

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 algorithm is shown below.

Algorithm 1 Pseudocode for `Camgaze.js`

```
1:  $\mathcal{F} \leftarrow \text{INITGAZEMAPPING}()$ 
2: while STILLCALIBRATING() == true do
3:    $P_{list} \leftarrow \text{DETECTPUPILS}()$ 
4:    $\mathcal{G} \leftarrow \text{DETERMINEGAZEMETRIC}(P_{list})$ 
5:    $\mathcal{F} \leftarrow \text{CALIBRATE}(\mathcal{G}, \mathcal{F})$ 
6: while SESSIONFINISHED() == false do
7:    $P_{list} \leftarrow \text{DETECTPUPILS}()$ 
8:    $\mathcal{G} \leftarrow \text{DETERMINEGAZEMETRIC}(P_{list})$ 
9:   PROJECTGAZEONTOSCREEN( $\mathcal{F}(\mathcal{G})$ )
```

4.1 Pupil Detection

4.2 Determining the Gaze Metric

4.3 Calibration

5 Testing

6 Applications

7 Discussion

References