

1. Week 3 26-04-2015 (Edge detection en thresholding)

1.1. Week3 26-05-2015

Gemaakt door Thijs van Tiem & Jos Roijackers

1.2. Doel

Het doel van dit onderdeel is te kijken naar de verschillende algoritmes en methodes die er gebruikt kunnen worden voor edge-detection. Hiervoor hebben wij een document gemaakt namelijk: "onderzoek edge-detection.docx". Uiteindelijk hebben we de methode uit het onderzoek niet gebruikt maar de laplacian of gaussian methode.

Het totale doel van deze opdracht is dus om meer te weten te komen over edge-detection en thresholding

1.3. Methoden

We hebben gekeken naar de volgende methodes en uiteindelijk een keuze gemaakt:

1. Canny edge detection
(maakt gebruik van rand detectoren op basis van een gaussian)
2. Edge thinning
(haalt alles weg wat niet nodig is)
3. Fuzzy logic method
(stap voor stap alles ontleden en omzetten aan de hand van fuzzy logic)
4. Sobel method
(krachtige manier van objecten herkennen als de achtergrond straks en normaal is)

Eerste keuze:

Wij gaan voor de Canny methodes omdat dit nog steeds een van de oudste en meeste gebruikte methodes is. Op deze manier kunnen wij hier veel van vinden en tevens een van de meest gebruikte methodes uitvoeren.

Leraar raad sobel methode aan of gaussian uit de sheets

Uiteindelijk gemaakte keuze:

Uiteindelijk zijn we niet uit de Canny method gekomen maar hebben wij gekozen voor de Laplacian of Gaussian methode op aanraden van Arno zelf. Aan de hand van deze methode is het ons wel gelukt en zijn wij zeer tevreden met de resultaten. Wel is het jammer dat de gebruikte methode met Otsu thresholding zo snel een edge detecteerde dat de afbeeldingen teveel edges aangeeft wat de methode ongeschikt maakte voor edge detection in een vision programma die wordt toegepast op mensen.

Bronnen:

http://en.wikipedia.org/wiki/Edge_detection
<http://nl.wikipedia.org/wiki/Canny-randzoeker>
<http://nl.mathworks.com/discovery/edge-detection.html>
<http://nl.mathworks.com/help/fuzzy/examples/fuzzy-logic-image-processing.html>
http://docs.opencv.org/doc/tutorials/imgproc/imgtrans/canny_detector/canny_detector.html

mooi voorbeeld:

http://boofcv.org/index.php?title=Example_Canny_Edge
<http://www.imagemagick.org/Usage/transform/#edge>

1.4. Keuze

We hebben gekozen zoals hierboven al toegelicht is voor de Laplacian of gaussian methode met Otsu thresholding en hebben hiervoor gekozen omdat de eerste methode canny te moeilijk was. Arno heeft ons toen deze methode aangeraden en ons aan de hand van een bord tekening uitgelegd hoe alles werkt.

1.5. Implementatie

Uiteindelijk zorg je ervoor dat er een kernel van 5x5 gemaakt word en een lapkernel van 5x5 en dat aan de hand van van de pixels * de kernel zal er een formule op worden toegepast die daarna doorgaat met de waardes van de lapkernel en daarvan de intensity van de image mee zet. Naderhand return je dan de waardes uit de lap in een intensity image en zorg je dat dit in zijn geheel geretourneerd word.

Naderhand word er nog thresholding toegepast op de image die dus werkt met de Otsu methode. De Otsu methode heeft een formule die het gemiddelde bepaald van de afbeelding en de waardes die hierboven zitten op zwart zet en de waardes die hieronder zitten op wit zet.

1.6. Evaluatie

Wij hebben de afbeeldingen van de leraar vergeleken met die van ons op twee verschillende onderdelen. We hebben gekeken of alles zowel bij de code van leraar als de code van ons correct werd uitgevoerd. En we hebben gekeken of de afbeeldingen overeenkwamen met elkaar en wat de mogelijke verschillen hier in waren.

Er valt te concluderen dat onze thresholding niet zo goed werkte als die van de leraar en dat het vision algoritmen hierdoor niet perfect meer uitgevoerd kan worden. Hij detecteert door de uiteindelijke Otsu waardes te snel de edges.