-glint vector and eye position

The problems to solve

- 1 Locate eye region
- 2 Locate iris center
- 3 Locate glint
- 4 Calculate gaze from iris-glint vector and eye position

The problems to solve

- 1 Locate eye region
- 2 Locate iris center
- 3 **Locate glint**
- 4 Calculate gaze from iris-glint vector and eye position

The problems to solve

- ① Locate eye region
- ② Locate iris center
- ③ Locate glint
- ④ Calculate gaze from iris-glint vector and eye position

Components

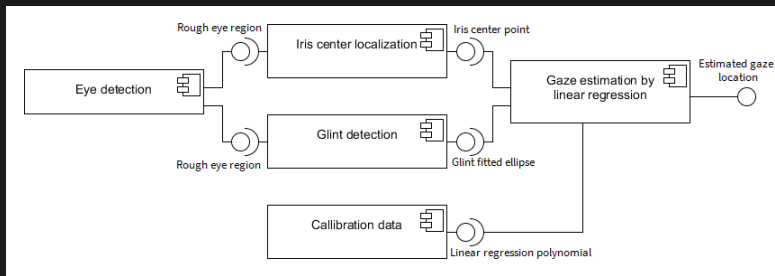


Figure: Component diagram

The methods used

Eye detection

Haar features

Iris localization

Iris detection by way of gradients

Glint detection

Direct Least Squares Fitting of Ellipses

Eye-gaze estimation

Linear Regression

Haar features

- Simple to implement
- Efficient
- Robust
- Very useful to detect one specific thing

Iris detection by means of gradients

- Simple to implement
- Efficient
- Robust
- Precise
- Specialized

Iris detection by means of gradients - How it works

- 1 Generate gradient map
- 2 Find places with high gradients
- 3 On each high gradient, create a line along the jump
- 4 Each pixel that is hit by the line gets incremented
- 5 Select pixel with highest count

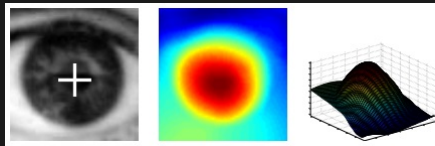


Figure: Eye image and resulting heatmap

Source:

Accurate Eye Centre Localisation by Means of Gradients.
E. Barth, F.Timm, 2011

Direct ellipse fitting

- Very simple to implement
- Very efficient
- Robust
- By putting smart constraints on a 5-DOF minimization problem, we get a set of 2 equations
- Up to six solutions - pick the one with lowest error

Direct ellipse fitting

- But will one ellipsis be enough?
- Use 2 ellipses?

Example input

Linear regression

- Simple to implement
- Very efficient
- The difficult calculations are performed in calibration stage
- Possibly not the best performer

Linear regression - how it works

- Create a polynomial transformation function
- Input(variables) = eye center position, glint ellipse parameters
- Output = screen coordinates
- In calibration phase, let user look at many (15-30) points and calculate polynomial coefficients that minimize squared error
- In action phase, plug the actual values into the polynomial and hope for the best

GANTT timeline

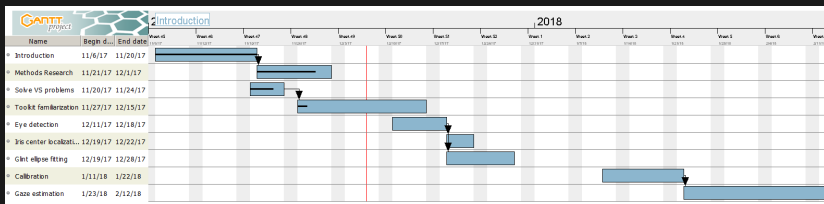


Figure: Gantt time-chart

That's it, folks!

Questions?