

Meet rapport Grayscale algorithms



Gemaakt door Bob Thomas en Robbie Valkenburg

Technische informatica 2016-2017

Datum 6 maart 2017

1. Doel

Wij gaan het experiment vastleggen doormiddel van een controleerbare hypothese. Graag zouden wij onze implementatie van 'intensity' en 'luminance' willen vergelijken met de bestaande code. Onze onderzoeksvraag luidt: "Geeft minstens een van onze geïmplementeerde code een betere snelheidsperformance dan de bestaande code op de geleverde test afbeeldingen?".

2. Hypothese

De verwachting is dat het wel mogelijk kan zijn om een snellere performance te leveren als de bestaande code. Echter zijn we er niet direct van overtuigd dat wij dit daadwerkelijk gaan opleveren.

3. Werkwijze

Er zal zowel bij het uitvoeren van de bestaande code als de door ons geschreven code een executie tijd moeten worden bijgehouden. Meerdere images zullen meerdere keren getest worden tijdens het experiment.

4. Resultaten

De executie tijd is het gemiddelde van de 1000 executies in milliseconden

Als er een **x** naast de tijd staat betekend het dat de volledige herkenning niet werkte

En een **✓** Betekend dat het de volledige herkenning flow met succes afmaakte.

Voor het 'single color channel' algoritme hebben we het groene kanaal gepakt.

We hebben voor het groene kanaal gekozen omdat we het beste resultaat er mee kregen

Plaatje	Executie tijd	Executie tijd	Executie tijd	Executie tijd
	Bestaande code	'Intensity'	'Luminance'	'Single color'
Female-1.png	23.595 ms ✓	14.752 ms ✓	15.157 ms ✓	10.69 ms ✓
Female-3.png	21.76 ms ✓	13.189 ms ✓	15.813 ms ✓	9.746 ms ✓
Child-1.png	22.252 ms ✓	13.082 ms ✓	15.839 ms ✓	10.075 ms ✓
Female-2.png	8.664 ms ✓	3.377 ms. X	3.959 ms ✓	2.331 ms ✓
Male-1.png	27.884 ms ✓	14.115 ms ✓	16.675 ms ✓	9.916 ms ✓
Male-2.png	27.648 ms ✓	12.926 ms ✓	17.111 ms ✓	10.035 ms ✓
Male-3.png	27.518 ms ✓	18.747 ms X	17.979 ms ✓	10.643 ms ✓

5. Verwerking

Uit de resultaten is gebleken dat onze code minder executie tijd nodig heeft. Dit is voor zowel voor beide algoritmes gebleken. De oorzaak dat onze code sneller is, is vanwege het feit dat de bestaande code met een library werkt. Voor het aanroepen van de library is er meer tijd nodig dan rechtstreeks onze code uitvoeren in de “studentPreProcessing.cpp”. Uit de resultaten is gebleken dat onze ‘intensity’ algoritme bij 2 van de 7 testen niet geldig waren verklaard. De reden van ongeldigheid was de lokalisatie die niet uitgevoerd kon worden. En met verassing werkte het color channel algoritme het snelst en het slaagde ook in het lokaliseren en herkennen.

6. Conclusie

Uit ons experiment is gebleken dat minstens een van onze implementatie code een betere snelheid performance levert. Onze ‘intensity’ algoritme was niet bij elke afbeelding succesvol. Hierdoor is besloten onze ‘intensity’ niet geldig te klaren als een betere performance. Echter was dit wel het geval bij het ‘Luminance’ algoritme wat door ons geïmplementeerd is. Hiermee is vastgesteld dat een van onze implementatie een betere snelheidsperformance leverde.

7. Evaluatie

Eerlijk gezegd had ik niet verwacht dat onze code daadwerkelijk een snellere executie tijd zou hebben. Bij nader inzien is het wel logisch aangezien het feit dat er een library worden gebruikt in de bestaande code waardoor er meerdere calls gemaakt moeten worden. Ook hadden we niet verwacht dat simpel algoritme als single color channel zo goed zou werken als we tenminste het groene kanaal gebruikte want met blauw en rood kregen we falende resultaten.