Camgaze.js : Mobile Eye Tracking and Gaze Prediction in JavaScript

Alex Wallar
Christian Poellabauer
Patrick Flynn

Abstract

1 Introduction

2 Motivation

3 Related Works

4 Implementation

`Camgaze.js` goes through two steps in order to predict the gaze direction. Firstly, `Camgaze.js` detects each pupil. It then uses the pupils deviation from a unique point on the face to determine the gaze metric, $\mathcal{G}$. This metric needs to be calibrated in order for there to be a mapping from $\mathcal{G}$ to a point on the screen. Once this gaze metric has been calibrated, `Camgaze.js` should be able to interpolate area of the screen the user is looking at. A high level description of the algortihm is shown below.

Algorithm 1 Pseudocode for `Camgaze.js`

$\mathcal{F} \leftarrow$ InitGazeMapping()
while StillCalibrating() == `true` **do**
  $P_{list} \leftarrow$ DetectPupils()
  $\mathcal{G} \leftarrow$ DetermineGazeMetric(P_{list})
  $\mathcal{F} \leftarrow$ Calibrate($\mathcal{G}, \mathcal{F}$)
while SessionFinished() == `false` **do**
  $P_{list} \leftarrow$ DetectPupils()
  $\mathcal{G} \leftarrow$ DetermineGazeMetric(P_{list})
  ProjectGazeOntoScreen($\mathcal{F}(\mathcal{G})$)

4.1 Pupil Detection

Detecting the pupils enables `Camgaze.js` to determine the gaze direction. Pupil detection in this approach is aimed to be fast in order to be deployable onto mobile devices. Firstly, the eyes are detected using the Viola-Jones Object Detection Framework [Viola and Jones 2001].

4.2 Determining the Gaze Metric

4.3 Calibration

5 Testing

6 Applications

7 Discussion

References

VIOLA, P., AND JONES, M. 2001. Robust real-time object detection. In *International Journal of Computer Vision*.