



MEETRAPPORT SNELHEID

Edge-Detection Methoden

[Inhoud](#)

Dit document bevat de testresultaten van het uitgevoerde onderzoek.

Jochem van Kranenburg en Joshua van Ree

18-3-2020

Inhoud

Doel	2
Hypothese	2
Werkwijze	2
Resultaten	3
Canny.....	3
Sobel	4
Laplacian	5
Verwerking	5
Conclusie	6
Evaluatie	7
Bijlagen	8
Canny Dataset	8
Laplacian dataset.....	9
Sobel dataset	10

Doel

Dit experiment wordt uitgevoerd om aan te tonen dat de door ons opgeleverde implementatie sneller werkt dan de standaard gegeven implementatie. Dit sluit dan ook aan op het doel van de gehele implementatie; *Het doel van deze implementatie is het zo snel mogelijk doch accuraat kunnen herkennen van gezichten zodat een alarmsysteem niet afgaat wanneer de eigenaar binnenkomt, maar wel afgaat bij een inbreker.*

De onderzoeksvraag luidt dan ook: Hoeveel sneller is het standaard gezichtsherkenningssysteem als deze de geïmplementeerde thresholding en edge detection stappen gebruikt in vergelijking met de standaard thresholding en edge detection stappen?

Hypothese

Aangezien de edge detection en thresholding stappen een klein deel van alle bezigheden omtrent het herkennen van gezichten betreft, vermoeden we niet meer dan vijf milliseconden sneller te zijn dan de gegeven implementatie.

De hypothese is zodoende dat het edge detection algoritme minstens vijf milliseconden sneller is dan de standaardimplementatie.

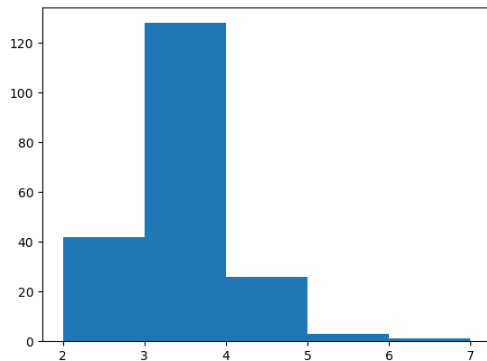
Werkwijze

Door de tijd sinds Epoch voorafgaand aan de gezichtsherkenning te meten en die af te trekken van de tijd sinds Epoch na afloop van de gezichtsherkenning kunnen we de herkenningstijd achterhalen. Daarnaast kunnen we door de tijd sinds Epoch voorafgaand aan de edge detection te meten en die af te trekken van de tijd sinds Epoch na afloop van de edge detection de detectietijd van de edges achterhalen. Deze detectie- en herkenningstijden zullen enkele honderden keren worden bepaald waarna deze geplot zullen worden in een grafiek zodat de verschillen duidelijk zijn. Indien deze verschillen consistent zijn, kunnen we concluderen wat de snelheidsimpact van de nieuwe methode is. Door van deze resultaten een normale verdeling te maken, en dus het gemiddelde en de standaardafwijking te bepalen, zal achterhaald worden of het snelheidsverschil significant is.

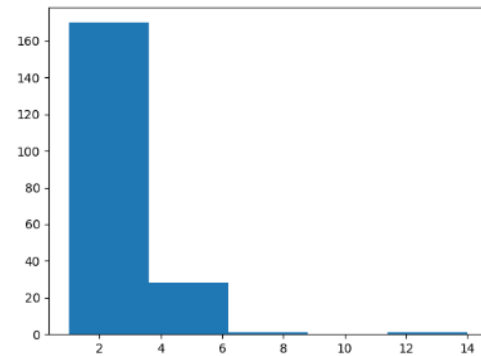
Bij al deze metingen maken we gebruik van de door de Hogeschool Utrecht aangeleverde dataset met afbeeldingen.

Resultaten

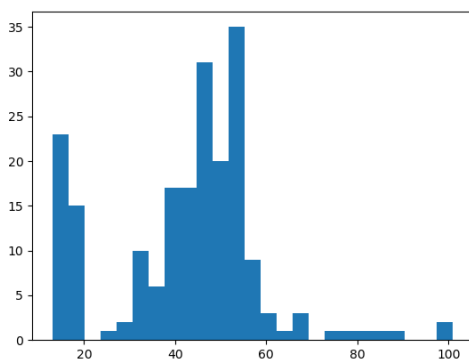
Canny



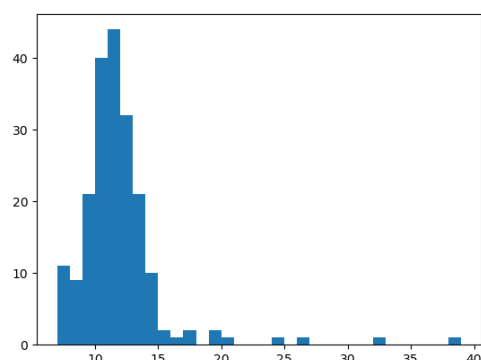
Default edge timings



Student edge timings



Default total timings



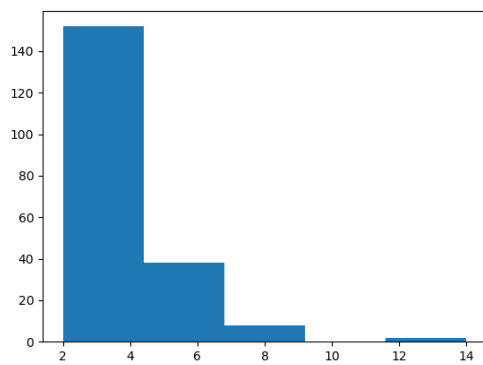
Student total timings

In de edge timing grafieken is te zien dat de gemiddelde tijd van onze implementatie (2.775ms) kleiner is dan die van de gegeven implementatie (2.975ms). Dit verschil is nog niet enorm uitgesproken. Als we dan echter kijken naar de tijd benodigd om de algehele herkenning uit te voeren, zien we wel een groot verschil; 42,565ms voor de default implementatie tegen 11,39ms voor onze implementatie.

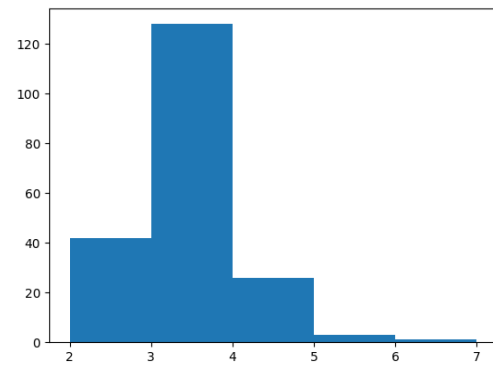
Het is opvallend dat de totale herkenningstijd afneemt bij de studentenimplementatie terwijl daar niks aan veranderd is. Wellicht komt dit door de mindere mate van matches waardoor dit niet berekend hoeft te worden?

Daarnaast is het opvallend dat de default edge detection timing 0.5 milliseconde verschilt van de overige methoden. Dit wijten we toe aan het besturingssysteem.

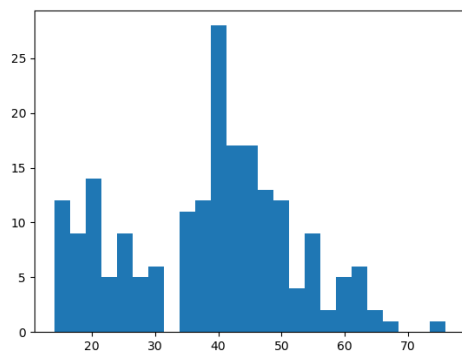
Sobel



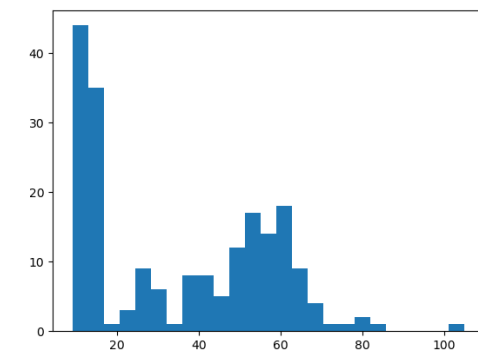
Student edge detection tijd



Default edge detection tijd



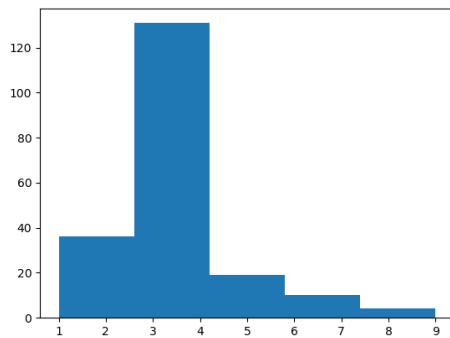
Default totale herkenningstijd



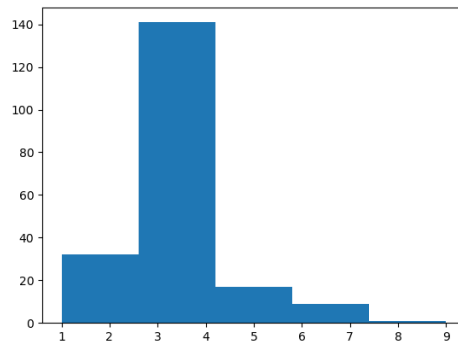
Student totale herkenningstijd

Wat betreft de tijden benodigd om de edges te detecteren is het verschil tussen de student (3.785ms) en de default (3.505ms) niet groot. Als er dan echter weer wordt gekeken naar de totale detectietijd zien we dat de student (35.595ms) drie milliseconden sneller is dan de default (38.455ms).

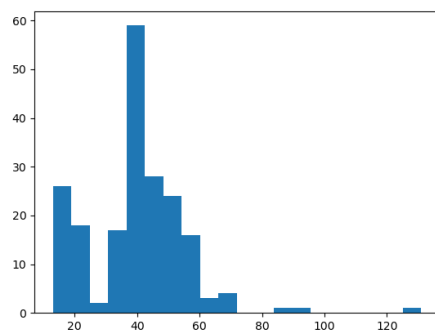
Laplacian



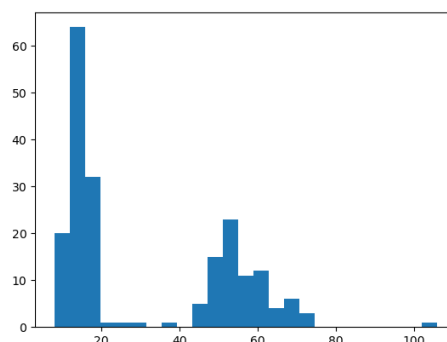
Default edge timings



Student edge timings



Figuur 1 Default total timings



Student total timings

Op het gebied van benodigde tijd om edges te detecteren, zien we dat de studentenimplementatie van Laplacian maar 3.31 milliseconden nodig hebben terwijl de default implementatie 3.525 milliseconden vereist. Bij de totale gezichtsherkenning wordt het al helemaal duidelijk; de studenten hebben dan 31.38 milliseconden nodig terwijl de standaard implementatie 39.625 milliseconden nodig heeft.

Verwerking

De Canny Operator is ruim 30 milliseconden sneller dan de default implementatie. Echter betwijfelen we het of het überhaupt mogelijk is dat de herkenningstijd drastisch verminderd terwijl de edge detection ongeveer even lang duurt.

Bij Sobel is de herkenningstijd 3 milliseconde korter. Echter steekt de Laplacian Operator hier met kop en schouders bovenuit; de herkenningstijd is verkort met ruim 8 milliseconden. Daarbij overtreffen we onze hypothese; de hypothese was 5 milliseconden sneller een gezicht te hebben gedetecteerd. De hypothese is dan ook correct.

Dit is betrouwbaar aantoonbaar aangezien de in het hoofdstuk Resultaten weergegeven diagrammen zijn bereikt door 240 herkenningsoperaties toe te passen op 6 verschillende gezichten. Daarnaast is het bereikte verschil tussen de default implementatie en de door ons geïmplementeerde Laplacian operator 8.245 milliseconden en dus significant.

Conclusie

De onderzoeksvraag is: Hoeveel sneller is het standaard gezichtsherkenningssysteem als deze de geïmplementeerde thresholding en edge detection stappen gebruikt in vergelijking met de standaard thresholding en edge detection stappen?

Als het standaard gezichtsherkenningssysteem gebruik maakt van de geïmplementeerde thresholding en edge detection stappen is het ruim 8 milliseconden sneller.

Evaluatie

Het implementatieplan hadden we al snel goedgekeurd; we konden dan ook al snel aan de slag met programmeren. Echter deed zich een probleem voor; de benodigde theorie hadden we toen nog niet gehad. Na een week of twee wachten konden we dan eindelijk beginnen.

In het begin was er wat moeite en verwarring met betrekking tot de hoe wij onze ideeën in code konden krijgen. Toen dat eenmaal gelukt was en de ontwikkelomgeving werkte is alles redelijk goed gegaan. Het was wel erg omslachtig om bij elke wijziging de DLL opnieuw te moeten compileren en weer verplaatsen, om vervolgens 1 voor 1 de afbeeldingen in de GUI te testen, dit zorgde ervoor dat het finetunen van het algoritme veel meer tijd kostte dan nodig zou zijn bij een efficiëntere ontwikkelomgeving.

Het meten van de performance presenteerde ook nog wat uitdagingen. Aangezien er geen hele goede manier is om de accurate van de edge detection te meten, hebben we geprobeerd een grotere testset te maken. Echter konden we geen afbeeldingen vinden waarmee het default en ons algoritme overweg kon. Een grotere testset zou dus geen toegevoegde waarde hebben geleverd.

Uiteindelijk zijn we tevreden met het eindresultaat en hebben we er veel van geleerd.

Bijlagen

Deze dataset kan worden gegeneerd door in de `StudentPreprocessing::stepEdgeDetection` methode van methode te wisselen. De in het document gebruikte diagrammen zijn gegeneerd met `matplotlib`.

Canny Dataset

Student total timings:

[illegible]

Default total timings:

[13,13,13,14,14,14,14,14,14,15,15,15,15,15,15,15,16,16,16,16,16,16,16,17,17,17,18,18,18,18,18,18,19,19,19,19,20,27,30,30,31,33,33,33,33,33,33,33,34,34,35,35,35,36,37,37,38,38,38,39,39,39,39,39,39,40,40,40,40,41,41,41,42,42,42,42,42,42,43,43,43,43,43,43,43,44,44,44,45,45,45,45,45,46,46,46,46,46,46,46,46,47,47,47,47,47,47,48,48,48,48,48,48,48,48,48,49,49,49,49,49,50,50,50,50,50,50,50,50,50,51,51,51,52,52,52,52,52,52,52,52,52,52,52,53,53,53,53,53,53,53,53,53,54,54,54,54,54,54,54,54,55,55,55,55,55,56,56,57,57,57,57,57,58,58,59,60,62,65,66,67,67,73,78,81,86,88,98,101,]

Student edge timings:

[6,3,4,4,3,4,5,4,4,4,3,2,3,3,3,3,3,4,3,2,3,3,2,3,3,3,3,3,3,4,2,3,3,2,3,3,2,3,3,4,2,3,4,3,3,3,2,2,3,3,3,3,3,
4,3,3,4,3,3,3,3,3,4,3,4,3,3,3,3,3,3,3,2,2,3,3,3,3,3,2,2,2,3,2,1,2,2,4,2,3,3,2,2,2,2,2,2,2,2,2,2,2,2,2,2,
2,3,2,3,2,4,4,2,2,2,2,2,3,3,3,3,3,3,2,2,3,2,3,4,3,3,3,3,3,3,3,3,4,3,
3,3,3,14,3,3,3,2,3,2,3,4,4,2,4,3,3,4,3,2,3,3,3,3,3,3,3,3,3,4,3,3,4,3,3,3,7,4,3.]

Default edge timings:

[7,4,4,3,4,3,3,3,3,3,3,3,3,3,3,3,4,3,3,2,4,3,3,3,3,3,3,3,4,2,3,3,3,4,3,3,3,3,3,3,3,4,3,4,3,3,3,3,3,5,4,4,
4,3,4,3,4,4,4,3,3,3,4,3,3,3,3,3,2,3,3,3,2,2,3,3,3,2,2,2,4,2,2,2,2,3,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,
2,2,3,3,5,4,2,2,2,2,2,2,3,4,3,3,3,3,3,3,3,3,3,3,3,3,4,3,3,3,3,3,4,3,3,
3,3,3,3,3,3,3,3,3,2,3,3,2,3,3,3,3,5,2,3,3,3,3,3,3,3,4,3,3,4,3,4,3,3,4,3,3,]

Accuracy:

[illegible]

[101,
101,101,101,101,101,101,101,101,101,101,101,101,101,101,101,101,103,103,103,103,103,1
03,10
3,103,103,103,103,103,103,103,103,103,104,104,104,104,104,104,104,104,104,104,104,
104,
104,104,104,104,105,105,105,105,105,105,105,105,105,105,105,105,105,105,105,105,105,
105,105,105,105,105,105,105,105,105,105,105,105,105,105,105,105,105,105,108,108,10
8,108,
108,108,108,108,108,108,108,108,108,108,108,108.]

Sobel dataset

Student total timings:

[illegible]

Default total timings:

[14,14,14,14,14,15,15,15,16,16,16,16,16,17,17,17,17,18,18,18,18,18,18,19,19,19,19,19,19,19,19,20,20,21,21,21,21,22,22,22,23,23,24,24,24,24,25,26,26,26,26,27,27,27,27,28,29,29,30,30,30,31,34,35,35,35,35,35,35,36,36,36,37,37,37,37,37,37,37,37,37,38,38,38,39,39,39,39,39,39,39,39,40,40,40,40,40,40,40,40,40,40,40,40,40,40,41,41,41,41,41,41,42,42,42,42,42,42,42,42,42,43,43,43,43,43,43,43,43,44,44,44,44,44,45,45,45,45,45,45,45,46,46,46,46,46,47,47,47,47,47,47,47,47,48,48,48,48,48,48,49,49,49,49,49,49,50,50,50,51,51,51,52,52,53,53,54,55,55,55,55,55,56,56,56,58,58,60,60,60,61,61,62,62,62,62,63,63,64,64,68,76,]

Student edge timings:

[9,7,3,3,3,6,5,3,3,3,3,5,6,3,6,3,3,3,5,7,3,3,3,3,4,4,3,4,3,4,8,3,3,3,4,4,3,3,7,3,3,4,3,3,5,3,3,4,3,5,4,5,3,3,3,4,3,4,5,4,14,6,3,6,4,5,7,3,4,4,3,13,4,4,3,7,4,5,2,4,2,2,2,3,4,2,2,4,2,4,2,2,2,2,2,2,2,2,2,2,3,2,2,2,6,2,2,2,3,2,2,2,2,2,2,2,6,3,4,3,4,4,3,3,3,4,4,3,4,4,3,3,5,4,6,3,4,3,6,3,3,3,4,5,3,3,5,3,6,6,3,6,5,3,4,3,3,5,3,4,5,3,5,3,3,3,6,5,3,3,3,4,3,4,3,4,4,7,3,3,6,5,3,5,6,3,3,5,5,4,3,4,5,3,3,5,]

Default edge timings:

[6,5,8,3,4,4,3,4,3,5,4,2,4,2,4,3,4,3,3,5,3,3,5,3,5,3,2,3,4,3,5,3,3,5,3,3,3,2,5,3,3,6,4,3,3,3,2,4,2,4,3,3,5,3,3,4,6,3,3,2,3,3,3,3,7,3,3,3,3,4,4,3,4,3,5,4,3,4,2,2,2,2,2,2,2,2,2,2,4,2,2,2,3,2,2,2,2,5,1,2,2,2,2,2,2,2,2,2,2,2,3,1,2,2,3,4,3,3,3,3,4,3,4,4,5,3,4,2,2,3,4,3,4,3,4,4,3,4,3,3,3,3,3,5,4,5,3,3,3,3,3,3,3,5,3,3,5,2,3,6,3,6,5,3,3,3,5,3,5,3,6,3,3,3,6,4,3,4,3,5,3,3,3,3,3,6,5,5,3,3,]

Accuracy:

[illegible]