13-2-2020

Stijn van Wijk, Stefan Stegeman

V2A

Implementatie-plan

Vision practicum

# Doel

Wij willen een gezichtsherkenningsalgoritme maken dat te gebruiken is in bijvoorbeeld een smart door. Hierbij is het aller belangrijkste dat het algoritme accuraat is, want het algoritme geeft je toegang tot gevoelige spullen. De camerabeelden kunnen soms ruis bevatten door de kwaliteit van de camera of door viezigheid op de lens. Dit kan een hoop problemen en narigheid opleveren. Om dit probleem te verminderen willen wij ons gaan richten op de accuraatheid van het algoritme.

Hierdoor zijn we dan ook minder streng op het gebied van snelheid, we willen dat het algoritme maximaal even lang duurt als de actie die het algoritme vervangt. Dit betekent overigens niet, dat wij de snelheid compleet zullen negeren. Wanneer het mogelijk is kiezen wij uiteraard voor de snelste oplossingen. Bij de smartdoor is de actie die wordt vervangen het open maken van de deur met een sleutel, dit duurt ongeveer drie seconden. De tijd om de deur open te maken kan erg verschillen, maar bij onze implementatie gaan we dan dus ook uit van drie seconden.

Er zijn meerdere manieren om de accuraatheid van een face recognition algoritme te verbeteren, zoals het verbeteren van features detection, extraction of edge detection. Wij kiezen ervoor om de edge detection accurater te maken. De edge detection maakt een afbeelding waar alle facial features gezocht worden. Als de edge detection accurater is, is de features detection waarschijnlijk ook accurater.

# Methoden

## Sobel

### Beschrijving

Wij hebben in de paper Sobel Edge Detection Algorithm van Gupta en Mazumdar (2013) de volgende informatie gevonden over de Sobel Operator.

De Sobel Edge Detection methode maakt gebruik van de afgeleide (differentiaal quotiënt) om edges te detecteren. Er worden twee kernels op het plaatje toegepast. Één van de kernels is voor het detecteren van edges in de x-as, en de andere voor in de y-as. De twee afbeeldingen met de edges in de x en y-as worden samengevoegd d.m.v. Pythagoras, dit geeft de afbeelding met de edges.

### Pluspunten

* De Sobel methode is erg snel.
* De kernels verminderen de ruis tot een zekere mate.
* De kernels zorgen ervoor dat het verschil tussen de twee rijen of kolommen wordt verbeterd. (Gupta & Mazumbar, 2013, p. 1579)

### Minpunten

* Heeft moeite om edges te detecteren die niet ‘duidelijk’ zijn. (Gupta & Mazumbar, 2013)
* Het is moeilijk om edges te detecteren die niet zo strak gedefinieerd zijn.

## The Prewitt

### Beschrijving

De Prewitt methode is bijna hetzelfde als de Sobel methode. Het verschil tussen deze twee methoden wordt uitgelegd in het volgende citaat. “Therefore, note that, unlike the Sobel operator, this operator does not place any emphasis on pixels that are closer to the center of the masks.”( G.T. Shrivakshan & Dr.C. Chandrasekar, 2012, p.272).

### Pluspunten

* De Prewitt methode is snel en gemakkelijk.

### Minpunten

* De uitkomst kan erg ruw zijn op sommige plekken­.
* De Prewitt methode kan inacuraat zijn met veel ruis.

## Canny

### Beschrijving

Canny edge detection is ontwikkeld om een optimale edge detector te maken. Het algoritme is op te delen in vijf stappen. De stappen zien er als volgt uit:

1. Weg filteren van de ruis met een Gaussian filter.
2. Het vinden van de edges met bijvoorbeeld Sobel of Prewitt. (wiki)
3. Het wegfilteren van NIET local maxima.
4. Double thresholding om potentiële edges te krijgen
5. D.m.v. Hysterese edges wegfilteren die niet aan een zekere edge zitten.

De laatste stap komt niet voor in elke paper over Canny.

### Pluspunten

* Een deel van de noise wordt weggefiltert met Gaussian.
* Betere edge detection d.m.v. thresholding.
* Vind edges met een lage error rate.
* Elke edge is maar één keer gemarkeerd.

### Minpunten

* Moeilijk om te implementeren.
* Veel berekeningen, dus neemt veel tijd in beslag.

## Fuzzy logic

### Beschrijving

Wij hebben het volgende kunnen vinden over Fuzzy logic in de paper Edge detection in digital images using Fuzzy logic technique (Shrivakhan & Chandrasekar, 2012).

Fuzzy logic maakt gebruik van 3x3 binary matrices. De pixels die aan de rand zitten worden toegewezen aan een bereik van waarden die allemaal van elkaar verschillen. Dit resulteert in scherpe randen, die gemakkelijk te detecteren zijn. Met behulp van Fuzzy logic reasoning worden edges gevonden zonder dat er thresholding moet worden toegepast.

### Pluspunten

* De Fuzzy rules zorgen ervoor dat er zo min mogelijk dubbele lijnen zijn in het resultaat.
* De edges zijn smooth.
* De hoeken zijn scherp waardoor ze goed gedetecteerd kunnen worden.

### Minpunten

* Langzaam en erg ingewikkeld ten opzichte van Sobel.
* Als er noise in image zit, dan moet je model afstellen.

# Keuze

Wij gaan voor de Sobel edge-detection methode. Dit doen wij om de volgende redenen. De Sobel methode werkt erg goed omgaan met ruis. Dit komt omdat de kernels die Sobel gebruikt de ruis van de foto kan verminderen. Dit is wat wij graag willen hebben voor ons doel. De camera in een smart-door is ten eerste niet van beste kwaliteit, en het kan voorkomen dat het buiten aan het regenen is. Dit zal voor ruis zorgen, wat niet gewenst is omdat dit zomaar de herkenning zou kunnen beïnvloeden. Ook kan er viezigheid op de lens komen, de camera zit immers aan de buitenkant van de deur.

Methoden zoals Prewitt en Fuzzy logic zijn dan dus ook niet slim om te gebruiken. Deze werken minder goed met ruis, wat voor ons cruciaal is. De Canny methode gaan we ook niet gebruiken. Ondanks dat wij de nadruk niet leggen op de snelheid van het systeem, gaan wij toch liever voor de Sobel methode, die sneller is dan de Canny methode. Ook zal de Canny methode moeilijker zijn om te implementeren, waarvan wij vrezen dat dit de accuraatheid zal kunnen verminderen omdat wij dan minder de focus kunnen gaan leggen op de methode verbeteren.

# Implementatie

# De eerste stap van de implementatie zal de afbeelding omzetten naar een opencv matrix worden.

Hierna zullen wij een x-as en een y-as Sobel kernel over de afbeelding laten lopen.

De resulterende afgeleiden van de kernels zullen wij hierna samenvoegen doormiddel van Pythagora tot één matrix. Dit zal dan de afbeelding opleveren die alle edges bevat.

# Evaluatie

Ons onderzoek zal als volgt verlopen.

Om te testen of onze methode een stuk accurater is dan de huidige geïmplementeerde methode zullen wij een test set nodig hebben met foto’s waarvan de edges al bekend zijn.

Zo kan er nadat beide implementaties (De aangeleverde en de door ons geschreven) hun resultaten hebben opgeleverd (afbeeldingen met detected edges), gekeken worden hoe groot het verschil is met de afbeelding die bij de dataset zat.

Doormiddel van het vergelijken van de edges die wij hebben gedetecteerd en de al bekende edges van de dataset, kunnen wij zien hoeveel ze van elkaar verschillen. Deze verschillen kunnen we dan naast elkaar leggen om te zien welke implementatie beter is.

Dit gaan wij uitvoeren op de hele dataset. Dit houdt in dat wij meerdere afbeeldingen gaan testen en niet maar een paar afbeeldingen. Dit levert een beter resultaat op omdat er zo meerdere proeven zijn gedaan.

# Bibliografie

Alshennawy, A. A., & Aly, A. A. (2009). Edge Detection in Digital Images Using Fuzzy Logic Technique Vol. 3, No. 3. *International Scholarly and Scientific Research & Innovation*, pp. 540-548.

Gupta, S., & Mazumbar, S. G. (2013). Sobel edge detection algorithm. *International Journal of Computer Science and Management Research Vol. 2, Issue 2*, 1578-1583.

Shrivakhan, G. T., & Chandrasekar, C. (2012). A Comparison of various Edge Detection Techniques used in Image Processing. *IJCSI International Journal of Computer Science Issues Vol. 9, Issue 5, No 1*, 269-276.