

A dark blue vertical bar runs along the left edge of the page. A blue arrow-shaped banner points to the right from this bar, containing the date. In the bottom-left corner, several thin, curved lines in dark blue and light grey sweep upwards and to the right.

2/13/2020

Implementatieplan

Plan van implementatie

Kevin Patist, Wietse ten Dam

Table of Contents

Doel.....	2
Hoofdvraag.....	2
Methoden.....	3
Canny edge detection	3
Canny met Gaussian filter	3
Sobel operator	3
Prewitt.....	4
Fuzzy logic.....	4
Voor- en nadelen	4
Keuze.....	5
Implementatie	6
Evaluatie.....	7
Accuratie	7
Snelheid.....	7
Memory efficiency	7
Robuustheid	7
Wat gaan we doen	7
Hypothese	7
Bronnen.....	8

Doel

- Om van langs lopende mensen de gezichten te kunnen vinden en daarop herkenning te kunnen uitvoeren.
- Tijdens de epidemie van het coronavirus willen we graag een camera met gezichtsherkenning en een warmte camera samen laten werken. Dan kunnen we snel de mensen identificeren die een te hoge lichaamstemperatuur hebben zodat die zo snel mogelijk medische hulp krijgen.
- De warmte camera en het koppelen daarvan doen wij niet, wij zorgen voor een programma dat gezichten detecteert en herkent. Wij hebben geen live view wij krijgen foto's van mensen binnen en detecteren een gezicht en voeren herkenning daarop uit. Om dit optimaal te kunnen toe te passen moeten wij een gang hebben met goede verlichting en een camera in het midden van de gang.
- Wij willen de edge detection verbeteren om de volgende reden: Als er betere edge detection is krijg je een accurater beeld van de gezichtskenmerken wat het makkelijker kan maken om de persoon te kunnen detecteren en te kunnen herkennen.

Hoofdvraag

Hoe kunnen wij de huidige edge detection verbeteren zodat wij een hogere accuraatheid krijgen?

Methoden

Canny edge detection

Bij Canny edge detection word de output van de Sobel operator gebruikt. De Sobel operator is ook een edge detection algoritme maar Canny perfectioneert die uitkomst. De output van Sobel ziet er meestal wat minder goed uit dan Canny omdat de lijnen dikker zijn. Canny neemt de uitkomst van Sobel en gaat daarop aan de slag. Alle lijnen die Sobel heeft gemaakt worden door Canny verdunt tot een dikte van 1 pixel. Dit zorgt voor duidelijkere en preciezere lijnen. Canny maakt gebruik van de lijnen van Sobel en zoekt voor een local optimum, waar de objecten zich echt scheiden en legt daar de 1 pixel dikke lijn op. Door dit toe te passen heb je na het goede algoritme van Sobel wat extra detail.

Het is ook belangrijk om een threshold toe te voegen aan de Canny edge detection omdat als je dat niet doet kan die alle kleine in perfecties ook zien als een lijn en dat willen we niet. Door een threshold toe te voegen kan je alleen maar lijnen toestaan boven een bepaalde kracht. Een dubbel threshold is nog beter om toch wat meer detail te behouden, Een dubbel threshold betekend dat je een top waarde kiest en een laag waarde. Alles wat boven de top waarde zit word sowieso getoond daarin tegen alles wat onder de laag waarde zit word sowieso niet getoond. Tussen de top en laag lijn zit een grijs gebiedje, Alles wat in die grijze laag ligt en alles wat vast zit aan de top laag word wel getoond maar als het niet aan de top laag vast zit en dus aan de laag lijn, dan word het niet getoond. Zo word de edge detectie toch preciezer maar blijf je toch de kleine in perfecties voor.

Dit is afkomstig van "Canny edge detection" <http://justin-liang.com/tutorials/canny/>

Canny met Gaussian filter

In https://link.springer.com/chapter/10.1007/978-3-642-33478-8_7 "The Canny Edge Detection and Its Improvement" staat dat Gaussian filter ervoor kan zorgen dat de grijs levels smoother over gaan in elkaar en dat je dan false hits makkelijker kan tegen gaan. Een false hit is natuurlijk niet wat je wil, een Gaussian filter is een tool om het plaatje een beetje onscherp te maken om zo alle kleine lijntjes die als edge gezien zouden kunnen worden, worden vervaagt en blijven echt alleen de dikke echt edges over. Een edge detectie programma kan dan beter zijn best doen en ook accurater zijn

Sobel operator

De sobel operator gebruikt twee masks, een voor de x as en een voor de y as. Hiermee zoekt hij op de desbetreffende as naar edges. Als beide masks over de image zijn gehaald wordt het resultaat samengevoegd.

https://www.researchgate.net/publication/297736749_Comparison_of_Edge_Detection_Techniques

Prewitt

Prewitt is zeer gelijk aan de Sobel operator en is gebruikt om randen te detecteren op de x en y assen. Prewitt legt in tegenstelling tot Sobel minder de nadruk op de pixels in de buurt van het midden. De kernel is bij Prewitt heeft gelijke nummer met min en plus maar sobel heeft meerdere nummers ook in plus en min.

ijser.org/researchpaper/Edge-Detection-by-Using-Canny-and-Prewitt.pdf

Fuzzy logic

Fuzzy logic is heel anders dan de andere soorten van edge detectie die wij gezien hebben hiervoor. Fuzzy logic werkt op basis van fuzzyed data en voert daar zijn operaties op uit. Fuzzyed data is data die door een bepaald filter is gehaald om de operaties er op te versimpelen, bijvoorbeeld als je 100.000 euro zo fuzzyen dan komt er te staan dat het heel duur is. Met plaatjes gaat het ook zo, er word veel specifieke data versimpeld naar simpele waarden. Hierna worden de member values aangepast, Hier word de echte edge detectie gedaan op de versimpelde waardes. Nu word het plaatje weer defuzzyed en krijg je de edge's. Deze informatie komt van "Edge Detection in Digital Images Using Fuzzy Logic Technique"<https://www.freeprojectsforall.com/wp-content/uploads/2018/10/Edge-Detection-Techniques-using-Fuzzy-Logic.pdf>

Voor- en nadelen

Methode	Voordelen	Nadelen
Canny edge detection	Hele scherpe edge detection	Sobel of Prewitt moet sowieso geïmplementeerd worden. Lastig te implementeren. Kost meer tijd en resources.
Canny met Gaussian filter	Het plaatje word soft gemaakt waardoor de ruis minder invloed heeft. Hele goede edge detection en reactie op randen. Zeer precies	Sobel moet en het Gaussian filter. Lastig om te implementeren Kost veel processor tijd Is niet tijd efficiënt.
Sobel operator	Edge detection met lichte smoothing Kost weinig tijd/resources Detecteert ook de oriëntatie van de randen	Werkt minder goed als er meer ruis is.
Prewitt	Detecteert de randen en de oriëntatie van de egde. Simpel	Hij is niet zo accuraat en werkt slecht met ruis
Fuzzy logic	Het is een efficiënt algoritme omdat alle waarden makkelijk worden geïnterpreteerd	Het is niet zo precies als Canny

Keuze

Er is gekozen voor het gebruik van de Sobel operator omdat het een efficiënt en snelle methode is. Sobel is boven Prewitt gekozen omdat Sobel ook automatisch wat smoothing toepast op de afbeelding, dit zorgt ervoor dat het wat minder gevoelig is voor ruis. Ook kunnen we de kernel van Sobel aanpassen waardoor het flexibeler is dan Prewitt. Canny en fuzzy logic zien er ook interessant uit maar in verband met de tijd en verwachtingen voor het implementeren van Canny en fuzzy logic is besloten het bij Sobel te laten.

Implementatie

Om de Sobel operator te gaan gebruiken moet er een C++ implementatie geschreven worden. Dit zal gedaan worden door de volgende bestanden aan te passen:

- StudentPreProcessing.h
- StudentPreProcessing.cpp

Binnen deze bestanden zullen de `stepEdgeDetection` en `stepThresholding` functies worden aangepast. In de `stepEdgeDetection` functie komt een nieuwe implementatie van Sobel en de `stepThresholding` wordt een `thresholding` functie die afgestemd staat om optimaal samen te werken met de `stepEdgeDetection` functie. Ook is het mogelijk dat de private variabelen van de `StudentPreProcessing` klasse aangepast worden als er wordt ontdekt dat dit nodig is voor het goed implementeren van Sobel.

Om de nieuwe Sobel implementatie optimaal te laten werken gaat er onderzocht worden wat de beste kernel en `thresholding` waarden zijn. De uitkomst van dit onderzoek bepaald welke kernel en `thresholding` waarden in het eindproduct gebruikt gaan worden.

Bij deze implementatie nemen wij aan dat de rest van het facial recognition proces in werkende staat is.

Evaluatie

Accuratie

De accuratie is bij onze applicatie de focus en zal hierom de meeste aandacht krijgen tijdens het maken van de nieuwe edge detection.

Snelheid

De snelheid van het programma moet ook redelijk zijn. Als er mensen langs lopen moet het programma snel genoeg zijn om de mensen bij te houden. De laplacian operator die nu wordt gebruikt is al snel genoeg. De sobel operator als deze goed geïmplementeerd wordt zal ook rond deze snelheid zitten en misschien zelfs erboven. De snelheid is dus geen prioriteit voor onze implementatie.

Memory efficiency

Omdat er genoeg resources beschikbaar zijn is de memory efficiency niet heel belangrijk voor onze implementatie. Hierom is het mogelijk dat onze implementatie minder memory efficiënt is dan de huidige implementatie.

Robuustheid

Het systeem moet natuurlijk ook robuust werken. We willen niet dat mensen met het corona virus ontsnappen. De robuustheid van het systeem moet een groot aantal van de mensen kunnen herkennen.

Wat gaan we doen

Zoals hier boven staat uitgelegd is een hogere accuratie waar de grootste focus op ligt binnen dit project. Om aan te tonen dat de nieuwe edge detection beter is in dit aspect krijgen de programma's met beide edge detection methoden 4 setjes van 25 duidelijke gezichtsphotos om de gezichten erin te herkennen. Daarvan berekenen we een gemiddelde van elk programma en berekenen we een gemiddelde afwijking. De edge detection in het programma dat de het hoogste gemiddeld en de laagste afwijking heeft als de edge detection met de hoogste accuratie.

Voor het testen gaat er eerst geëxperimenteerd worden met verschillende kernels en thresholding waarden. Hiervoor zullen we 5 verschillende kernels en 5 verschillende threshold waarden kiezen die we met elkaar gaan mixen. Zo ontstaan er uiteindelijk 25 tests die we door een set van tien foto's laten gaan om gezichten te herkennen. De combinatie met de meeste goed herkende gezichten wordt gebruikt als waarden voor het eindproduct. Als de twee beste combinaties eindigen met hetzelfde aantal zullen deze beide een tweede set foto's krijgen om hiermee te bepalen welke de beste is.

Hypothese

Over de beste kernel en thresholding waarde hebben wij nog geen idee, omdat wij hier beide nog geen ervaring mee hebben. Wel verwachten we dat de nieuwe edge detection beter gezichten zal herkennen dan de huidige versie. Dit omdat wij van onze docent hebben gehoord dat de huidige edge detection heel slecht is geïmplementeerd en het niet moeilijk is iets beters te maken.

Bronnen

Liang, Justin, J. (15-02-2020). Canny edge detection. Geraadpleegd van <http://justin-liang.com/tutorials/canny/>

Ma, Xiaoju, X. (16-02-2020). The Canny Edge Detection and Its Improvement. Geraadpleegd van https://link.springer.com/chapter/10.1007/978-3-642-33478-8_7

Niet gespecificeerd. (21-02-2020). Edge Detection by Using Canny and Prewitt. Geraadpleegd van ijser.org/researchpaper/Edge-Detection-by-Using-Canny-and-Prewitt.pdf

Abdallah, A. (21-02-2020). Edge Detection in Digital Images Using Fuzzy Logic Technique. Geraadpleegd van <https://www.freeprojectsforall.com/wp-content/uploads/2018/10/Edge-Detection-Techniques-using-Fuzzy-Logic.pdf>