

# Implementatieplan edge detection

Door Renske Kuip en Olivier Verwoerd

Vision - Arno Kamphuis - 2018



## Doel

Het doel van het project is om een bruikbare edge detection filter te maken die gebruik maakt van een thresholding filter om zo een afbeelding te maken die duidelijk laat zien waar de randen zijn. Dit resultaat moet ervoor zorgen dat andere berekeningen makkelijk uitgevoerd kunnen worden zoals gezichtsherkenning.

## Methoden

De twee main categoriën zijn search-based(first derivative ofwel gradient) en zero-based(second derivative ofwel Laplacian).

### 1. Canny methode

Je haalt als eerste een Gaussian filter over je afbeelding. Deze maakt de afbeelding iets gladder waardoor de 'noise' vermindert. Deze filter maakt gebruik van een kernel ter grootte van  $(2k+1) \times (2k+1)$ . Vaak is een 5x5 filter genoeg. Hoe groter de Gaussian kernel, hoe minder de performance van de detector.

Vervolgens ga je op zoek naar edges en elke kant ze op gaan. Het Canny algoritme gebruikt vier filters om horizontaal, vertikaal en diagonale edges te vinden.

Hierna verdun je de edges door middel van *non-maximum* suppressie. Hierbij bekijk je van elke pixel naar de waardes van de omliggende pixels in de y-richting. Als de huidige pixel een hogere edge-waarde heeft dan de omliggende, zal de pixel worden vergeleken met de omliggende in de verticale as en zal de waarde hetzelfde blijven. Anders zal hij worden 'suppressed'.

Om de laatste ruis weg te halen worden edge pixels met een lage gradatie weggehaald. Er worden twee 'threshold' waardes aangemaakt en de pixels worden vergeleken met deze waardes. Als de gradatie waarde van de pixel hoger is dan de hoge threshold wordt deze gezien als een sterke edge pixel en is hij lager dan de lage threshold waarde zal hij worden gezien als zwakke edge pixel en zal worden 'suppressed'.

### 2. Sobel operator

De Sobel operator bestaat uit onderstaande 3x3 kernel.

-1	0	+1
-2	0	+2
-1	0	+1

G<sub>x</sub>

+1	+2	+1
0	0	0
-1	-2	-1

G<sub>y</sub>

Deze kernels zijn gemaakt om met name horizontale en verticale edges te herkennen. Je kunt allebei de kernel toepassen zodat je de absolute gradatie van elke pixel kunt vinden en oriëntatie hiervan.

### 3. Prewitt's operator

Prewitt's operator is in feite hetzelfde als de Sobel operator alleen met andere waarden.

-1	0	+1
-1	0	+1
-1	0	+1

G<sub>x</sub>

+1	+1	+1
0	0	0
-1	-1	-1

G<sub>y</sub>

### 3. Robert's cross operator

De Robert's Cross operator maakt gebruik van spatiale gradatie meting op een afbeelding. Er wordt gebruik gemaakt van twee 2x2 kernels die in een kruisvorm lopen. Het is een vergelijkbare methode als de Sobel operator, maar deze methode werkt het best voor schuine edges.

+1	0
0	-1

G<sub>x</sub>

0	+1
-1	0

G<sub>y</sub>

Je past de kernels weer apart toe en voegt de resultaten van de metingen bij elkaar. Hieruit kun je de absolute waarde van de gradatie en de oriëntatie.

### 4. Laplacian of Gaussian

Bij de Laplacian methode is ook eerst een Gaussian filter gehaald om hem minder gevoelig te maken voor noise/ruis, want de kernels die worden gebruikt voor deze methode zijn enorm gevoelig voor ruis. Normaal gesproken wordt de Laplacian methode toegepast op een single gray level afbeelding. De filter gaat op zoek naar snelle intensiteit verandering. Bij de Laplacian methode kan de grootte van de kernel worden van tevoren worden berekend en aangepast naar de grootte van de afbeelding, waardoor je een groter gebied rond de pixel gecontroleerd.

## Keuze

Voordeel van Canny edge detectie is dat het een relatief simpele maar precieze methode is. Nadeel is echter dat bij het 'verzachten' met de Gaussian filter je ook de edges minder duidelijk maakt waardoor ze moeilijker te vinden zijn. Er is een grotere kans dat je zwakke edges mist. De Canny methode is ook relatief zwaarder dan de andere methodes.

Gradatie gebaseerde algoritmes als de Prewitt filter hebben meerdere nadelen. Zo is de grootte van kernel vastgesteld en kun je deze niet aanpassen aan de grootte van de afbeelding. Ook is een groot nadeel dat ze heel gevoelig zijn voor ruis. Wel kunnen ze makkelijker de richting van een edge vinden door het gebruik van twee kernels.

Voordeel van Laplacian is dat je maar één berekening nodig hebt doordat de kernel van tevoren wordt berekend. Dus de methode is sneller vanwege de weinige rekenkundige operaties. Een nadeel is echter dat de richting van de edges minder goed te vinden zijn, omdat je maar één kernel gebruikt en niet zowel een horizontale en verticale kernel.

## Implementatie

De edge detection filter zal beginnen met een intensiteit afbeelding (grayscale). Omdat we geen bewerkingen over elkaar willen gebruiken, beginnen we met een blanco afbeelding in de exacte grootte als het origineel.

### **Edge detection:**

We willen gebruik maken van de Laplacian methode met Gaussian filter. Deze methode willen we implementeren zoals hiervoor besproken.

Het resultaat is een grayscale afbeelding met hoe donkerder de pixel hoe duidelijker de edge is. Dit is echter niet heel leesbaar om berekeningen op uit te voeren en heeft daarom nog een threshold na-berekening nodig.

### **Threshold:**

De threshold bewerking geldt per pixel. Er is dan geen extra afbeelding nodig omdat de bewerkingen dan niet overlap hebben.

De functie threshold krijgt een afbeelding mee en een integer waarde met een default van 220. Per pixel wordt er gekeken of de waarde in het bereik 0 tot 255. Minder of is gelijk aan of groter dan. Als de waarde minder is dan de threshold waarde zal de waarde van de pixel op minimaal worden gezet (0). Als de pixelwaarde hetzelfde of meer is dan de threshold waarde.

Na deze stappen hopen we een goed eindresultaat te hebben en de nieuwe nieuwe afbeelding terug te kunnen geven aan de functie.

## Bronnen

<https://www.mathworks.com/discovery/edge-detection.html>

[https://en.wikipedia.org/wiki/Canny\\_edge\\_detector](https://en.wikipedia.org/wiki/Canny_edge_detector)

<http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.301.927&rep=rep1&type=pdf>

<https://www.omicsonline.org/open-access/methods-of-image-edge-detection-a-review-2332-0796-1000150.php?aid=57249>