



MEETRAPPORT

RGB naar intensity

Jasper Oosterbroek & Jens Bouman

31-3-2019

Inhoud

Doel	2
Hypothese	2
Werkwijze	2
Snelheid.....	2
Kwaliteit	2
Test data	3
Resultaten	4
Snelheid.....	4
Kwaliteit	5
Verwerking.....	6
Conclusie.....	6
Evaluatie.....	6

Doel

In dit meetrapport willen wij erachter komen welk van de 3 implementaties het beste kan worden gebruikt in het facial recognition programma. Dit doen wij door de volgende onderzoeksvraag op te stellen: Welk van de drie geïmplementeerde kleur-naar-intensiteits-conversies levert de beste snelheid naar resultaat verhouding op?

Hypothese

De hypothese is dat de methode dat de RGB kanalen bij elkaar optelt en dan door drie deelt sneller is dan de andere twee implementaties maar de methode dat een gewogen gemiddelde berekent een beter resultaat geeft dan de vorig genoemde methode. Ook verwachten wij dat beide eerder genoemde methode sneller zijn dan de standaard implementatie.

Werkwijze

Om onze deelvraag te kunnen beantwoorden moeten wij de snelheid van de conversie en de resultaten van de facial recognition meten.

Beide tests gaan in een vermenigvuldigingsmaat van 10 en wordt de gemiddelde tijd of score gebruikt als resultaat.

Aangezien kwaliteit moeilijk aan te geven is wordt er gekeken naar de afwijking van de facial parameters ten opzichten van de standaard implementatie.

Snelheid

Voor het testen van de snelheid laten wij het programma de conversie uitvoeren met een tijds meting die start net voor de conversie en weer stopt nadat de conversie gedaan is. Bij de grotere metingen meten wij de tijd die het kost om alle metingen te doen en slaan dit op. Om deze metingen uit te voeren gebruiken wij vision timer die specifiek hiervoor gemaakt is. Deze timer is via de volgende URL te vinden <https://github.com/arnokamphuis/vision-timer>.

Kwaliteit

Voor het meten van de kwaliteit nemen wij de standaard implementatie als base line en hoe meer het resultaat van de andere methode afwijkt hoe slechter het is. Voor data gebruiken wij de facialParameters die in debug mode in de console te zien zijn.

Test data

Voor de test gaan wij de volgende afbeelding als test data gebruiken:



Child-1.png

Resultaten

De volgende resultaten zijn verkregen door de eerder beschreven werkwijzen te gebruiken.

- Methode 1 is de methode de drie RGB waarden bij elkaar optelt en het deelt door drie.
- Methode 2 is de methode die de RGB waarden met een gewicht vermenigvuldigt.
- De standaard methode is de meegeleverde implementatie.

Snelheid

De volgende tabel geeft de totaal kostende tijd aan per sample size.

<i>Sample size</i>	<i>10</i>	<i>100</i>	<i>1000</i>	<i>10000</i>
<i>Methode 1</i>	47ms	448ms	3767ms	36128ms
<i>Methode 2</i>	42ms	457ms	3665ms	35581ms
<i>De standaard methode</i>	99ms	735ms	6464ms	65707ms

De volgende tabel geeft de gemiddelde run time per methode per sample size.

<i>Sample size</i>	<i>10</i>	<i>100</i>	<i>1000</i>	<i>10000</i>
<i>Methode 1</i>	4.7ms	4.5ms	3.8ms	3.6ms
<i>Methode 2</i>	4.2ms	4.6ms	3.7ms	3.6ms
<i>De standaard methode</i>	9.9ms	7.3ms	6.5ms	6.6ms

In de tabellen is goed te zien dat beide methode sneller zijn dan de standaard implementatie. Methode 1 en methode 2 zitten erg dicht op elkaar, dit kan komen omdat de uiteindelijke hoeveelheid berekeningen erg dicht op elkaar liggen.

Kwaliteit

In de volgende tabel zijn de droge resultaten van de test te zien.

	STANDAARD	METHODE 1	METHODE 2
1	2.89129	2.89129	2.89129
2	0.422661	0.422661	0.422661
3	0.622484	0.622484	0.622484
4	0	0	0
5	1.26055	1.25492	1.26055
6	0.870693	0.870693	0.870693
7	1.0407	1.04837	1.04801
8	2.88449	2.88449	2.88449
9	0.552073	0.552073	0.552073
10	0.0659228	0.0659228	0.0659228
11	1.26748	1.26748	1.26748
12	0.366161	0.366161	0.366161
13	0.395332	0.395332	0.395332
14	0.647823	0.647823	0.647823
15	1.59526	1.59526	1.59526
16	3.46154	3.46154	3.46154

In de volgende tabel is het verschil in vergelijking met de standaard te zien

	STANDAARD	METHODE 1	METHODE 2
1	0	0	0
2	0	0	0
3	0	0	0
4	0	0	0
5	0	-0.00563	0
6	0	0	0
7	0	0.00767	0.00731
8	0	0	0
9	0	0	0
10	0	0	0
11	0	0	0
12	0	0	0
13	0	0	0
14	0	0	0
15	0	0	0
16	0	0	0

In de tabel is goed te zien dat er weinig verschil zit in alle methode als we het over resultaat hebben.

Verwerking

Onze hypothese was dat de twee te implementeren methode sneller was, dit is aan de tabellen onder het kopje snelheid goed te zien. Beide nieuwe implementaties zijn bijna twee keer zo snel als de standaard implementatie en de resultaten gezien in de kwaliteit tabellen zijn bijna hetzelfde. Dit betekend dat de nieuwe methode bijna dezelfde kwaliteit leveren ten opzichten van de standaard implementatie. De 2^e methode is meer richting de standaard wat goed in de tabel is te zien, dit hadden wij ook al verwacht.

Conclusie

Aan de hand van de resultaten en verwerking is te zien dat methode 1 en 2 beter zijn in de snelheid dan de standaard methoden. Waar wel verschil in methode 1 en 2 zit is de resultaten in vergelijking van elkaar, methode 2 geeft een resultaat dat dichterbij de standaard implementatie ligt. Dit betekend dat methode 2 een beter snelheid kwaliteit verhouding heeft.

Evaluatie

Het proces van evaluatie plan naar het meetrappport ging best vloeiend. Er zat in verhouding erg veel tijd in het implementatie plan omdat er veel manieren waren om de conversie te implementeren. De implementatie zelf ging erg vlot er gebeuren werden geen gekke dingen gedaan en de gekozen conversie manieren waren erg simpel om te implementeren. Het meetrappport was erg interessant om uit te voeren en de resultaten waren al helemaal verrassent. In de toekomst moet er wel meer gebeuren aan de hand van testen aangezien de resultaten van dit ene plaatje erg op elkaar lag per methode. Dit had achteraf met meerdere plaatjes gedaan moeten worden.