



MEETRAPPORT CANNY VS. LAPLACIAN

Welk algoritme kan beter gezichten herkennen?

Nathan Houwaart en Jochem van Weelde
V2A

1. Inleiding

Dit onderzoek is uitgevoerd in opdracht van de Hogeschool Utrecht voor het vak Vision. Het doel voor dit vak is een kleinschalig onderzoek doen naar een bepaald onderdeel van de pipeline van gezichtsherkenning. Daarbij is een hoofddoel opgesteld met een kwestie en gewenste situatie:

De toepassing waarvoor dit onderzoek dient.

Gebruikers kunnen foto's uploaden op Facebook. Facebook gebruikt gezichtsherkenning om te kijken of er gezichten aanwezig zijn op de foto en tekent vervolgens een box om elk gezicht heen. De gebruiker kan deze personen "taggen" in de foto. Doordat dit offline kan gebeuren (Niet real-time), is accuraatheid belangrijker dan snelheid. Door gezichten te analyseren in foto's kunnen die ook gecategoriseerd worden op personen.

De kwestie:

Als er een foto op een onlinewebsite wordt geüpload van waar personen op afgebeeld staan, moet de gebruiker een makkelijke manier tot zijn beschikking hebben om de personen in de afbeelding te "taggen".

De gewenste situatie:

Als een persoon een foto uploadt, moet het programma een blokje om gezichten tekenen zodat de gebruiker die de foto uploadt, makkelijk personen kan "taggen" in de foto's.

In dit onderzoek is er gekeken naar het verbeteren van Edge Detection binnen de pipeline van gezichtsherkenning.

2. Meetrapport

2.1. Doel

Het overkoepelende hoofddoel van dit onderzoek is een offline gezichtsherkenning toepassing te ontwikkelen met hoge accuraatheid. Hierbij wordt in het specifiek gekeken naar de voorbewerking van afbeeldingen middels edge detection. In dit meetrapport zal worden gekeken naar twee methoden van edge detection: de Canny Edge Detection methode en de Marr-Hildreth Edge Detection methode.

De basis software die gebruikt wordt voor het herkennen van gezichten, is de HU-Vision-1819-Base software die beschikbaar is gesteld door de Hogeschool Utrecht ("HU-Vision-1819-Base (Commit 21) [Computerprogramma]", 2019). De software is voor de implementatie van Canny Edge en Marr-Hildreth Edge detection deels gemodificeerd.

In dit onderzoek wordt er gekeken naar hoeveel gezichten de gezichtsherkenningsssoftware kan herkennen met behulp van preprocessing van beide algoritmen. Daarnaast wordt er gekeken naar de zekerheid dat het algoritme geeft bij het herkennen van een al bekend gezicht. Hieronder volgt een kleine toelichting daarop:

Als een gezicht wordt herkend, wordt de afbeelding in het geheugen van de gezichtsherkenningsssoftware opgeslagen. De herkende afbeelding wordt opnieuw door het algoritme gehaald. Vervolgens wordt er gekeken naar de 'zekerheid' die het programma geeft bij het matchen van de nieuwe afbeelding t.o.v. het gezicht dat al opgeslagen staat in het geheugen van het programma.

Voor de implementatie van Canny Edge detection is gebruik gemaakt van de volgende bronnen:

- (Sahir, 2019) <https://towardsdatascience.com/canny-edge-detection-step-by-step-in-python-computer-vision-b49c3a2d8123>
- (Wang, 2013) <http://fourier.eng.hmc.edu/e161/lectures/canny/node1.html>

Voor de implementatie van Marr-Hildreth Edge detection is gebruik gemaakt van de volgende bronnen:

- (OpenCV, 2019) https://docs.opencv.org/2.4/doc/tutorials/imgproc/imgtrans/laplace_operator/laplace_operator.html

2.2. Hypothese

Er wordt verwacht dat de gezichtsherkenningsssoftware in combinatie met het Canny Edge algoritme meer gezichten kan herkennen dan met de Margreth-Hill Edge Detection methode. Daarnaast wordt er verwacht dat het Canny Edge algoritme een hogere accuraatheid geeft bij het matchen van opgeslagen gezichten in de database van het programma. Deze hypothese is gebaseerd op het vooronderzoek dat gedaan is voor het implementatieplan (Houwaart & Van Weelde, 2020).

2.3. Werkwijze

Om te kijken welke van de twee edge detection methoden superieur is aan de ander, zijn de volgende tests opgezet:

De herkenningstest

Een set van honderd foto's van personen (zelfportret) worden door zowel het Canny Edge Detection algoritme gehaald als door het Marr-Hildreth Edge Detection algoritme. De afbeeldingen van personen komen van de website <https://www.thispersondoesnotexist.com> (Karras et al., 2020). De code die wordt gebruikt om dit te doen staat in de volgende repository op GitHub: <https://github.com/jochemvanweelde/HU-Vision-1920-Nathan-Jochem> (Van Weelde & Houwaart, 2020).

Elke foto geeft een bepaald resultaat:

- 1) **Localization step 1-2 failed:** Programma herkent geen gezicht.
- 2) **Localization step 3-4-5 failed:** Programma herkent een gezicht, maar kan de mond en/of ogen en/of neus niet vinden.
- 3) **Extraction step 1-5 failed:** Programma herkent een gezicht met alle gezichtscomponenten, maar foto kan niet worden geëxtraheerd.
- 4) **Geslaagd!** Het programma herkent een gezicht en geeft een lijst met facial parameters

De afbeeldingen die worden herkend door de software, zullen worden opgeslagen in het geheugen van het programma.

De matching test

In het oorspronkelijke implementatieplan was er een matching test beschreven. Deze is hier nader beschreven:

Nadat alle afbeeldingen door het algoritme zijn gehaald, worden de herkende afbeeldingen nogmaals door het algoritme van de software gehaald. Nu zal er gekeken worden naar de zekerheid dat het programma geeft bij het matchen van de input afbeelding met een gezicht dat al opgeslagen staat in de database. Dit geeft een bepaald resultaat:

- *Een getal tussen de 0 en de X. Waar 0 aangeeft dat het een match heeft gevonden met geen verschillen met de input afbeelding en waar X aangeeft dat de gevonden match afwijkingen bevat, maar dat dat toch de beste match is die de software kon vinden.*

Echter blijkt dat de resultaten die deze test op zullen gaan leveren niet relevant zijn voor het hoofddoel waarvoor dit onderzoek in gang is gezet. Het hoofddoel is om gezichten te kunnen herkennen. Niet zozeer het matchen van input afbeeldingen met afbeeldingen uit een database. Om deze reden wordt de matchingtest niet uitgevoerd.

Door alle honderd foto's door de edge detection algoritmes heen te halen, is het mogelijk om te bepalen welk algoritme het beste werkt met de huidige gezichtsherkenningsoftware. Er wordt gekeken naar het aantal foto's dat wel wordt herkend. Meer foto's die worden herkend in het algoritme betekent dat het beter gezichten kan herkennen in de set foto's die gebruikt zijn.

2.4. Resultaten

Canny Edge Detection

Canny edge heeft veel input parameters die veranderd kunnen worden die de output afbeelding beïnvloeden. Door Evolutionary Optimization is er gezocht naar een parametercombinatie die het beste resultaat geeft. Evolutionary Optimization pakt de parameters van de beste resultaten. Deze worden gebruikt in de volgende tests om zo te kunnen kijken of er nog een beter resultaat te bereiken is (Vikhar, P. A. 2016). Een resultaat is goed wanneer het programma duidelijke contouren van het gezicht genereert die bijna perfect overeenkomen met de origine afbeelding.

De Canny Edge Tests zijn uiteindelijk uitgevoerd met de volgende parameters:

- Sigma (Gaussian Blur) = 1,4
- Sobel filters voor x en y richting + beide diagonale richtingen:

```
cv::Mat sobel_x = (cv::Mat_<double>(3, 3) <<  
    1, 0, -1,  
    2, 0, -2,  
    1, 0, -1);
```

```
cv::Mat sobel_d1 = (cv::Mat_<double>(3, 3) <<  
    0, 1, 2,  
    -1, 0, 1,  
    -2, -1, 0);
```

```
cv::Mat sobel_y = (cv::Mat_<double>(3, 3) <<  
    1, 2, 1,  
    0, 0, 0,  
    -1, -2, -1);
```

```
cv::Mat sobel_d2 = (cv::Mat_<double>(3, 3) <<  
    -2, -1, 0,  
    -1, 0, 1,  
    0, 1, 2);
```

- Double Threshold:
 - o Weak threshold van 20 (max. 255)
 - o Strong threshold van 60 (max. 255)

Hieronder volgt een de output van één van de honderd gezichten dat door het Canny Edge algoritme is gehaald.



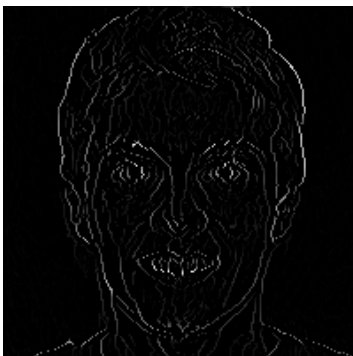
Figuur 1: Originele afbeelding. Onbewerkt



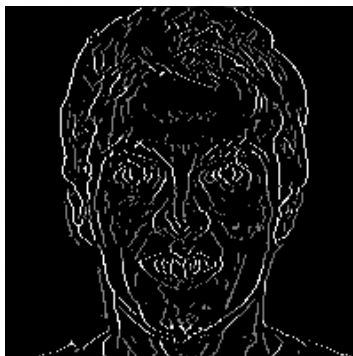
Figuur 2: Afbeelding met Gaussian Blur



Figuur 3: Sobell filter



Figuur 4: Edge Thinning



Figuur 5: Double threshold



Figuur 6: Final image

Marr-Hildreth Edge Detection

De Marr-Hildreth Tests zijn uitgevoerd met de volgende parameters. Deze zijn ook gebaseerd op de Evolutionary Optimisation methode (Vikhar, P. A. 2016).

- Sigma (Gaussian Blur) = 1,4
- Kernel grootte = 3x3



*Figuur 7: Originele afbeelding.
Onbewerkt*



Figuur 8: Afbeelding met Gaussian Blur



Figuur 9: Laplacian filter / final image

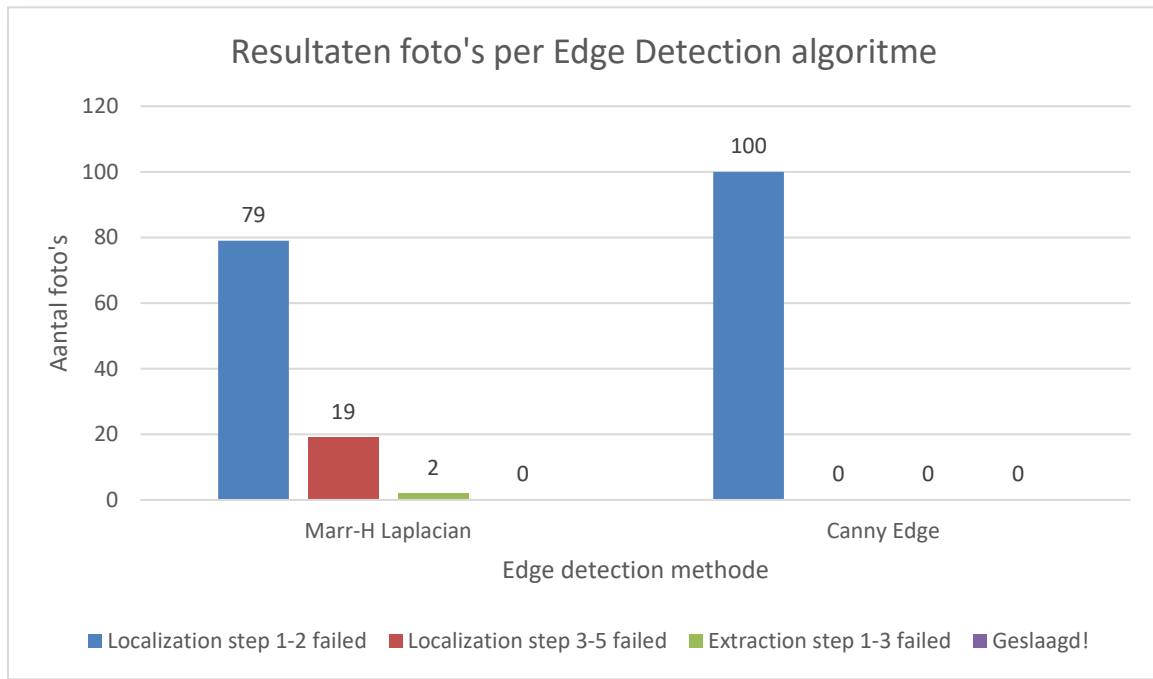
Na het uitvoeren van de herkenningstest zijn de volgende resultaten waargenomen:

	Localization step 1-2 failed	Localization step 3-5 failed	Extraction step 1-3 failed	Geslaagd
Marr-H Laplacian	79 afbeeldingen	19 afbeeldingen	2 afbeeldingen	0 afbeeldingen
Canny Edge	100 afbeeldingen	0 afbeeldingen	0 afbeeldingen	0 afbeeldingen

In de herkenningstest is geen van de honderd testafbeeldingen succesvol herkend. Daardoor is er geen mogelijkheid geweest de matching test uit te voeren; er zijn geen afbeeldingen om te matchen.

2.5. Verwerking

De resultaten van de herkenningstest zijn verwerkt in het onderstaande diagram.



In de grafiek is te zien dat het programma m.b.v. het Canny Edge algoritme in staat was 0 gezichten te herkennen, terwijl het programma m.b.v. het Marr-Hildreth algoritme in staat was 21 gezichten te herkennen. Hiervan konden 2 gezichten worden herkend met alle gezichtscomponenten, maar waarbij de extractie van de gezichtscomponenten faalde.

De resultaten van de tests zijn onverwacht en verrassend. In de hypothese werd voorspeld dat Canny edge beter zou presteren dan Marr-Hildreth. Er zijn onderzoeken (Maini, R., & Aggarwal, H., 2009) & (Nadernejad, E., Sharifzadeh, S., & Hassanpour, H., 2008) gedaan naar de voor- en nadelen van verschillende edge detection technieken. Daaruit blijkt dat Canny Edge Detection de best presterende edge detection methode is, maar wel veel rekenkracht vereist.

Echter heeft het programma geen gezichten kunnen herkennen met Canny Edge Detection. Het Marr-Hildreth algoritme heeft ook geen volledige gezichten kunnen herkennen maar kwam wel verder in de pipeline. Het lijkt erop dat de gebruikte software, met de huidige pipeline, niet geschikt is voor het herkennen van gezichten in afbeeldingen die bewerkt zijn met Canny Edge Detection. Daarnaast lijkt het er ook op dat het Marr-Hildreth algoritme ook niet volledig geschikt is voor deze software, ondanks dat het algoritme verder in de pipeline komt.

2.6. Conclusie

Op basis van de uitgevoerde tests kan er geconcludeerd worden dat de huidige gezichtsherkenningsssoftware niet geschikt is voor de implementatie van het Canny Edge Detection algoritme en het Marr-Hildreth Edge Detection algoritme.

2.7. Evaluatie

Het doel van dit onderzoek was om te bepalen welk van de twee edge detection algoritmen, Canny Edge of Marr-Hildreth, beter is dan de ander. Dit werd onderzocht voor het overkoepelende hoofddoel: een Offline gezichtsherkenning toepassing te ontwikkelen met hoge accuraatheid.

Uit dit onderzoek blijkt dat de verkregen software niet geschikt is voor het herkennen van gezichten met behulp van het Canny Edge Detection en het Marr-Hildreth Edge Detection algoritme. Daardoor is het niet mogelijk geweest om te bepalen welk van de twee algoritmen beter presteert dan de ander.

Deze conclusie is echter snel getrokken; het Canny Edge algoritme is een algoritme met veel verschillende parameters die aangepast kunnen worden. Aan de hand van deze parameters zal het resultaat van de output afbeelding worden beïnvloed. Het kan zijn dat er een parametercombinatie is die een afbeelding creëert die wel bruikbaar is voor de gebruikte gezichtsherkenningsssoftware. Om met honderd procent zekerheid te kunnen zeggen dat de conclusie van dit onderzoek klopt, moet daarom een vervolgonderzoek gedaan moeten worden naar de parametercombinaties van het Canny Edge algoritme samen met de gebruikte gezichtsherkenningsssoftware.

Daarnaast zou er een vervolgonderzoek gedaan kunnen worden naar een andere kernel grootte voor het Marr-Hildreth algoritme en een andere sigma voor de Gaussian Blur.

Tot slot zou er een onderzoek gedaan kunnen worden naar een verbetering van de gebruikte software. Hoe kan de pipeline van de software dusdanig aangepast worden dat het wel goed werkt in combinatie met Canny Edge en/of Marr-Hildreth?

2.8. Bronnen

- Houwaart, N. M., & Van Weelde, J.N.H. (2020). *Implementatieplan Edge Detection onderzoek*. Geraadpleegd van https://hogeschoolutrecht-my.sharepoint.com/:b:/g/personal/nathan_houwaart_student_hu_nl/EbtMjzpwDT5KhMnQNYFIAJkBQnNlpFzV_GjZD_IejwCoCg?e=ebEnTh
- HU-Vision-1819-Base (Commit 21) [Computerprogramma]*. (2019). Geraadpleegd van <https://github.com>
- Karras, T., Laine, S., Aittala, M., Hellsten, J., Lehtinen, J., & Aila, T. (2020, 5). *This Person Does Not Exist*. Geraadpleegd op 20 2020, van <https://www.thispersondoesnotexist.com/>
- Maini, R., & Aggarwal, H. (2009). Study and comparison of various image edge detection techniques. *International journal of image processing (IJIP)*, 3(1), 1-11.
- Nadernejad, E., Sharifzadeh, S., & Hassanpour, H. (2008). Edge detection techniques: evaluations and comparisons. *Applied Mathematical Sciences*, 2(31), 1507-1520.
- OpenCV (2020, December 31) *Laplace Operator*. Geraadpleegd van https://docs.opencv.org/2.4/doc/tutorials/imgproc/imgtrans/laplace_operator/laplace_operator.html
- Sahir, S (2019, January 25) *Canny Edge Detection Step by Step in Python — Computer Vision*. Geraadpleegd van <https://towardsdatascience.com/canny-edge-detection-step-by-step-in-python-computer-vision-b49c3a2d8123>
- Van Weelde, J., & Houwaart, N. M. (2020, 22). *Jochemvanweelde/HU-Vision-1920-Nathan-Jochem*. Geraadpleegd op 22 2020, van <https://github.com/jochemvanweelde/HU-Vision-1920-Nathan-Jochem>
- Vikhar, P. A. (2016). "Evolutionary algorithms: A critical review and its future prospects". *Proceedings of the 2016 International Conference on Global Trends in Signal Processing, Information Computing and Communication (ICGTSPICC)*. Jalgaon. pp. 261-265. doi:10.1109/ICGTSPICC.2016.7955308. ISBN 978-1-5090-0467-6.
- Wang, R (2013, September 25) *Canny Edge Detection*. Geraadpleegd van <http://fourier.eng.hmc.edu/e161/lectures/canny/node1.html>