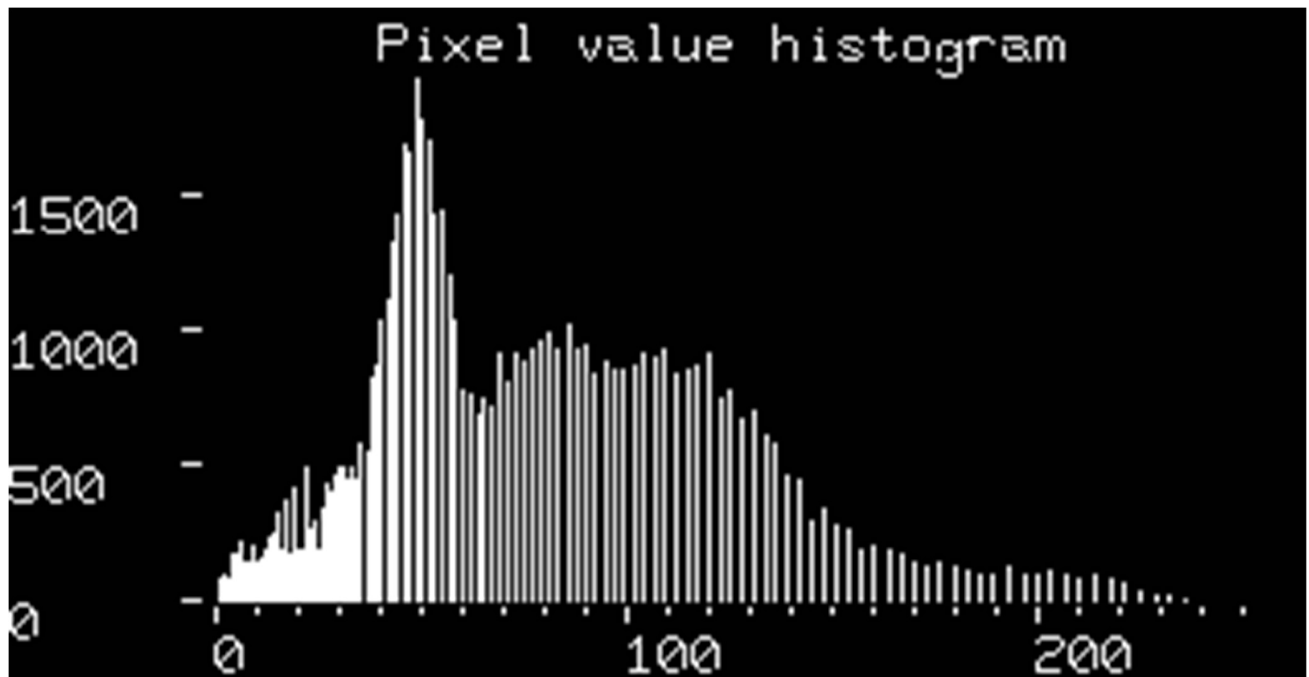


## Meetrapport RGB to Intensity algoritmes – speed



Jeremy ruizenaar 21-06-17

## Doel

In dit experiment zal er onderzocht worden welk algoritme het snelste een afbeelding om kan zetten naar een intensityimage.. De resultaten uit dit onderzoek kunnen gebruikt worden om een geschikt algoritme uit te kiezen. De onderzoeksvraag luid “welke algoritme kan het snelst een RGBimage omzetten naar een Intensityimage?”.

## Hypotheses

### Single color channel algoritme

Verwacht wordt dat het single-color-channel algoritme het snelst werkt. Dit omdat de uitgevoerde code geen ingewikkelde rekenkundige operaties bevat.

### Luminance correction algoritme

Voor het luminance-correction algoritme wordt verwacht dat dit algoritme het langzaamst werkt. Dit vanwege de vele floating-point operaties die uitgevoerd moeten worden.

### Average algoritme

Bij het average algoritme wordt de snelheid tussen de twee voorgaande algoritmes geschat. Dit vanwege de wat eenvoudigere delingen die uitgevoerd worden. In vergelijking met het luminance correction algoritme

## Werkwijze

De algoritmes worden stuk voor stuk getest op de volgende manier: als het algoritme start wordt de tijd opgeslagen in een start variabele. zodra het algoritme 100000x uitgevoerd is wordt de tijd nogmaals opgeslagen in een end variabele. Vervolgens wordt het verschil tussen deze twee tijden gedeeld door het aantal iteraties in dit geval 100000x. de uitkomst is de gemiddelde executie tijd van het algoritme. Dit wordt gedaan voor twee afbeeldingen. Deze worden hieronder weergegeven.



## Resultaten

Name / Time	Average	Luminance correction	Single color channel
Avg Execution time over 100000 iterations male	13352 ms	13497 ms	12144 ms
Avg Execution time over 100000 iterations female	12286 ms	12394 ms	11173 ms

## Verwerking

Aan de hand van de tabel bij de resultaten kan afgeleid worden dat het single-color-channel algoritme, het average algoritme, en het luminance-correction algoritme respectievelijk eerste, tweede en derde staan op basis van snelheid bij beide afbeeldingen. Het verschil tussen het average algoritme en luminance correction algoritme is niet zo groot.

## Conclusie

Na de verwerking van de meetresultaten kan de conclusie getrokken worden dat het single-color-channel algoritme constant het snelst is van alle algoritmes. Ook is er marginaal verschil tussen het luminance correction algoritme en het average algoritme.

## Evaluatie

Na uitvoeren van het experiment kunnen we met zekerheid zeggen dat het single-color-channel algoritme het snelst werkt van de drie gekozen oplossingen. Dit komt overeen met de hypothesen die gesteld zijn. Ook het feit dat luminance correction op de derde plaats staat vanwege de wat meer complexe rekenkundige operaties komt overeen met de hypothesen. Toch is het marginale verschil tussen het luminance correction algoritme en het average algoritme opvallend aangezien de verwachtingen anders waren.

