Meetrapport snelheid conversie

Florian Humblot - 1720570

Vera Schoonderwoerd - 1721202

Datum: 22 februari 2019

Inhoudsopgave

[Doel 3](#_Toc4330405)

[Hypothese 3](#_Toc4330406)

[Werkwijze 3](#_Toc4330407)

[Resultaten 4](#_Toc4330408)

[Verwerking 5](#_Toc4330409)

[Conclusie 6](#_Toc4330410)

[Evaluatie 6](#_Toc4330411)

## Doel

Het doel van dit meetrapport is het testen hoe veel sneller onze implementatie is, ten opzichte van de default implementatie. Dit onderzoek is met betrekking van zowel onze container als onze intensity conversie.

## Hypothese

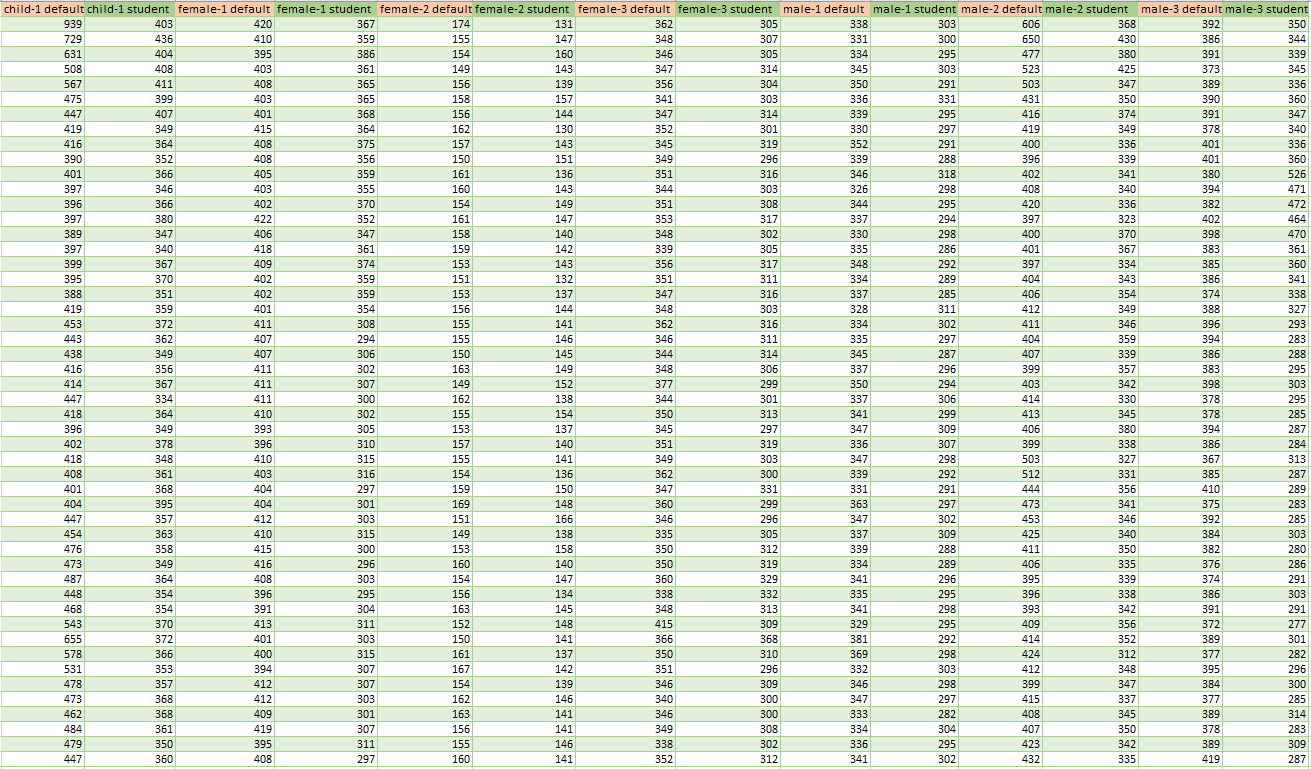
Wij verwachten dat onze implementatie 30% sneller zal zijn, omdat onze implementatie maar met 1 RGB kleur rekent. Wij verwachten dat de default conversie namelijk met 3 RGB waarden rekent.

## Werkwijze

Deze test gaan wij met 7 verschillende fotos doen. Per foto wordt het inladen van de foto, met onze zelfgemaakte methode, en het converteren naar intensity 100 keer aangeroepen: 50x met de default methode en 50x met onze eigen methode. De code is in debug-mode gerunned, zonder flag optimalizaties. Om de tijd bij te houden gebruiken wij de functie std::clock(). De beginwaarden en de eindwaarden halen we van elkaar af en dit delen we door (CLOCKS\_PER\_SEC/1000). Hieruit volgt de conversietijd in milliseconden. Deze waarden schrijven we naar een CSV bestand, om de resultaten makkelijk naar Excel te kunnen converteren. Als alle 700 conversies gedaan zijn, gaan we het gemiddelde berekenen per foto per methode.

## Resultaten

De resultaten van het experiment zijn in onderstaande tabel weergegeven. Zoals te zien is, zijn er per foto 50 conversies geweest van de default methode en 50 conversies van onze eigen implementatie. De default implementatie is rood gekleurd, de studentversie is groen gekleurd.

**

Tabel

## Verwerking

Aan de hand van tabel 1 zijn er per kolom van de 50 conversies een gemiddelde berekend via de onderstaande formule:

Voorbeeld bij child-1 default: = 466,8 milliseconden

Door dit bij alle kolommen uit te voeren, kwamen wij uit op de volgende grafiek:

Grafiek 1

Zoals in grafiek 1 te zien is, is onze conversie bij elke foto sneller. Het kleinste verschil zit bij female-2, de default conversie is 9% langzamer dan onze conversie. Het grootste verschil zit bij child-1. Hierbij is de default conversie 27% langzamer dan onze conversie.

## Conclusie

Uit de grafiek is gebleken dat onze conversie sneller is dan de default conversie. Dit snelheidsverschil is wel per foto afhankelijk. Onze hypothese klopte maar deels. Onze conversie is dus inderdaad sneller, maar niet 30% zo snel. Ons grootste verschil is maar 27% en de kleinste is 9%.

## Evaluatie

Zoals in de conclusie is geschreven, onze hypothese klopte maar deels. Er zijn natuurlijk veel meer stappen die het programma aflegt naast de conversieberekening. Hier hadden wij geen rekening mee gehouden.

Het doel van dit experiment was om te controleren of onze implementatie sneller is dan de default implementatie. Dit doel hebben wij goed kunnen bereiken.

Wij hebben tijdens dit experiment geen grote meetfouten kunnen ontdekken. Het enige wat misschien invloed heeft gehad, is dat wij eerst de default methode op een foto hebben toegepast, en toen pas onze student versie. Het kan zijn dat het ophalen van de foto op een ander adres de eerste paar keren in de default methode daarom iets langer duurde dan normaal. De andere keren kan het adres van de foto in de cash zijn opgeslagen, waardoor de executietijd minder lang zou duren.