

1. Implementatieplan Thresholding

1.1. Namen en datum

Stein Bout

Nick Swaerdens

Laurens van der Sluis

09-03-2018

1.2. Doel

Na edge-detection zijn de randen nog vrij lastig te zien. Het doel is om met thresholding de echte randen beter naar voren te halen en om zwakke randen en ruis er uit te filteren. Nauwkeurigheid heeft de hoogste prioriteit.

1.3. Methoden

Basic thresholding

Dit is de meest simpele vorm van thresholding. In het begin wordt er een vaste threshold waarde ingesteld. Dan wordt elke pixel langsgelopen en gekeken of de pixelwaarde hoger is dan de ingestelde threshold waarde. Als dit het geval is dan wordt deze pixel maximaal wit gemaakt (waarde 255). In het geval dat de pixelwaarde lager is dan de ingestelde threshold waarde wordt de pixel uitgezet (waarde 0). Hoewel dit heel simpel te implementeren is geeft het niet zulke goede resultaten. Bovendien moet voor elke afbeelding handmatig een threshold waarde ingesteld worden. Dit is niet alleen onhandig, maar ook lastig. Zet de threshold waarde te hoog en er worden randen gemist, zet hem te laag en er komen verkeerd gedetecteerde randen mee. Hoe de thresholding waarde ook wordt ingesteld, er zal altijd een compromis gemaakt moeten worden.

Dynamic thresholding

Met dynamic thresholding wordt het grootste probleem van basic thresholding opgelost, namelijk het bepalen van de threshold waarde. Aan het begin wordt de afbeelding een keer doorlopen en wordt er bijgehouden wat de hoogste pixelwaarde is. Aan de hand van deze waarde wordt de threshold waarde bepaald. Er moet nog wel een vaste threshold ratio vastgesteld worden. Deze waarde zal één keer bepaald moeten worden afhankelijk van welke edge-detectie methode er gebruikt wordt. Hoewel dit het probleem voor het bepalen van de threshold waarde wegneemt, zal er nog steeds een compromis gesloten moeten worden. De perfecte threshold waarde bestaat niet. Net zoals bij Basic thresholding, zet de thresholding waarde te hoog en er worden randen gemist, zet hem te laag en er komen verkeerd gedetecteerde randen mee.

Double thresholding

Double thresholding is eigenlijk een dubbele implementatie van basic thresholding. Voor double thresholding zijn er ook twee threshold waarden nodig. Één voor de sterke randen(high-threshold) en één voor de zwakke randen(low-threshold). Elke pixel in de afbeelding wordt weer doorlopen. Als de pixel waarde dan hoger is dan de ingestelde high-threshold waarde, wordt deze pixel net zoals bij basic thresholding maximaal aangezet(waarde 255). Maar als dit niet het geval is dan wordt de pixelwaarde ook nog vergeleken met de low-threshold waarde. Als de pixel waarde dan hoger is dan de low-threshold waarde wordt deze pixel grijs gemaakt(waarde 128). Is de pixelwaarde ook niet hoger dan de low-threshold waarde dan wordt de pixel wel uitgezet(waarde 0). Door dit op deze manier te doen worden de sterke randen er heel netjes uit gehaald maar vallen de zwakke randen niet weg. Double thresholding is ook te combineren met Dynamic

thresholding. Op die manier hoeven de twee threshold waarden niet met de hand uitgezocht te worden.

Adaptive thresholding

Bij adaptive thresholding wordt de threshold waarde per pixel uitgerekend. Hiervoor wordt er naar de omliggende pixels gekeken. In het begin wordt er aangegeven hoe groot de regio is waarin er per pixel wordt gekeken. Met een regio van 5x5 zal er dus gekeken worden naar de 24 omliggende pixels. Om de threshold waarde te bepalen is het mogelijk om het gemiddelde te nemen van de omliggende pixels. Het is ook mogelijk om het te bepalen door middel van een gaussian filter. In dat geval zullen de pixels dicht bij de oorsprong zwaarder wegen dan de pixels die verder weg zitten. Adaptive thresholding kan vooral handig zijn als het uitgevoerd wordt op een afbeelding met flinke intensiteitsverschillen.

Edge tracking by hysteresis

De meeste thresholding methoden kijken alleen naar de pixels en controleren of ze sterk genoeg zijn om te mogen blijven bestaan. Hoewel dit, als het op de juiste manier gedaan wordt best goede resultaten kan geven, mist het vaak toch nog onderdelen van randen. Het zou mooier zijn als een gedetecteerde rand gevolgd zou worden. Dit is precies wat Edge tracking by hysteresis doet. Door eerst double thresholding toe te passen worden de duidelijke randen goed gedetecteerd en de zwakkere randen gemarkeerd. Edge tracking by hysteresis kan daarna naar de zwakke randen kijken en controleren of deze rand vast zit aan een sterke rand. Als dat het geval is dan wordt die zwakke rand ook als sterke rand gemarkeerd. Dit heeft ook het mooie effect dat ruis verwijderd zal worden. Als er namelijk een losse pixel gevonden wordt dan ziet edge tracking by hysteresis dat deze niet verbonden zit met een sterke rand en wordt de pixel verwijderd.

1.4. Keuze

Thresholding is de laatste stap van edge detectie. Aangezien de gebruikte edge detectiemethode Deriche is, moet deze thresholding geoptimaliseerd zijn op de uitvoer van Deriche. Adaptive thresholding valt hierdoor hoe dan ook af aangezien Deriche een uitvoer geeft dat niet enorm in intensiteit verschilt.

Met het doel in gedachten, echte randen naar voren brengen en zwakke randen en ruis verwijderen met nauwkeurigheid als prioriteit, is de volgende conclusie getrokken: Om aan alle eisen van het doel te voldoen is het mogelijk om verschillende methoden samen te voegen. Om op elke afbeelding een goed resultaat te krijgen is het van belang dat de thresholding waarde niet elke keer handmatig aangepast hoeft te worden. Hiervoor wordt er gebruik gemaakt van Dynamic thresholding. Dan, om de echte randen naar voren te brengen zonder delen van die randen die minder goed te zien zijn te verwijderen, wordt er gebruik gemaakt van Double thresholding. Om dan de echte zwakke randen te onderscheiden van de losse zwakke randen en eventuele ruis te verwijderen wordt er gebruik gemaakt van Edge tracking by hysteresis.

Deze drie methoden gecombineerd Dynamic thresholding, Double thresholding en Edge tracking by hysteresis, geven de beste resultaten in samenwerking met het Deriche edge detection methode.

1.5. Implementatie

De thresholding implementatie bestaat in feite uit drie stappen.

1. Bepalen van thresholding waarden.
2. Markeren van sterke en zwakke randen.
3. Controleren welke zwakke randen echte randen zijn.

Voor stap één (bepalen van thresholding waarden) moet er één keer door de hele afbeelding heen worden gelopen. Hierbij wordt er bijgehouden wat de hoogste pixel waarde is. Deze pixel waarde wordt dan vermenigvuldigt met een constante waarde. Deze waarde wordt de waarde wat bepaalt of een rand sterk is. Er zal ook een constante nodig zijn voor de waarde dat bepaald of een rand zwak is. Met deze twee constante is de thresholding ook makkelijk te tweaken.

Voor stap twee (markeren van sterke en zwakke randen) moet er een tweede keer door de afbeelding heen worden gelopen. Deze keer wordt elke pixel waarde vergeleken met de voorheen vastgestelde thresholding waarde. Als de pixel waarde hoger is dan de high thresholding waarde dan wordt de pixel maximaal wit. Als dit niet zo is, dan zal de pixel vergeleken worden met de low thresholding waarde. Is de pixel waarde dan wel hoger dan wordt het grijs. Als de pixel ook niet hoger is dan de low thresholding waarde dan wordt de pixel zwart.

Voor stap drie (Controleren welke zwakke randen echte randen zijn) wordt er voor de laatste keer door de hele afbeelding heen gelopen. Deze keer wordt er voor elke pixel naar de acht burens gekeken. Als één van deze burens een fel witte pixel is dan betekent dat deze pixel tegen een sterke rand aan ligt. Dit houdt zeer waarschijnlijk in dat de pixel onderdeel uitmaakt van deze rand en dus wordt ook deze pixel maximaal wit gemaakt. Zijn er geen fel witte burens dan is dit een losliggende pixel en wordt deze op zwart gezet.

1.6. Evaluatie

Er zullen verschillende tests gedaan worden met verschillende afbeeldingen en instellingen. Aangezien nauwkeurigheid belangrijk is zullen de resultaten vergeleken worden met de huidige implementatie.

1.7 Bronvermelding

<https://dsp.stackexchange.com/questions/2411/what-are-the-most-common-algorithms-for-adaptive-thresholding>
<http://hanzratech.in/2015/01/21/adaptive-thresholding.html>
<https://www.wavemetrics.com/products/igorpro/imageprocessing/thresholding.htm>
https://en.wikipedia.org/wiki/Canny_edge_detector
<http://justin-liang.com/tutorials/canny/#double-thresholding>
<https://de.wikipedia.org/wiki/Schwellenwertverfahren>
[https://en.wikipedia.org/wiki/Thresholding_\(image_processing\)](https://en.wikipedia.org/wiki/Thresholding_(image_processing))