

Implementatie Plan *ImageShell & Intensity*

# Intensity

Gemaakt door : Bob Thomas en Robbie Valkenburg

Technische informatica 2016 – 2017

Datum : 13 maart 2017

## 1.1. Doel

Het doel van dit project is een digitale afbeelding omzetten naar een intensity image. Gebaseerd op de grijswaarden van het plaatje in plaats van de 3 kleur kanalen (RGB). Dit zal het belast op het geheugen verminderen en de snelheid van calculaties gebaseerd op intensiteit verhogen.

## 1.2. Methoden

Wij hebben onderzoek gedaan naar verschillende algoritmes om een intensity image (afbeelding gebaseerd op grijs tinten).

De verschillende algoritmes die we ontdekt hebben en voor ons interessant leken zijn.

Alle ondergenoemde algoritmes gaan per pixel

- **Intensity algoritme**  
Dit is een van de meest bekende en simpelste manieren om een intensity image te creëren. Gemiddelde van de 3 RGB kanalen en dat word de intensiteit.  
$$G_{intensity} = \frac{1}{3}(R + G + B)$$
- **Gleam algoritme**  
Dit algoritme is bijna precies het zelfde alleen hier gaan we er van uit dat het over afbeelding gaan waarbij de kanalen gamma correctie hebben ondervonden. Bij de formule gebruikt ' om te laten zien dat dit een kanaal met gamma correctie is  
$$GG_{leam} = \frac{1}{3}(R' + G' + B')$$
- **Luminance**  
Dit algoritme is gemaakt om de helderheid te matchen met de perceptie van het menselijk zicht. Het is niet compleet ontworpen om de menselijke perceptie van helderheid na te bootsen maar door gewicht aan de RGB kanalen te hangen komt dit algoritme aardig in de buurt.  
$$GL_{uminance} = (0.3 * R + 0.59 * G + 0.11 * B)$$
  
Dit algoritme word ook gebruikt in [GIMP](#) om afbeelding om te zetten naar grijs tinten. Ook word deze methode gebruikt in matlab om RGB naar grijswaarden om te zetten.  
[matlab-rgb2array](#)
- **Luma**  
is het zelfde als Luminance alleen met andere gewichten en gamma correctie op de waardes.  
$$GL_{uma} = 0.2126 * R' + 0.7152 * G + 0.0722 * B'$$
- **Lightness**  
Is een algoritme wat werkt in de CIELAB en CIELUV kleur ruimtes aangezien wij met RGB werken lijkt mij deze geen kandidaat voor ons project.  
$$Y = 0.2126R + 0.7152G + 0.0722B$$

$$G_{Lightness} \leftarrow \frac{1}{100} (116f(Y) - 16),$$

$$f(t) = \begin{cases} t^{1/3} & \text{if } t > (6/29)^3 \\ \frac{1}{3} \left(\frac{29}{6}\right)^2 t + \frac{4}{29} & \text{otherwise.} \end{cases}$$

- **Value**

is een algoritme wat de max neemt van de 3 RGB kanalen dit is natuurlijk een algoritme wat ons niet duidelijk informatie kan geven aan gezien de intensiteit erg hoog of erg laag zonder duidelijke verschillen.

$$GValue = \max(R, G, B)$$

- **Luster**

Is een algoritme wat de gemiddelde van de max en min van de RGB waardes neemt. Waar door die minder intense is dan Value maar nog steeds niet erg bruikbaar is voor Vision detection problemen

[Kanan, C., & Cottrel;, G. W. \(2012\).](#)

### **1.3.      Keuze**

We zijn aan het twijfelen tussen 2 methodes en die zijn Luminance/luster en Gleam.

Na dat ik de paper van [Kanan, C., & Cottrel;, G. W. \(2012\)](#) had gelezen kwam bij hun testen eruit dat Gleam de beste resultaten gaf. Zelf weten we er nog vrij weinig van af behalve de resultaten uit het artikel. Maar aangezien Matlab en GIMP het luminance algoritme gebruiken lijkt mij dit ook een sterke kandidaat en zeker omdat Matlab het gebruikt voor computer vision doeleinden.

De methode die we gekozen hebben is Luminance omdat we willen onderzoeken hoe snel het is en of de resultaten kunnen vergelijken tegen de implementatie van GIMP of Matlab wat ons wel een interessante test leek

### **1.4.      Implementatie**

Wij hebben er voor gekozen om te werken in Microsoft Visual studio 2015. Dit omdat het project grotendeels is ontwikkeld in Microsoft Visual studio 2015. Vanuit deze omgeving hebben wij de code geïmplementeerd.

### **1.5.      Evaluatie**

De metingen die we gaan doen bestaat uit.

- Snelheid van het proces vergeleken de bestaande implementatie
- Memory efficiency vergeleken de bestaande implementatie

## 2. Bronnen

Kanan, C., & Cottrell, G. W. (2012). *Color-to-Grayscale: Does the Method Matter in Image Recognition?*. Retrieved from

[http://tdlc.ucsd.edu/SV2013/Kanan\\_Cottrell\\_PLOS\\_Color\\_2012.pdf](http://tdlc.ucsd.edu/SV2013/Kanan_Cottrell_PLOS_Color_2012.pdf)

(<https://nl.mathworks.com/help/matlab/ref/rgb2gray.html>)<https://nl.mathworks.com/help/matlab/ref/rgb2gray.html>

(<https://www.gimp.org/>)<https://www.gimp.org/>