Meetraport Practica Vision

Runtime performance

Edge Detection

10 april 2018

Door:

- Kiet van Osnabrugge
- Wiebe van Breukelen

Inhoud

1.1. Doel

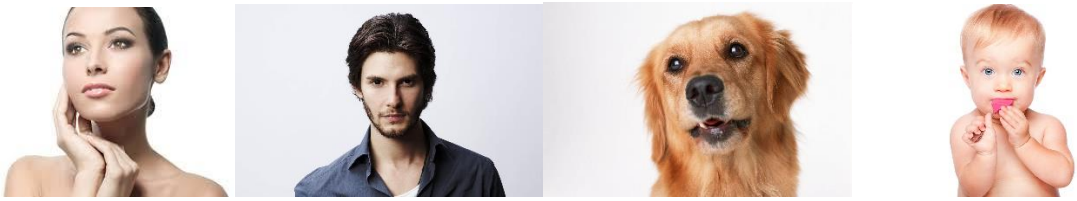
Onderzocht gaat worden of het Prewitt en Sobel algoritme een verschil in runtime performance hebben. Voor toepassingen zoals realtime edge detectie is dit van belang. Om dit grondig te testen zal deze proef gedaan worden voor verschillende resoluties.

1.2. Hypothese

Wij verwachten dat Sobel een wat langere runtime heeft omdat Sobel een iets zwaardere bereiking heeft. Bij Prewitt kan de linker van de rechterkant van de kernel worden afgetrokken en bij Sobel zal er vermenigvuldigt moeten worden om de meetfactor van twee te bereiken. Dit zal volgens ons een negatieve invloed hebben op de runtime van Sobel.

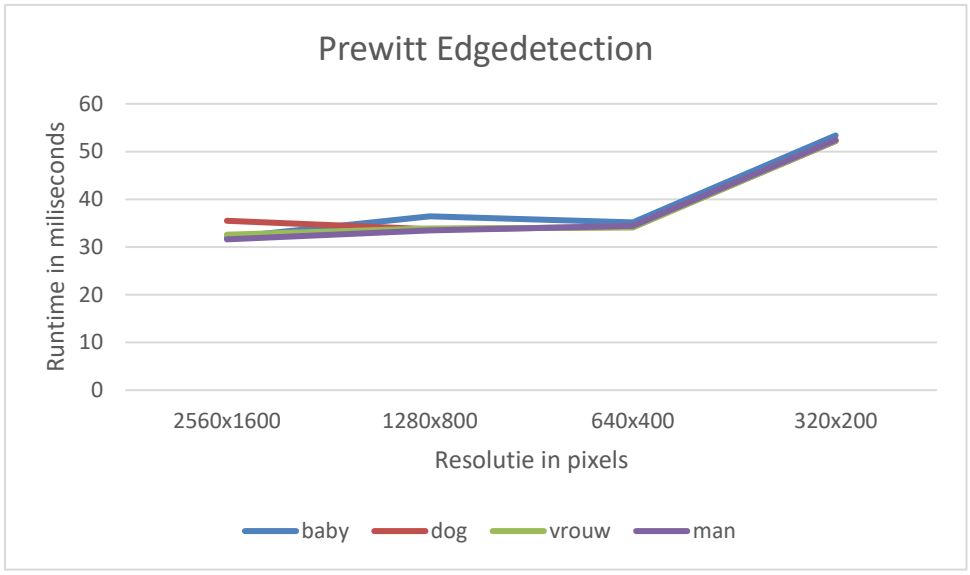
1.3. Werkwijze

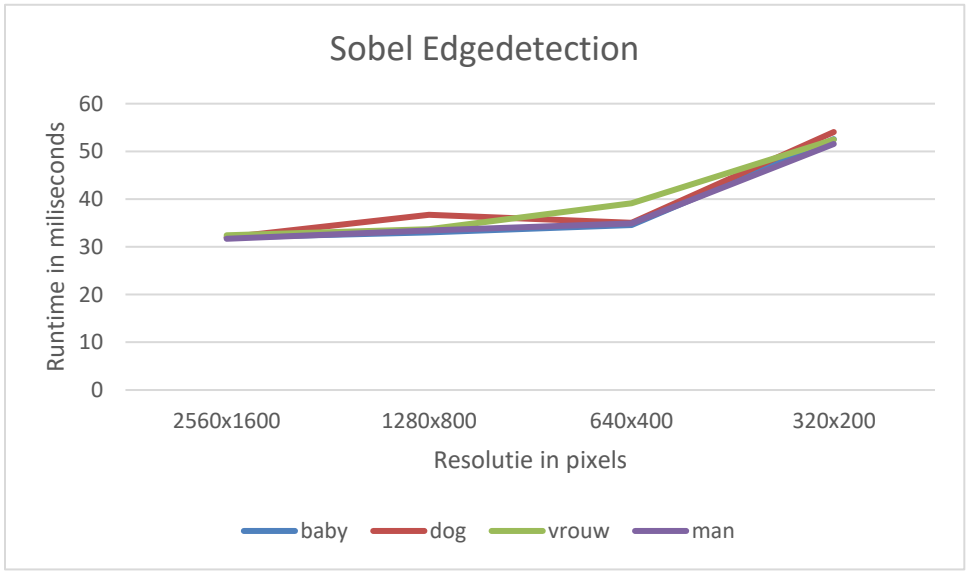
We hebben vier hoge resolutie (2560x1600) plaatjes gekozen die volgens ons interessant waren om te testen. Hoe interessant we de foto's vonden hing af van de hoeveelheid edges die we zelf al zagen of de belichting.

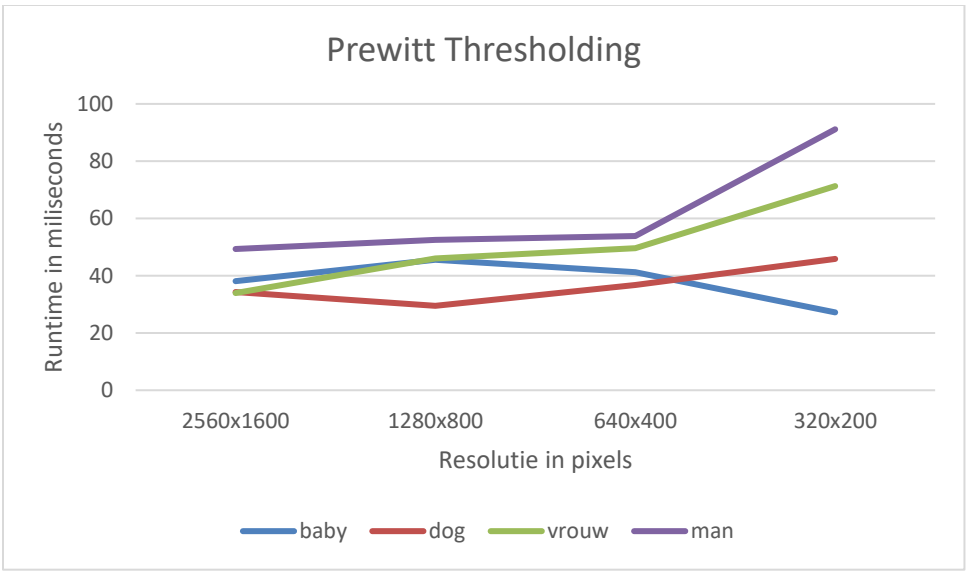


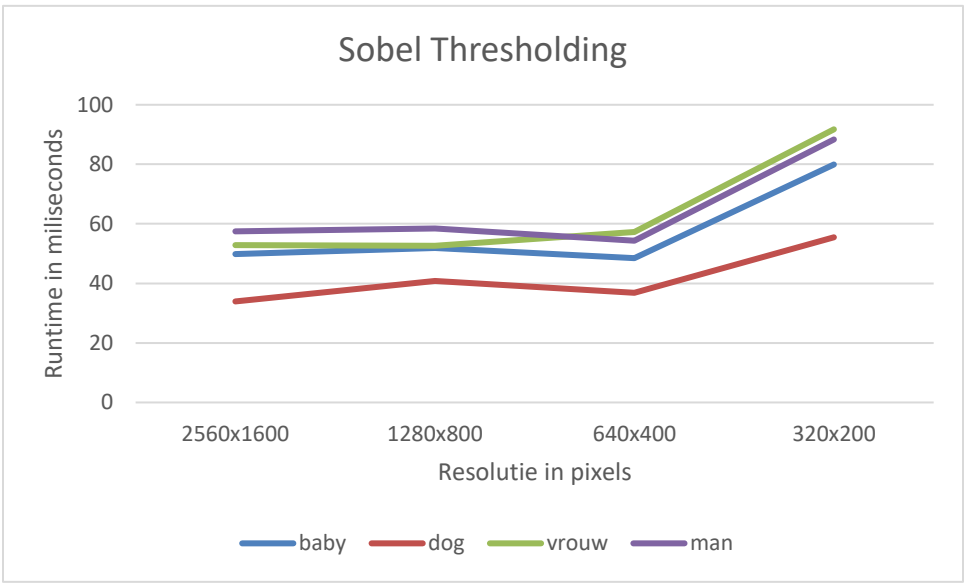
We zullen deze plaatjes stapsgewijs downscalen (2560x1600, 1280x800, 640x400, 320x200) en op deze manier met verschillende resoluties de runtime van Sobel en Prewitt vergelijken tijdens het proces voor de tres holding en de edge detection. We voor de resultaten van deze test zullen we het gemiddelde nemen van de tijd als het proces 50 keer is uitgevoerd.

1.4. Resultaten

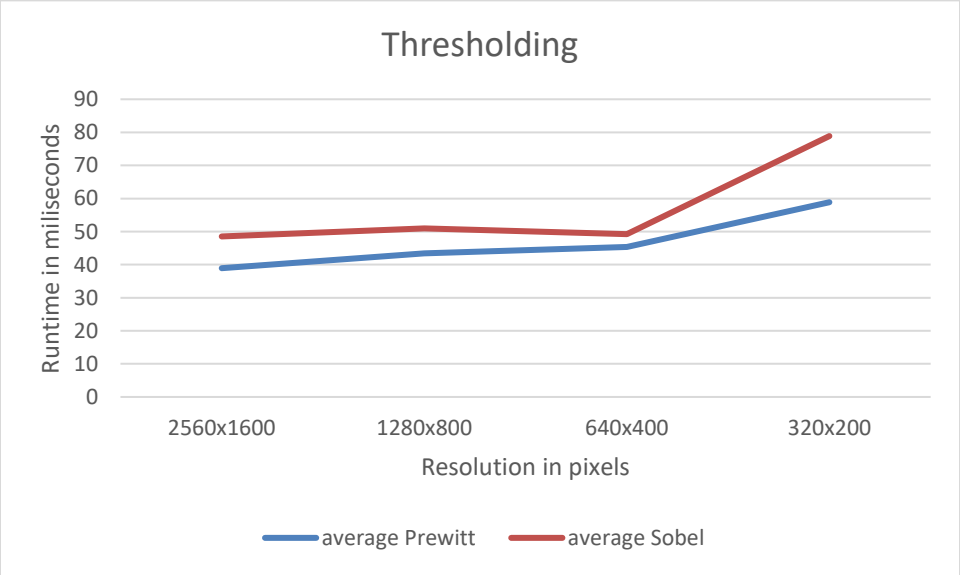





Prewitt Thresholding
Runtime in miliseconds
100
80
60
40
20
0
2560x1600
1280x800
640x400
320x200
Resolutie in pixels
baby
dog
vrouw
man


Sobel Thresholding
Runtime in miliseconds
100
80
60
40
20
0
2560x1600
1280x800
640x400
320x200
Resolutie in pixels
baby
dog
vrouw
man

1.5. Verwerking



1.6. Conclusie

Bij de edge detectie is de snelheid gelijk tussen de verschillende methodes. Onze hypothese dat de bereiking van Sobel iets zwaarder zou zijn was dus fout. De computer gaat waarschijnlijk zo efficiënt om met de berekeningen dat er geen verschil zichtbaar is.

Bij thresholding is Prewitt aanzienlijk sneller dan Sobel. Doordat er in Sobel minder edges gedicteerd worden is het verschil tussen de intensiteit tussen de voor en achtergrond groter. Daardoor zal de versimpeling van Otsu (de methode wat wij geïmplementeerd hebben) meer thresholding waarden af moeten gaan en dus meer gemiddelden moeten berekenen.

Wij kunnen dus concluderen dat Prewitt de snelste methode is om te gebruiken als je ook het proces van thresholding mee neemt in het edge detectie proces.

Toen de resolutie van 320x200 werd getest ging de runtime naar beneden. Dit is mogelijk te wijzen aan inefficiënter geheugengebruik. Alle pixels in de afbeeldingen zijn opgeslagen in array's. Grotere array's worden dichter bij elkaar in het geheugen geplaatst (geheugenaddressen opvolgend), in tegenstelling tot relatief kleine array's. Een afbeelding met een lagere resolutie leidt tot minder pixels en dus tot kleinere array's. De processor besteed daardoor meer tijd aan het 'scrapen' van gegevens over het gehele geheugen.

1.7. Evaluatie

Het doel van het experiment was om een verschil te vinden in de snelheid tussen het Prewitt en Sobel algatime. Hier verwachte wij dat Sobel langzamer zou zijn door een iets zwaardere berekeningen. Het resultaat van het experiment was dat er geen verschil zit in de snelheid tijdens edge detectie. Maar het experiment gaf ons wel onverwachte interessante resultaten. Tijdens het thresholding proces is Sobel bijvoorbeeld langzamer en dit was eigenlijk niet is wat we hadden verwacht als een resultaat wat van belang zou zijn. Dit hebben we echter getest om een betere conclusie te kunnen maken over de verkregen resultaten.

Er zijn een aantal onderdelen die de resultaten kunnen hebben beïnvloed en waar in een vervolgonderzoek verbeteringen mogelijk zijn. De computer waar we het getest hebben werd ook voor ander doeleinde gebruikt en dat veroorzaakt achtergrond processen die de snelheid kunnen beïnvloeden. Daarnaast kregen wij als resultaat uit het experiment dat de snelheid bij een resolutie van 320x200 pixels langzamer was dan bij hogere resoluties dit wijten wij aan hoe de computer omgaat met data in arrays (zoals eerder vermeld in de conclusie). Dit is een resultaat wat wij niet verwacht hadden, maar het uiteindelijke doel van het experiment (een verschil vinden in snelheid tussen het Prewitt en Sobel algoritme) kan door gedrag van de computer sterk beïnvloed worden.

Het experiment heeft dus interessante resultaten opgeleverd maar het zou beter zijn als de computer waarop getest is constant met de data om zou gaan (geen optimalisatie toepassen) en niet voor andere doeleinde gebruikt zou worden.