# Implementatieplan edge detection

## Namen en datum

Team: On the edge

Teamlid 1: Rik Ruttenberg, 1677636

Teamlid 2: Maarten Kroon, 1567452

Datum:

## Doel

Voor de practicum opgaven van het vak TCTI-V2VISN1-13 hebben we de opdracht gekregen om één van de standaard algoritmes voor gezichtsherkenning opnieuw te implementeren. De eis is dat het nieuwe algoritme een vergelijkbaar of zelfs een beter resultaat oplevert dan het standaard algoritme.

Ons gekozen onderdeel is edge detection en ons doel is randen sneller en eventueel ook beter te herkennen. Er is nu op sommige afbeeldingen nog ruis te zien na de edge detection. Dit levert soms problemen op tijdens de allocatie stap. We willen proberen deze ruis te verminderen. Ook hebben we meerdere afbeeldingen getest van verschillende grootte. Als de afbeeldingen erg groot worden werkt het standaard edge detection algoritme erg traag. We denken dat hier tijdwinst te behalen is.

## Methoden

Er zijn heel veel verschillende methodes die gebruikt kunnen worden voor edge detection. De meeste van deze methodes lijken sterk op elkaar. We willen ons focussen op onderstaande methodes:

* Sobel
* Canny
* Robert
* Prewitt
* Laplacian

We hebben deze methodes gekozen omdat dit de meest gebruikte technieken zijn en hier veel informatie over beschikbaar is. De verschillende methoden hebben elk hun voor en nadelen welke opgesomd staan in de tabel hieronder:

|  |  |  |
| --- | --- | --- |
| **Methode** | **Voordelen** | **Nadelen** |
| Sobel | Gemakkelijk te implementeren.  Kan lijnen in alle richtingen detecteren. | Gevoelig voor ruis.  Onnauwkeurig. |
| Prewitt (lijkt sterk op Sobel) | Iets sneller dan Sobel. | Produceert meer ruis dan Sobel.  Detecteert minder lijnen dan Sobel. |
| Canny | Weinig last van ruis. | Complex en lastig te implementeren.  Trager dan andere edge detection mehodes.  Detecteert soms lijnen die er niet zijn. |
| Robert’s Cross | Eenvoudig te implementeren. | Veel detail gaat verloren. |
| Laplacian | Goed in het vinden van de juiste locatie van lijnen.  Neemt een groter gebied rondom elke pixel mee. | Foutgevoelig in randen en gebogen lijnen waar de intensiteit van de grijswaarden varieert. |

(Raman Maini, 2009)

Wanneer we kijken naar enkele voorbeeld plaatjes zijn de verschillen tussen de methodes duidelijk te zien:

|  |  |  |
| --- | --- | --- |
|  |  |  |
| Origineel | Sobel | Prewitt |
|  |  |  |
| Roberts | ZeroCross (laplacian) | Canny |

|  |  |  |
| --- | --- | --- |
|  |  |  |
| Origineel | Sobel | Prewitt |
|  |  |  |
| Roberts | ZeroCross (laplacian) | Canny |

(Monica Avlash, 2013)

## Keuze

Onze uiteindelijke keuze is Sobel edge detection geworden. Met name de eenvoud van implementeren en de snelheid van het algoritme zijn hier belangrijke factoren voor. We denken dat de resultaten van het algoritme voldoende detail bevatten voor gezichtsherkenning. Door gebruik van een goed smoothing filter hopen we problemen met ruis te voorkomen.

## Implementatie

Je geeft aan hoe deze keuze is geimplementeerd in de code

## Evaluatie

We willen timers gebruiken, welke we voor en na het edge detection algoritme plaatsen om het verschil in snelheid tussen de algoritmen te meten. We zullen met een aantal afbeeldingen van verschillend formaat meerdere testen doen om zo een gemiddelde te bepalen. Dit gemiddelde vergelijken we met het gemiddelde van het standaard algoritme om te bepalen of het door ons gemaakte algoritme daadwerkelijk sneller is.

# Bibliografie

Monica Avlash, D. L. (2013). Performances analysis of different edge detection methods on road images. *International Journal of Advanced Research in Engineering and Applied Sciences*, 12.

Raman Maini, D. H. (2009). Study and Comparison of Various Image Edge Detection Techniques. *International Journal of Image Processing*, 12.