Implementatie Plan Edge Detection

Menno van der Jagt

Faizal Supriadi

14-02-2020

Inleiding

1.2. Doel

Het doel van de implementatie is, dat de edge detection sneller en efficiënter wordt.
Met de methode die nu wordt gebruikt als edge detector in de ExternalDLL, vinden we dat het veel sneller en efficiënter kan worden. Wij zullen dan ook nieuwe methodes onderzoeken en daarmee een betere methode proberen te vinden om die vervolgens toe te passen. We richten ons hierbij voornamelijk op snelheid en minder op juistheid van de edge detection, omdat we real-time belangrijker vinden dan hoe goed het is.

1.3. Methoden

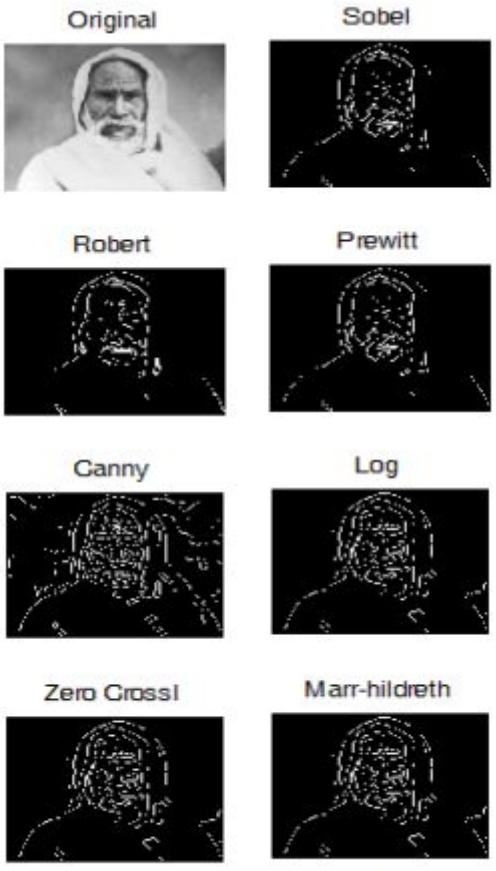

Figure 1. Various Edge Detection Techniques

1.3.1 Sobel Methode

Dit is een veel gebruikte edge detector. Het maakt gebruik van een 3 bij 3 matrix een voor elke x en y richting (figuur 2). Het verloop van de x-richting heeft min getallen aan de linkerkant en positieve getallen aan de rechterkant. Op dezelfde manier heeft de y-richting min getallen aan de onderkant en positieve getallen aan de bovenkant. Om te kijken waar een edge is voor bijvoorbeeld de x-richting kijken we naar de werkelijke pixel waardes en doen dit keer de waardes op de matrix (alleen linker en rechter kant, niet het midden). Vervolgens doen we die getallen plus elkaar en daaruit krijgen we een bepaalde waarde. Hoe kleiner deze waarde is hoe kleiner het eindantwoord is, en hoe groter hoe meer verschil er is tussen de linker kant en de rechter kant. Dit geldt hetzelfde voor de y-richting. Voor een volledige en duidelijke uitleg zie (https://www.youtube.com/watch?v=uihBwtPIBxM).

$$M_x = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} \qquad M_y = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix}$$

Figuur 2

Voordelen:

- Het is relatief goedkoop in termen van berekeningen.
- snel
- werkt goed met ruis

Nadelen:

- De gradiënt benadering is relatief grof, voornamelijk bij veel hoge variaties in het beeld.
- werkt slecht met kleur.
- onnauwkeurig
- werkt slecht met zwakke hoeken

1.3.2 Prewitt Methode

De Prewitt Edge methode is heel erg vergelijkbaar met de Sobel Edge methode. Het voornaamste verschil zit hem in de waardes van het 3 bij 3 matrix, Prewitt heeft -1, 0 en 1 (figuur 3) terwijl sobel -1, -2, 0, 1 en 2 (figuur 1) heeft. Hierdoor krijg je net iets andere resultaten dan bij de Sobel methode. Zoals te zien is in het onderzoek “Comparative Study on Various Edge Detection Techniques for 2-D Image” Geeft de Prewitt Edge Detection methode de hoeken van een object met edges beter weer in vergelijking tot de Sobel Edge Detection methode die in de hoeken van een vierkant object geen rechte lijnen geeft.

$$M_x = \begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix} \qquad M_y = \begin{bmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{bmatrix}$$

Figuur 3

Voordelen:

- snel
- Het is relatief goedkoop in termen van berekeningen
- werkt redelijk goed met ruis

Nadelen:

- gevoelig voor veel variatie
- onnauwkeurig
- werkt slecht met kleuren

1.3.3 Canny Methode

Dit is een van de meest wijd gebruikte edge detectors. Het werkt door eerst de afbeelding te smoothen door middel van Gaussian convolution. Dan wordt een simpele 2D derivative operator gebruikt om op de gemodificeerde afbeelding, gebieden te highlighten die een groot spatial derivatives hebben. De edges laten dan ridges zien. Het algoritme gebruikt dan non maximal suppression..

Voordelen:

- De Canny methode geeft single-pixel-wide afbeeldingen met een goede voortzetting tussen aangrenzende pixels.
- Het is erg gevoelig voor zijn parameters, die moeten worden aangepast voor verschillende toepassingsdomeinen. Hiermee kan de berekeningstijd veranderen.

Nadelen

- De Gaussian filter zal ook de edges smoothen, waardoor zwakke edges kunnen worden gemist.
- Ruisgevoelig.
- Hoe complexer de afbeelding, hoe complexere calculaties voor de thresholds nodig zijn.

1.3.4 Roberts Methode

Er zijn twee methode van Roberts Edge Detector. De tweede methode is meestal gebruikt vanwege zijn efficiëntie.

1	$\sqrt{[I(r,c) - I(r-1,c-1)]^2 + [I(r,c-1) - I(r-1,c)]^2}$
2	$\lvert I(r,c) - I(r-1,c-1)\rvert + \lvert I(r,c-1) - I(r-1,c)\rvert$

Het gebruikt Gaussian smoothing

Voordelen:

- Simpele benadering bij het eerste derivaat
- Het simpelst van de edge operators en werkt met binaire afbeeldingen

Nadelen:

- Slecht met ruis

$$M_x = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} \qquad M_y = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix}$$

1.3.5 Laplacian Methode

In vergelijking tot de sobel edge detector, de laplacian edge detector maakt gebruik van een enkele kernel. Laplacian heeft twee verschillende soorten kernels de negatieve en de positieve.

$$\begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} 1 & 1 & 1 \\ 1 & -8 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

Voordelen:

- vind goede hoeken

Nadelen:

- werkt heel slecht met ruis (peper and salt)

1.3.6 FuzzySet Methode

De fuzzy set methode gebruikt wiskundige en logische redeneringen op basis van benaderingen in plaats van scherpe waarden, daarom heeft deze methode een significant verminderde complexiteit van problemen waar vaste waarden niet kunnen worden bereikt of voorspeld. De fuzzy set methode converteerd de gekleurde afbeelding naar een gesegmenteerde afbeelding en dan wordt een edge detector geconvolueerd daarover om een edged afbeelding te krijgen.

$$\begin{vmatrix} 0 & b & a \\ 0 & b & a \\ 0 & b & a \end{vmatrix} \begin{vmatrix} a & a & a \\ 0 & 0 & 0 \\ b & b & b \end{vmatrix} \begin{vmatrix} a & a & b \\ a & b & 0 \\ b & 0 & 0 \end{vmatrix}$$

$$\begin{vmatrix} a & a & a \\ b & b & b \\ 0 & 0 & 0 \end{vmatrix} \begin{vmatrix} a & b & 0 \\ a & b & 0 \\ a & b & 0 \end{vmatrix} \begin{vmatrix} 0 & 0 & 0 \\ a & a & a \\ b & b & b \end{vmatrix}$$

$$\begin{vmatrix} b & b & b \\ 0 & 0 & 0 \\ a & a & a \end{vmatrix} \begin{vmatrix} b & a & a \\ 0 & b & a \\ 0 & 0 & b \end{vmatrix} \begin{vmatrix} b & a & 0 \\ b & a & 0 \\ b & a & 0 \end{vmatrix} \begin{vmatrix} a & 0 & b \\ a & 0 & b \\ a & 0 & b \end{vmatrix} \begin{vmatrix} 0 & 0 & 0 \\ b & b & b \\ a & a & a \end{vmatrix}$$

$$\begin{vmatrix} 0 & a & b \\ 0 & a & b \\ 0 & a & b \end{vmatrix} \begin{vmatrix} b & b & b \\ a & a & a \\ 0 & 0 & 0 \end{vmatrix} \begin{vmatrix} b & 0 & a \\ b & 0 & a \\ b & 0 & a \end{vmatrix} \begin{vmatrix} b & 0 & 0 \\ a & b & 0 \\ a & a & b \end{vmatrix} \begin{vmatrix} 0 & 0 & b \\ 0 & b & a \\ b & a & a \end{vmatrix}$$

16 templates used in detection method

Voordelen:

- Minder complexiteit
- Blurred de afbeelding niet

Nadelen:
- Redelijk sloom
- Werkt slecht met ruis
- Double edges worden minder geïdentificeerd

1.4. Keuze

In de applicatie wordt standaard gebruik gemaakt van de laplacian operator met als kernel:

$$kernel = \begin{matrix} & & & 1 & 1 & 1 & & & \\ & 0 & & 1 & 1 & 1 & & 0 & \\ & & & 1 & 1 & 1 & & & \\ 1 & 1 & 1 & -4 & -4 & -4 & 1 & 1 & 1 \\ 1 & 1 & 1 & -4 & -4 & -4 & 1 & 1 & 1 \\ 1 & 1 & 1 & -4 & -4 & -4 & 1 & 1 & 1 \\ & & & 1 & 1 & 1 & & & \\ & 0 & & 1 & 1 & 1 & & 0 & \\ & & & 1 & 1 & 1 & & & \end{matrix}$$

Er wordt standaard gebruik gemaakt van de Laplacian methode, wij hebben gekozen voor de Prewitt en de Sobel methode om die te implementeren. We hebben hiervoor gekozen, omdat de Prewitt methode en de Sobel methode heel erg veel op elkaar lijken.

Verder vinden wij het belangrijk dat de methode erg snel moet zijn en dat het niet te veel resources gebruikt. Daarnaast willen wij dat het ook goed werkt met ruis (peper and salt). De Laplacian methode is hier erg slecht in en geeft dus vaak te veel edges weer die er eigenlijk niet zijn, terwijl de Prewitt en de Sobel methode hier beter mee werken en iets minder edges geven.

We hebben niet gekozen voor de Fuzzy Set, Canny en Roberts -methode, omdat ze slomer zijn, meer resources gebruiken en ruis gevoeliger zijn en daarom te veel edges weergeven.

1.5. Implementatie

We beginnen met het begrijpen van de code die al geïmplementeerd is, vervolgens zullen we in de studentenversie de prewitt en sobel methode toe voegen. En de threshold te veranderen naar Otsu thresholding method.

1.6. Evaluatie

Deze experimenten zullen gedaan worden om te zien of de nieuwe methode beter is:
Tijd meten tijdens het runnen van de methodes.

- We zullen hiervoor gebruik maken van een c++11 standaard library genaamd chrono, om de tijd te berekenen van de edge detectie methodes. Dit doen we door aan het begin en aan het einde van de functie de cpu clock op te vragen. Verder zullen we dit op verschillende hardware meerdere keren uittesten om te kijken hoeveel invloed de hardware er op heeft. Hiervoor zullen wij gebruik maken van 3 computers/laptops.

Geheugen meten tijdens het runnen van de methodes.

- Wij zullen hiervoor de diagnostic tools gebruiken van Visual Studio 2019. Met de diagnostic tools kunnen wij de process memory meten van het programma. Dit doen we op 1 computer, omdat het een tijd kostende experiment is.

Voor het vergelijken van de snelheid nemen we de zeven testfoto's en zetten we in een tabel neer wat de gemiddelde snelheid per foto is per methode en hardware.

Voor het vergelijken van het geheugen nemen we zeven testfoto's en zetten we ze in een tabel wat de process memory weergeeft in megabytes. Dit doen we dan 10 keer per foto, daarna berekenen we het gemiddelde van de resultaten en vergelijken de methoden met elkaar.

1.7. Bronnen

Kamlesh Kumar, Saeed Ahmed Khan, & Jian- Ping Li. (2015, 0 augustus). *Comparative Study on Various Edge Detection Techniques for 2-D Image*. Geraadpleegd van https://www.researchgate.net/publication/279062441_Comparative_Study_on_Various_Edge_Detection_Techniques_for_2-D_Image

Othman O. Khalifa, Abdulrahman Moffaq Alawad, Farah Diyana Abdul Rahman, & Norun Abdul Malek. (2018, 3 augustus). *Fuzzy Logic based Edge Detection Method for Image Processing*. Geraadpleegd van https://www.researchgate.net/publication/325428685_Fuzzy_Logic_based_Edge_Detection_Method_for_Image_Processing

Haq, I. (2015, 25 september). *Fuzzy Logic Based Edge Detection in Smooth and Noisy Clinical Images*. Geraadpleegd van https://journals.plos.org/plosone/article?id=10.1371/journal.pone.0138712

Kamlesh Kumar, Saeed Ahmed Khan, & Jian- Ping Li. (2015, augustus). *Comparative Study on Various Edge Detection Techniques for 2-D Image*. Geraadpleegd van https://www.researchgate.net/publication/279062441_Comparative_Study_on_Various_Edge_Detection_Techniques_for_2-D_Image

Kumar, K. (2015. *Comparative Study on Various Edge Detection Techniques for 2-D Image*. Geraadpleegd van https://www.semanticscholar.org/paper/Comparative-Study-on-Various-Edge-Detection-for-2-D-Kumar-Li/e9f9e8674c467252a58e7a55d8a64236ca06bf23

Deepika Adlakha. (z.d.). *Analytical Comparison between Sobel and Prewitt Edge Detection Techniques* . Geraadpleegd op 13 maart 2020, van https://www.ijser.org/researchpaper/Analytical-Comparison-between-Sobel-and-Prewitt-Edge-Detection-Techniques.pdf

Laplacian Operator - Tutorialspoint. (z.d.). Geraadpleegd op 13 maart 2020, van https://www.tutorialspoint.com/dip/laplacian_operator.htm

Othman O. Khalifa, Abdulrahman Moffaq Alawad, Farah Diyana Abdul Rahman, & Norun Abdul Malek. (2018, 3 augustus). *Fuzzy Logic based Edge Detection Method for Image Processing*. Geraadpleegd van https://www.researchgate.net/publication/325428685_Fuzzy_Logic_based_Edge_Detection_Method_for_Image_Processing

Mamta Joshi, & Ashutosh Vyas. (z.d.-a). *Comparison of Canny edge detector with Sobel and Prewitt edge detector using different image formats*. Geraadpleegd op 13 maart 2020, van https://www.ijert.org/research/comparison-of-canny-edge-detector-with-sobel-and-prewitt-edge-detector-using-different-image-formats-IJERTCONV2IS03009.pdf

Pushpajit A. Khaire, & Dr. Nileshsingh V. Thakur. (z.d.-b). *A Fuzzy Set Approach for Edge Detection*. Geraadpleegd op 18 maart 2020, van https://www.cscjournals.org/manuscript/Journals/IJIP/Volume6/Issue6/IJIP-695.pdf

Saad J Bedros. (z.d.-b). *Edge detection*. Geraadpleegd op 13 maart 2020, van http://dept.me.umn.edu/courses/me5286/vision/VisionNotes/2017/ME5286-Lecture7-2017-EdgeDetection2.pdf

Canny Edge Detector. (z.d.). Geraadpleegd op 15 maart 2020, van http://homepages.inf.ed.ac.uk/rbf/HIPR2/canny.htm

Computerphile. (2015, 11 november). *Canny Edge Detector - Computerphile*. Geraadpleegd van https://www.youtube.com/watch?v=sRFM5IEqR2w