

# Analyse

**Bachelorprojektet:  
Real-time eye-tracking  
Projektnummer: 15017**



**Version 1.0 18/03/2015  
Studerende: Søren Vøgg Krabbe Lyster (SVL) 10920,  
Martin Degn Kristensen (MDK) 10441  
Studieretning: Elektro  
Vejleder: Preben Kidmose**

---

## Revision History

Revision	Date	Author(s)	Description
1.0	18.03.15	SVL,MDK	Oprettet

## Indhold

<b>1</b>	<b>Formål</b>	<b>3</b>
<b>2</b>	<b>Systemoversigt</b>	<b>4</b>
2.1	Algoritme Oversigt . . . . .	4
2.1.1	Calculate pupil and gaze . . . . .	4
2.1.2	Locate corneal reflection . . . . .	5
2.1.3	Starburst pupil contour detection . . . . .	5
2.1.4	Locate Edge Points . . . . .	5
2.1.5	Fit ellipse ransac . . . . .	5
2.2	Fokuspunkter . . . . .	5
2.2.1	Kantdetektion . . . . .	5
2.2.2	Ellipse Tilpasning . . . . .	5
2.2.3	Kegleparametre til Ellipseparametre . . . . .	6
2.2.4	Kamera Input . . . . .	6
<b>3</b>	<b>Starburst Analyse</b>	<b>7</b>
3.1	Simulering . . . . .	7
3.1.1	Extract gaze vector from video . . . . .	7
3.1.2	Variabel Ændringer . . . . .	7
3.1.3	Resultater . . . . .	7
<b>4</b>	<b>Overvejelser angående systemkrav</b>	<b>8</b>
4.1	Funktionelle krav . . . . .	8
4.1.1	Real-time eye-tracking . . . . .	8
4.1.2	Kalibrering . . . . .	8
4.1.3	Output . . . . .	8
4.1.4	Brugertilgang . . . . .	8
4.2	Ikke-funktionelle krav . . . . .	8
4.2.1	Fejlmargen . . . . .	8
4.2.2	Real-time . . . . .	8
4.3	Overvejelser angående kodesprog . . . . .	8

# 1 Formål

Det følgende afsnit beskriver analyseprocessen i projektforsøget. Formålet med denne analyse er at skabe et teoretisk fundament som kan anvendes i design og implementeringsfaserne af projektet. Fundamentet bygger på en kombination af gennemgang og simulering af det kode som projektet tager udgangspunkt i, og derudover en undersøgelse af eksisterende metoder til eyetracking og sammenligning af performance<sup>1</sup>. Sidst men ikke mindst undersøges også om en Python-baseret platform er en fornuftig løsning, og i så fald hvorfor. (Tilføjelser?).

---

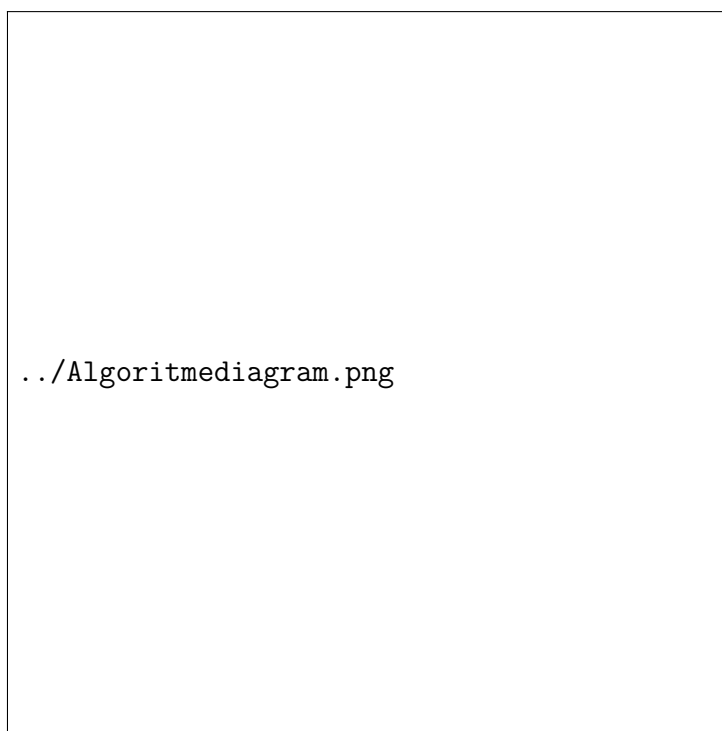
<sup>1</sup>Performance med henblik på processeringstid, eventuelt også præcision+nøjagtighed

## 2 Systemoversigt

I det følgende vil koden som projektet tager udgangspunkt i blive gennemgået i dybden, hvilket også indebærer processeringstid for de enkelte subrutiner. I forlængelse af dette vil begrundelserne for krav der har med algoritmen at gøre også blive givet i en relevant kontekst.

Bemærk at procestiden er opgivet som procentdel af samlet procestid, og at værdierne kun omfatter processen selv, og ikke medregner den tid der bruges på metoder der kaldes undervejs. Procestiden for disse metodekald står ud for de enkelte metoder i stedet.

### 2.1 Algoritme Oversigt



Figur 1: Systemdiagram for Real-time eye-tracking

#### 2.1.1 Calculate pupil and gaze

Main funktion - 0.17

### 2.1.2 Locate corneal reflection

Finder reflektionspunkter - 0.3

### 2.1.3 Starburst pupil contour detection

Starburst algoritme - 0.63

### 2.1.4 Locate Edge Points

Find pupil kant punkter - 44.71

### 2.1.5 Fit ellipse ransac

Tilpas en ellipse til punkterne - 11.7

## 2.2 Fokuspunkter

I det følgende underafsnit vil der kort blive beskrevet hvilke dele af systemet der med fordel kan fokuseres på. Dette er baseret på antal gange rutinen bliver kørt, samt hvor lang tid det tager for processen at blive færdig. Begrundelsen for dette er at det højst sandsynligt er lettere at øge performance med en betydelig del hvis de dele der tager længst tid først bliver optimeret.

### 2.2.1 Kantdetektion

Den mest tidskrævende del af algoritmen er selve starburst-delen, altså den del hvor pupilkantpunkter findes. Udover at optimere koden ved at køre i C, er der også mulighed for at justere variabler for at få processen til at tage kortere tid. Dette vil dog højst sandsynligt ske på bekostning af resultaternes nøjagtighed/præcision, eller systemets overordnede stabilitet. Der vil altså formentligt være behov for en balancering af performance overfor kvalitet. Denne balancering vil foregå iterativt i løbet af implementeringsfasen af projektet.

### 2.2.2 Ellipse Tilpasning

Den næstmest tidskrævende del af algoritmen er ellipsetilpasningen. Her vil det igen være muligt at justere på variabler, for at balancere performance og kvalitet. Derudover skal der ses nærmere på undtagelsestilstande hvor algoritmen gentages mange gange, RANSAC iterations = 10000. Hvis disse undtagelsestilstande opsluger meget tid og sker ofte, kunne det have en markant effekt på performance for systemet.

### 2.2.3 Keglparametre til Ellipseparametre

En mindre betydelig, men dog stadig mærkbar del af algoritmen er den del som oversætter kegleparametre til ellipseparametre. Da denne del af algoritmen kun udgør omkring 6 procent af den samlede procestid, er denne blot nævnt som en mulig kandidat for optimering hvis de primære dele begynder at være sammenlignelige i procestid.

### 2.2.4 Kamera Input

En sidste ting der med fordel kan fokuseres på at forbedre er indlæsning af data fra kameraet. Idet der ikke sker meget kompression bør det meste af arbejdet bestå af overførsler i harddisken, men hvis det viser sig at være for tungt kan vi forsøge at gøre noget.

## 3 Starburst Analyse

I det følgende vil simuleringer i forbindelse med starburst algoritmen blive gennemgået. Disse simuleringer er blevet udført i et forsøg på at danne en bedre indsigt i sammenhængen imellem performance og resultater. Idet en overgang til en anden platform end MATLAB formentligt ikke forøger performance tilstrækkeligt, skal der i stedet undersøges hvorvidt det er muligt at tilpasse forskellige variabler til at opnå en kortere processeringstid, alt imens resultaterne forbliver gode.

### 3.1 Simulering

Simuleringerne er foretaget med det kode og videodata som Daniel Sibozka har udleveret til gruppen. I det første afsnit vises resultaterne, og i andet afsnit beskrives hvilke ændringer af variabler der afprøves, og derefter vises resultaterne af disse ændringer.

#### 3.1.1 Extract gaze vector from video

Processeringstid, CPU = 2.4 GHz

Figur over resultater.

#### 3.1.2 Variabel Ændringer

Antal Rays Antal RANSAC iterationer

#### 3.1.3 Resultater

Figur

## 4 Overvejelser angående systemkrav

De krav der er blevet nået frem til i kravspecifikationen vil blive forklaret i det følgende:

### 4.1 Funktionelle krav

#### 4.1.1 Real-time eye-tracking

#### 4.1.2 Kalibrering

#### 4.1.3 Output

#### 4.1.4 Brugertilgang

### 4.2 Ikke-funktionelle krav

#### 4.2.1 Fejlmargin

#### 4.2.2 Real-time

### 4.3 Overvejelser angående kodesprog



## Litteratur