

Implementatie Plan

Edge Detection



Naam: Peter Schenkels en Bas van der Geer

Klas: V2A-TI

Docent: Diederik Roijers

Student Assistent: Roxanne van der Pol

Datum: 14-02-2020

Inhoudsopgave

1. Doelen	3
2. Methoden	3
3. Keuze	6
4. Implementatie	6
5. Evaluatie	6
Bronnen	6

1. Doelen

Het doel van Edge detection is om een afbeelding te verminderen in details en de omlijning duidelijker aan te geven. In dit geval wordt edge detection gebruikt als preprocessing stap voor gezichtsherkenning. Het doel van thresholding is om de afbeelding op te splitsen in een binaire waarde, de foreground en de background. Ons doel is om de lijnen duidelijker te maken en dunner te maken.

2. Methoden

In dit hoofdstuk beschrijven wij de methoden voor edge detection. In de volgende tabel zijn deze methode te zien

Methoden	Voordelen	Nadelen
Sobel	Lage time complexity, Lage space complexity	gevoelig voor ruis
Prewitt	Lage time complexity, Lage space complexity	gevoelig voor ruis
Laplacian	Lage time complexity, Lage space complexity	Heel gevoelig voor noise
Canny	Nauwkeurig, flexibel	Sloom
Rober	Gemiddeld het snelst, Lage space complexity	Erg Onnauwkeurig, gevoelig voor ruis

Sobel methode:

Sobel edge detection methode gebruikt derivative approximation om de edges te detecteren. Daarmee wordt bedoeld dat het algoritme een edge returned wanneer de gradiënt op een gegeven punt op zijn hoogst is. Dit doet hij met een verticale en horizontale kernel.

X – Direction Kernel

-1	0	1
-2	0	2
-1	0	1

Y – Direction Kernel

-1	-2	-1
0	0	0
1	2	1

Uiteindelijk wordt de gradiënt intensity zo uitgerekend:

$$|G| = \sqrt{G_x^2 + G_y^2}$$

Gx en Gy Stellen de horizontale en verticale kernels voor.

Robert cross edge detection methode:

De robert cross edge detection gebruikt een simpele en snel uitreken baren kernel om edges te berekenen. De kernels worden ook toegepast op de gradiënt intensity alleen zijn de kernels een stuk kleiner waardoor hij sneller een edge kan berekend, maar dan is het nadeel is dat het algoritme dan wel onnauwkeurig is.

1	0
0	-1

0	+1
-1	0

Prewitt methode:

Prewitt is kort gezegd bijna gelijk aan de sobel methode. Het enige grote verschil is dat de kernel andere waardes heeft.

-1	0	+1
-1	0	+1
-1	0	+1

G_x

+1	+1	+1
0	0	0
-1	-1	-1

G_y

Prewitt werkt het best met foto's die weinig of geen noise bevatten en goed gecontrasteerd zijn.

Canny methode:

De canny methode werkt op de volgende manier. De methode maakt gebruik van verschillende algoritmes. Deze methode wordt het meest gebruikt en is ook het meest effectief maar wel een stuk complexer. Hieronder zijn de stappen beschreven van deze methode van edge detection.

1. Smoothing image and reducing noise with the Gaussian filter.
2. Calculate the gradient using Sobel or Prewitt.
3. Extract the edge points with non maximum suppression.
4. Thresholding
5. Hysteresis

Laplacian methode:

Laplacian gebruikt maar een kernel voor edge detection. Laplacian berekent een tweede orde afgeleide in een keer. Twee veel gebruikte kernels zijn de volgende.

0	-1	0
-1	4	-1
0	-1	0

-1	-1	-1
-1	8	-1
-1	-1	-1

<https://medium.com/@nikatsanka/comparing-edge-detection-methods-638a2919476e>

Laplacian is heel erg gevoelig voor noise dus wordt er vaak eerst een Gaussian filter toegepast voor dat de Laplacian filter wordt toegepast.

3. Keuze

We willen graag dat ons algoritme snelle resultaten levert, omdat het doeleinde een bewakingscamera wordt bijvoorbeeld. Daarvoor vallen algoritmes die gebaseerd zijn op precisie af. Het leek ons het beste om sobel te kiezen als algoritme. Sobel is in principe een eenvoudig en bekend algoritme dus dat zal ons implementatie proces makkelijker maken.

4. Implementatie

We zullen eerst een "Student" implementatie moeten schrijven voor een intensity image, omdat we een intensity image binnen krijgen. Daarna gaan we op deze image een horizontale en verticale sobel kernel toepassen. De uitkomsten van deze twee kernels bereken we dan de stelling van pythagoras mee uit. Hieruit komt een pixelwaarde, dit doen we dan voor elke pixels. Die voegen we allemaal samen en daaruit komt een rauwe sobel image. We moeten dan nog wat thresholding dan toepassen zodat de ruis verdwijnt.

5. Evaluatie

Wij gaan gebruik maken van 5 test sets met elk 10 foto's. We gaan met Visual Studio Code timen met de tool die daar is ingebouwd. Als dit niet werkt gaan we gebruiken maken van code om de snelheid te bepalen. Om het verschil te meten tussen het originele algoritme en ons algoritme gaan we de snelheid vergelijken. Ook gaan we kijken naar welke algoritme beter gezichten herkent in afbeeldingen. Al deze resultaten verwerken wij in ons meetrapport.

Bronnen

Artikel Bronnen:

Tsankashvili, N. (2019, 15 juni). *Comparing Edge Detection Methods*. Geraadpleegd van <https://medium.com/@nikatsanka/comparing-edge-detection-methods-638a2919476e>

Gupta, S., Mazumdar, S. G., & Student, M. T. (2013, January 1). Sobel Edge Detection Algorithm. Geraadpleegd van <https://pdfs.semanticscholar.org/6bca/fdf33445585966ee6fb3371dd1ce15241a62.pdf>

Salem Saleh Al-amri et. al. (2010, November 3) image segmentation using edge detection Geraadpleegd van <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.302.9543&rep=rep1&type=pdf>

Afbeelding bronnen:

https://www.projectrhea.org/rhea/index.php/An_Implementation_of_Sobel_Edge_Detection
https://www.researchgate.net/figure/Masks-for-the-Prewitt-gradient-edge-detector-The-Laplacian-operator-is-based-on-second_fig3_317754223
<https://medium.com/@nikatsanka/comparing-edge-detection-methods-638a2919476e>