

Implementatieplan Week 3 - Edge Detection

Namen en datum

Coen Andriessen en Mathijs Arends 13-7-2015 Versie 1.2.

Inleiding

Edge detection is de naam voor een set van wiskundige methode die gericht zijn op het identificeren van de punten in een digitaal beeld waarin de helderheid van het beeld sterk verandert. De punten waarop de helderheid van het beeld verandert zijn meestal georganiseerd in een reeks van geboden lijnsegmenten ook wel randen genoemd. Edge detection is een fundamenteel bewerkingsmethode in beeldverwerking.

Doel

Het detecteren van randen in een afbeelding.

Methoden

Om randen te ontdekken in en afbeelding wordt er vaak gekeken waar de afbeelding een hoge intensiteit verandering heeft.

Methodes:

1. Canny Edge Methode.
2. Sobel Methode.
3. Prewitt Methode.
4. Frei-Chen Methode.
5. Laplacian Methode.

Canny Edge Methode

Stap 1

Uitfilteren van ruis. Om dit te realiseren wordt er gebruik gemaakt van een Guassian filter. Hieronder een voorbeeld van een Guassian kernel met de size = 5.

$$K = \frac{1}{159} \begin{bmatrix} 2 & 4 & 5 & 4 & 2 \\ 4 & 9 & 12 & 9 & 4 \\ 5 & 12 & 15 & 12 & 5 \\ 4 & 9 & 12 & 9 & 4 \\ 2 & 4 & 5 & 4 & 2 \end{bmatrix}$$

Stap 2

Vind het intensiteitverloop van de afbeelding. Dit kan gerealiseerd worden met Sobel.

1. Breng een paar convolution masks aan op de afbeelding.
2. Zoek naar de gradiënt sterkte en de richting. Deze wordt afgerond naar de volgende vier mogelijke richtingen (0, 45, 90 of 135).

Stap 3

Pas Non-maximum suppression toe op de afbeelding. Dit verwijdert pixels die niet worden beschouwd als deel van een rand te zijn. Door deze bewerking zullen alleen dunne lijnen zichtbaar blijven.

Stap 4

Haal ruis weg door middel van thresholds (upper and lower). Als de pixel gradiënt hoger is dan de upper threshold, wordt de pixel gezien als een edge. Als de pixel lager is dan de lower threshold wordt het genegeerd. Als de pixel gradiënt tussen de twee threshold inzit, wordt het alleen geaccepteerd als de gradiënt van de aanliggende pixel hoger is dan de upper threshold.

Sobel Methode

Stap 1

Lees de afbeelding in.

Stap 2

Het object dat moet worden gesegmenteerd verschilt sterk in contrast met de achtergrond. Veranderingen in contrast kan worden gedetecteerd door operators die het verloop van een beeld berekenen. De gradiënt kan worden berekend en een drempelwaarde kan worden toegepast op een binary mask. Als eerste gebruiken we de Sobel operator om de drempelwaarde te berekenen. Vervolgens creëren we een binary mask van de afbeelding.

Stap 3

De binary mask toont lijnen van hoog contrast in het beeld. Deze lijnen zullen niet volledig op elkaar aansluiten. Ten opzichte van het oorspronkelijke beeld kunnen lijnen rond het object in het gradiënt mask te zien zijn. Deze onderbrekingen kunnen verdwijnen met behulp van lineaire structurerende elementen.

Stap 4

Vul de onderbrekingen in de omtrek van het object op.

Stap 5

Verwijder aanliggende objecten

Stap 6

Maak het object "glad", hierdoor zullen scherpe lijnen afgevlakt worden.

Prewitt Methode

Bij de Prewitt Methode wordt er gebruik gemaakt van 2 kernels, een horizontale en verticale. Aan de hand van deze twee kernels worden afgeleiden geschat van de omliggende pixels op een bepaalde lijn. Door deze afgeleiden samen te voegen kan de gradiënt sterkte bepaald worden evenals de richting.

Frei-Chen Methode

Deze methode maakt gebruik van een 3x3 texel footprint maar geldt als een totaal van negen convolution masks. Frei-Chen maskers zijn unieke maskers die alle basisvectoren bevatten. Het 3x3 taxel footprint bestaat uit negen convolution masks.

$$\begin{aligned} G_1 &= \frac{1}{2\sqrt{2}} \begin{bmatrix} 1 & \sqrt{2} & 1 \\ 0 & 0 & 0 \\ -1 & -\sqrt{2} & -1 \end{bmatrix} & G_2 &= \frac{1}{2\sqrt{2}} \begin{bmatrix} 1 & 0 & -1 \\ \sqrt{2} & 0 & -\sqrt{2} \\ 1 & 0 & -1 \end{bmatrix} & G_3 &= \frac{1}{2\sqrt{2}} \begin{bmatrix} 0 & -1 & \sqrt{2} \\ 1 & 0 & -1 \\ -\sqrt{2} & 1 & 0 \end{bmatrix} \\ G_4 &= \frac{1}{2\sqrt{2}} \begin{bmatrix} \sqrt{2} & -1 & 0 \\ -1 & 0 & 1 \\ 0 & 1 & -\sqrt{2} \end{bmatrix} & G_5 &= \frac{1}{2} \begin{bmatrix} 0 & 1 & 0 \\ -1 & 0 & -1 \\ 0 & 1 & 0 \end{bmatrix} & G_6 &= \frac{1}{2} \begin{bmatrix} -1 & 0 & 1 \\ 0 & 0 & 0 \\ 1 & 0 & -1 \end{bmatrix} \\ G_7 &= \frac{1}{6} \begin{bmatrix} 1 & -2 & 1 \\ -2 & 4 & -2 \\ 1 & -2 & 1 \end{bmatrix} & G_8 &= \frac{1}{6} \begin{bmatrix} -2 & 1 & -2 \\ 1 & 4 & 1 \\ -2 & 1 & -2 \end{bmatrix} & G_9 &= \frac{1}{3} \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} \end{aligned}$$

De eerste vier Frei-Chen maskers worden gebruikt om de randen te detecteren. De vier volgende worden gebruikt voor lijnen en door middel van de laatste wordt het gemiddelde berekend.

Laplacian Methode.

Bij deze methode wordt een Laplacian kernel gebruikt om edge detection te realiseren. Er kan gebruik gemaakt worden van een 9x9 kernel of een 3x3 kernel.

$$\text{kernel} = \begin{array}{cccccccc} & & & 1 & 1 & 1 & & & \\ & 0 & & 1 & 1 & 1 & & 0 & \\ & & & 1 & 1 & 1 & & & \\ 1 & 1 & 1 & -4 & -4 & -4 & 1 & 1 & 1 \\ 1 & 1 & 1 & -4 & -4 & -4 & 1 & 1 & 1 \\ 1 & 1 & 1 & -4 & -4 & -4 & 1 & 1 & 1 \\ & & & 1 & 1 & 1 & & & \\ & 0 & & 1 & 1 & 1 & & 0 & \\ & & & 1 & 1 & 1 & & & \end{array}$$

Door op elke pixel van de afbeelding het kernel te leggen, zullen edge zichtbaar zijn. Aan het Laplacian filter kan eventueel een Gaussian filter toegevoegd worden. Dit zal resulteren tot minder ruis in de eind afbeelding.

Keuze

Wij hebben ervoor gekozen om een Laplacian filter te gebruiken. De keuze is op deze methode gevallen omdat het principe van een Laplacian filter eenvoudig is en het eindresultaat zal geschikt moeten zijn.

Implementatie

In de klasse StudentPreProcessing is een const int aangemaakt waarin de waardes van het laplacian kernel zijn opgeslagen. In de klasse StudentPreProcessing is een functie stepEdgeDetection die gebruikt zal worden voor het detecteren van edges in een afbeelding. Deze functie heeft als parameter een afbeelding waarin edges gedetecteerd dienen te worden. In de functie zal de afbeelding geconverteerd worden naar een output afbeelding.

Evaluatie

De implementatie edge detection zal getest worden op de volgende punten:

- Snelheid
- Robuustheid
- Resultaat

De testresultaten zullen verwerkt worden in een meetrappport.