

# Meetrapport edge detection

Door Renske Kuip en Olivier Verwoerd

Vision - Arno Kamphuis  
april 2018



# Inhoud

<b>Doel</b>	<b>3</b>
<b>Hypothese</b>	<b>3</b>
<b>Werkwijze</b>	<b>4</b>
<b>Resultaten</b>	<b>4</b>
Edge detection	5
Threshold	6
<b>Verwerking</b>	<b>6</b>
Edge detection	6
Thresholding	7
<b>Conclusie</b>	<b>7</b>
<b>Evaluatie</b>	<b>7</b>

## Doel

Het doel van het project is om ervoor te zorgen dat edge detection te implementeren, zodat opencv gezichten kan herkennen. In ons implementatieplan is gekeken naar verschillende methoden voor edge detection en hebben we deze met elkaar vergeleken. Hieruit is gekomen dat wij willen weten of door middel van zowel een apart Gaussian filter gevolgd door een Laplacian filter als de Laplacian of Gaussian (hierna LoG genoemd) methode edges in grijswaarden afbeeldingen gedetecteerd kunnen worden. Deze twee methodes willen we allebei implementeren en de resultaten worden met elkaar vergeleken. De LoG methode bestaat uit slechts één kernel, dus deze zal naar verwachting sneller zijn. Echter willen we weten of het resultaat hier ook daadwerkelijk beter van wordt, aangezien deze gevoeliger is dan de twee bewerkingen apart uitvoeren.

Het resultaat van de gekozen edge detection wordt vervolgens verwerkt door thresholding naar een zwart-wit afbeelding, zodat de gezichtsherkenning hierop kan worden toegepast.

Hieraan zijn een aantal eisen waar de methode aan moet voldoen. Het moet onder meer een efficiënte methode zijn. Om deze reden willen we niet te veel operaties uitvoeren op de afbeelding, om te voorkomen dat dit veel rekenkracht kost. Hierbij wordt ook in relatie gekeken met het geleverde eindresultaat van beide methodes.

In het implementatieplan is een hoofdvraag beschreven. Deze hoofdvraag is als volgt: “Welke combinatie van edge detection en thresholding methode zal voor gezichtsherkenning het beste werken?” Door middel van literatuurstudie is een antwoord gevonden op deze vraag in theorie. In dit document worden de resultaten van de implementatie beschreven.

## Hypothese

De hypothese is dat de LoG methode sneller is dan de Gaussian filter apart van de Laplacian methode, maar dat de resultaten niet ver van elkaar af zullen liggen. Dit omdat de LoG minder berekeningen vereist, maar ze in feite hetzelfde doen. Echter zal de LoG in combinatie met een dynamische threshold iets minder snel zijn, maar wel het beste resultaat geven. De verwachting is dat een 5x5 kernel voor de filters een goed resultaat geeft. Omdat de afbeeldingen relatief klein zijn, zal een klein filter al zeer veel effect hebben. De edges zijn namelijk relatief smal.

Door het verschil in achtergrond en huidskleur zal het moeilijk worden om threshold met een vaste waarde goed te laten werken. We hebben daarom hoogstwaarschijnlijk een dynamische thresholding nodig om een goed resultaat te verkrijgen.

## Werkwijze

Om te beginnen hebben we voor de edge detection en voor threshold een timer gezet die meet hoe lang de functie erover doet om uit te voeren. Daarnaast controleren we de voorbeeldafbeeldingen allemaal bij het uitvoeren van de test. Hieruit komt een getal hoeveel afbeeldingen er correct zijn verwerkt en hoeveel het er niet deden.

We geven alle methoden een score op basis van de vision\_timer van Arno Kamphuis. Met deze benchmark kunnen we CPU onafhankelijker meten. We gebruiken daarmee de volgende formule om de score te kunnen berekenen:

$\text{Score} = 1/\text{vision\_timer} \times \text{milliseconden}$

Lager = beter

Om verschillen tussen de afbeeldingen te voorkomen in de meetresultaten, zoals grootte en pixelwaardes die kunnen leiden tot een snellere berekeningen, gebruiken we alleen female-1.png als afbeelding.

Na de implementatie van een meting van pixels voor minimum, maximum en gemiddelde pixels kunnen er formules worden opgesteld voor een dynamische thresholding. Door een meting te doen van alle threshold waardes per afbeelding die een positief resultaat leveren bij de gezichtsherkenning en daaruit de verbanden te trekken, komt er een dynamische threshold waarde uit die goed werkt.

## Resultaten

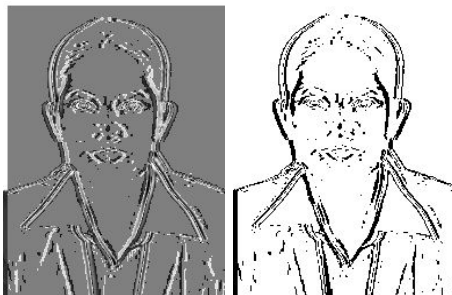
Dit zijn de resultaten van de verschillende edge detection methoden die we hebben getest. Links de afbeelding female-1 met de edge detection en rechts de afbeelding na de dynamische threshold. Boven elke afbeelding staat welke methode er is toegepast.

## Edge detection

Eerst Gaussian 5x5 en daarna Laplacian met kernel 5x5, score 35, resultaat 0/7



Eerst Gaussian 5x5 en daarna Laplacian met kernel 9x9, score 75, resultaat 2/7



Kernel LoG 7x7,  $\sigma = 1.05$ , score 33, resultaat = 7/7



Kernel LoG 9x9,  $\sigma = 1.4$ , = score 52, resultaat = 1/7



## Threshold

De threshold is meerdere keren gemeten met twee verschillende methodes. Hierbij is de constante van de edge detection mexican hat versie. Om te beginnen met de methode van een vaste waarde. Hierbij hebben we een waarde van 120 gekozen. Alle pixels onder deze waarde worden dan zwart. De overige pixels worden dan wit. Dit resulteert in een score van 1.7 en is daarbij zeer snel maar niet dynamisch. 4 van de 7 afbeeldingen resulteerde succesvol voor edge detection.

Threshold Waardes die werken met gezichtsherkenning

picture	Low	High	Diff	AVG	Mid	Range
female-1	34	255	221	177	144	118-122+124
female-2	26	217	191	172	121	120+121+128
female-3	104	235	131	177	169	126
male-1	86	255	169	181	170	98+100+101+120+122+125+128
male-2	40	255	215	176	147	114+116+117
male-3	94	243	149	179	168	125
child-1	95	255	160	184	175	116-120

Het resulteert in een beter beeld voor de gezichtsherkenning. 7 van de 7 afbeeldingen resulteerde succesvol voor edge detection wat een verbetering is van 75%.

De score stijgt naar 3.7. Stijging van tijd met 122%. Na de verwerking van een steekproef in plaats van alle pixels meenemen voor de waardes, daalde de score naar 2.5. Een stijging van tijd met 47% ten opzichte van tijd zonder dynamische waarde.

## Verwerking

Door te kijken naar de verbanden in resultaten merken we dat een betere verwerking ten koste gaat van tijd. Het beste werkt een Laplacian of Gaussian met een dynamische threshold. Dit laat zien dat onze hypothese grotendeels klopt.

## Edge detection

Het verschil met onze hypothese is de kernel grootte. Er is uiteindelijk gebruik gemaakt van een kernel van 7x7 in plaats van 5x5. Dit kan mogelijk komen door de JPEG-compressie op de afbeeldingen. Dit zorgt voor een meer geblokte afbeelding. Om deze verstoring te verhelpen is een grotere kernel nodig. Zou je de kernel nog groter maken, dan zorgt dit ervoor dat de fijne edges te globaal worden gezien. Dit levert te dikke strepen op voor de edge detection. Dit is

heel duidelijk te zien bij de LoG met een 9x9 kernel. Daarentegen met een kleinere kernel wordt er ruis meegenomen in het resultaat wat direct ongeschikt is voor de gezichtsherkenning. Er is uit de metingen voort gekomen dat de LoG methode sneller is dan de andere methodes. Dit is te zien aan de lage score van 33.

## Thresholding

Thresholding is een relatief simpele, maar niet onbelangrijke bewerking. Bij het vergelijken van de verschillen is het belangrijk om te zien dat beide scores vrij laag zijn in verhouding met de edge detection scores. Zo is de toename van score van 1.7 naar 3.7 een toename van 122%. Als we rekening houden met de edge detection van 33 wordt het 34.7 tegenover 36.7. Zo is het daadwerkelijke verschil in tijd in zijn geheel met edge detection slechts toegenomen met 6%, terwijl het resultaat is toegenomen met 75%. Door het verbeteren van de resultaten is het duidelijk dat de dynamische veel beter werkt dan een vaste threshold waarde en dat dit zeker de extra berekening waard is.

Dit zorgt dat de totaalscore van de dynamische thresholding bij onze test eindigt op  $33 + 3.7 \approx 37$ .

## Conclusie

Onze hoofdvraag is: "Welke combinatie van edge detection en thresholding methode zal voor gezichtsherkenning het beste werken?" We kunnen als antwoord hierop, op basis van onze metingen, concluderen dat edge detection het beste en het snelst werkt met een Laplacian of Gaussian filter met 7x7 kernel in combinatie met een dynamische threshold op basis van een steekproef.

De edge detection met een 7x7 kernel geeft het beste beeld wat ook wordt bevestigd door de gezichtsherkenningsssoftware. Door het combineren van de Gaussian en laplacian kernels worden de berekening sneller uitgevoerd.

De thresholding met een dynamische waarde op basis van een steekproef, heeft het meeste voordeel zonder dat dit ten koste gaat van de snelheid. Deze combinatie levert het beste resultaat op voor de gezichtsherkenning.

## Evaluatie

Het gehele onderzoek is goed verlopen. Dit is te zeggen op basis van het goed resultaat om 7 van de 7 afbeeldingen correct te kunnen gebruiken voor gezichtsherkenning, maar ook omdat we een duidelijke keuze hebben kunnen maken in het implementatieplan. De methode die we hebben gekozen was succesvol.

Er is helaas veel tijd verloren gegaan bij de implementatie aan het maken van het `intensityImageStudent.cpp` bestand. Dit omdat we dachten dat get- en setpixel ook nog bij deze casus hoorde.

Ook hebben we eigenlijk alleen maar gekeken naar hoe goed het werkt op een Intel multicore processor. Dit mede door het ontbreken van apparaten met een andere structuur. Het zou kunnen betekenen dat ons resultaat van scores niet geldt voor een andere processors. Dit zou een verbeterpunt kunnen zijn van ons onderzoek. Ook zou ter verbetering een betere threshold formule gemaakt kunnen worden, aangezien de huidige methode erg simpel is.

Daarnaast is er niet getest op andere afbeeldingen. 7 afbeeldingen is een zeer kleine test methode. Een betere methode zou zijn om nog veel meer afbeeldingen te gebruiken als test om zo de succes rate meer waarde te kunnen geven.