Luke Roovers & Luuk Steeman

6-4-2017

Een plan van aanpak en een bron van informatie

Sobel & Prewitt

Implementatieplan

Vision

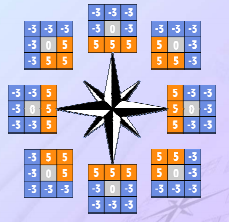
# Doel

Het doel van de implementatie van de Canny edge detection is om een edge detection systeem te realiseren waarvan het edge detection filter kan worden aangepast. Op deze manier kan er getest worden op snelheid, efficiëntie en kwaliteit van het resultaat.

# Methodes

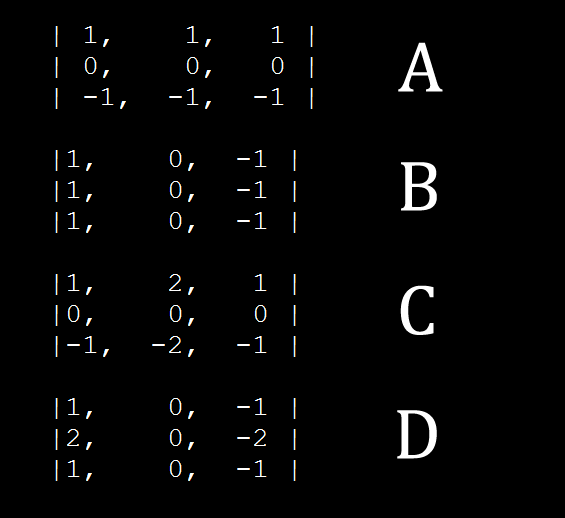
## Edge detection:

**Kirsch compass & Robinson compass**



Kirsch compass & Robinson compass zijn eerste afgeleide methoden om randen te zoeken binnen een afbeelding. Beide methodes maken gebruik van maskers die gedraaid worden om zo randen te vinden vanuit verschillende hoeken. De richtingen komen overeen met de richtingen op een kompas, vandaar de naam. In tegenstelling tot Prewitt en Sobel, die vanuit twee richtingen zoeken. Een verschil tussen beide methodes is dat Robinson coëfficiënten gebruikt van 0, 1 of 2 en Kirsch een vast masker heeft.[[1]](#footnote-1) Met deze methodes is te achterhalen welke richting een edge heeft. De kernels geven een waarde, de kernel met de hoogste waarde bepaald de richting. [[2]](#footnote-2)

**Prewitt & Sobel operators**

****Zowel de Prewitt als de Sobel operator passen 3x3 kernels toe op een afbeelding in de verticale richting en de horizontale richting. In de afbeelding rechts zijn de 4 kernels te zien die deze methode gebruiken. A & B zijn de horizontale en verticale kernels van de Prewitt methode en de C & D zijn de horizontale en verticale kernels van de Sobel methode. Op deze manier worden de horizontale en verticale edges ontdekt.

De beide kernels geven vervolgens een resultaat terug: *Gx & Gy*. Dit is het resultaat van de som van kernels. Beide resultaten worden vervolgens gebruikt in de wiskundige formule die de nieuwe intensiteit van de pixel bepaald. Deze formule is : √(x^2+y^2). Hierbij zorgt de machtsverheffing dat een pixel dus nooit een negatieve waarde kan krijgen. De richting van de pixel is belangrijk om later in Canny thresholding toe te passen. De formule hiervoor is: inverse Tan(x,y)\*180)/pi. [[3]](#footnote-3)[[4]](#footnote-4)

De Prewitt operator werkt op dezelfde wijze, maar gebruikt een andere kernel. Sobel is beter in het detecteren van diagonale edges, terwijl Prewitt beter is in het herkennen van horizontale en verticale edges.[[5]](#footnote-5)

## Canny edge detection

Canny is gewoon Sobel met extra toevoegingen. Deze toevoegingen bestaan uit een Gaussian blur voor noise reduction en na het vinden de randen en de richtingen wordt hysteresis uitgevoerd om duidelijker de lijnen aan te geven met twee thresholds om de lijnen te verdunnen. Daarna wordt nonmaximum suppression toegepast om edges te verdunnen tot één pixel breed.[[6]](#footnote-6)

# Thresholding

**Nonmaximum Suppression**

Bij nonmaximum suppression wordt er langs alle edges gelopen en gekeken naar de pixels die op de normaal van de edge richting liggen. Als de een van de twee pixels naast die edge pixel een hogere intensity heeft dan de pixel die dan gemeten wordt, wordt de pixel waarde op 0 gezet en is het geen edge meer. Hierdoor wordt alleen de sterkste edges meegenomen.  
  
Samengevat

|  |  |  |
| --- | --- | --- |
| Edge detection methode (kernel) | Voordelen | Nadelen |
| Sobel | Relatief snel | Minder resultaat op diagonale edges |
| Prewitt | Relatief snel | Minder resultaat op horizontale/verticale edges |
| Kirsch & Robinson | Edge richting wordt uitgerekend door de kernel | Kosten heel veel tijd. |
| Canny edge detection | Verzameling van Sobel en verschillende thresholdings, 1 oplossing. | Gevoelig voor Noise |
| Hough transform | Goed in opmerken van bepaalde vormen | Detecteren van subtiele edges is niet het sterkste punt |

Keuze  
  
Na een diepgaand methodisch onderzoek is er besloten te focussen op de Canny edge methode. Dit omdat deze methode te boek staat als de meest relevante oplossing in het edge detection verhaal. Het filter dat gebruikt wordt, Sobel, wordt in verschillende bronnen geprezen als de snelste oplossing met het beste resultaat samen met het Prewitt filter. Dit werpt al gelijk een mooie onderzoeksvraag op: “Wat geeft een beter resultaat met betrekking tot gezicht herkenning, Prewitt of Sobel?”.  
  
Tijdens het onderzoek zijn we op vele methodes gestuit voor het detecteren van edges, niet alle zijn echter van toepassing op gezicht herkenning. Er zijn ook methodes die betere resultaten geven waar het vormen betreft, maar die zijn voor ons niet relevant.

De keuze is dan ook gemaakt om voor implementatie van de Canny edge detection.

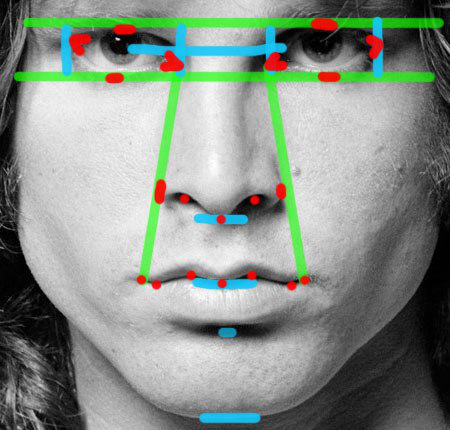
We hebben gekozen om het bij deze twee filters te houden met het oog op de haalbaarheid, de code dynamisch maken en meerdere filters ( ook filters die meerdere kanten op bewegen ) is een mooie extra maar niet een streefdoel op dit moment.

# Implementatie

De implementatie van de Canny edge detection gebeurd binnen de bestaande software. Er zijn functies voor de verschillende convoluties die zorgen dat er een kopie van de originele Image wordt aangepast door middel van neighbourhood convoluties. Het resultaat hiervan komt bij de stepThresholding functie. Hierin wordt de hysteresis functie aangeroepen. Hierbij worden eerste alle edge strengths vergeleken met een upper treshold, als een edge daarboven zit, wordt deze op 255 gezet. Vervolgens wordt er nog een keer langs al die edges gelopen in de richting van de edge. Elke pixel die in die richting zit en boven de lower threshold zit, is ook een edge en krijgt de waarde 255. Nonmaximum supression is een erg vergelijkbare functie. Er wordt opnieuw langs alle edges gelopen, maar dit keer wordt er gekeken naar de pixels die op de normaal lijn van de edge staan op 1 pixel afstand. Als een van deze pixels hogere edge strength heeft dan de pixel van waaraf je kijkt, wordt de eerste pixel op 0 gezet en is geen edge meer.  
  
De aannames over de verschillende technieken zijn hierboven te vinden in de tabel.

Evaluatie  
  
Na het afsluiten van dit project is het bedoeling dat de hoofdvraag is beantwoord. De hoofdvraag wordt als volgt geformuleerd: “Welke kernel ( namelijk Sobel of Prewitt ) levert het beste resultaat qua tijd en kwaliteit in gezichtsherkenning software “

Wat er hierbij gemeten moet worden is de tijd die verstrijkt tijdens de convolutie van de kernel over het de image, om te kijken of welke van de twee kernel sneller is. Vervolgens gaan we de resultaten ( de image met alle edges ) verwerken in een enquête en die voorleggen aan meerdere testpersonen.   
  
De verwachting van het resultaat is als volgt:   
“Prewitt werkt sneller dan Sobel”. Dit is gebaseerd op het feit dat Sobel in principe meer berekeningen moet doen, met het oog op de kernel.   
  
“Sobel geeft een beter resultaat dan Prewitt”. De vage herinnering van portret schetsen op de middelbare school komt hier boven drijven. Een menselijk hoofd bestaat uit lijnen, en de neus ogen mond liggen allemaal op zo’n lijn. Hierdoor is de verwachting dat de kracht van Sobel, namelijk horizontale en verticale lijnen, meer tot zijn recht komt in gezicht herkenning dan Prewitt.

(Rolf, sd)

# Bibliografie

Computerphile. (2015, november 4). *Youtube*. Opgehaald van Computerphile Kanaal: https://www.youtube.com/channel/UC9-y-6csu5WGm29I7JiwpnA

gruup, e. d. (2017, maart 6). *cs.tau.ac.il*. Opgehaald van http://www.cs.tau.ac.il/~turkel/notes/sem.pdf

International Journal of Scientific & Engineering Research. (2017, april 6). *ijser*. Opgehaald van www.ijser.org: http://www.ijser.org/researchpaper/Analytical-Comparison-between-Sobel-and-Prewitt-Edge-Detection-Techniques.pdf

Kuntz, N. (2006). Opgehaald van Canny tutorial: http://www.pages.drexel.edu/~nk752/cannyTut2.html

Latecki, L. J. (2017, april 6). *lecturer*. Opgehaald van http://lecturer.ukdw.ac.id/~mahas/dossier/comvis\_06a.pdf

Rolf, P. (sd). *Leer een portret tekenen in 8 stappen*. Opgehaald van flor.nl: http://flor.nl/leer-zelf-tekenen-portret.php

Tutorialspoint. (2017, maart 6). *Robinson Compass Mask*. Opgehaald van Tutorialspoint: http://www.tutorialspoint.com/dip/Robinson\_Compass\_Mask.htm

1. (Tutorialspoint, 2017) [↑](#footnote-ref-1)
2. (gruup, 2017) [↑](#footnote-ref-2)
3. (International Journal of Scientific & Engineering Research, 2017) [↑](#footnote-ref-3)
4. (Computerphile, 2015) [↑](#footnote-ref-4)
5. (Latecki, 2017) [↑](#footnote-ref-5)
6. (Kuntz, 2006) [↑](#footnote-ref-6)