

1. Implementatieplan – Grayscale conversion

1.1. Namen en datum

Auteurs: Stephan Vivie en Mathijs van Bremen

Klas: V2B

Datum: 18-04-2016

1.2. Doel

Het doel van deze implementatie is om een functie te maken die een “RGBImageStudent” om kan zetten naar “IntensityImageStudent”. Hierbij willen we een zo’n duidelijk mogelijke afbeelding behouden die alleen de Intensity bevat.

1.3. Methoden

We hebben een aantal manieren gevonden waarop we een RGB image om kunnen zetten naar een Intensity image. Deze manieren zijn hieronder te vinden.

Average

De eerste manier is het nemen van het gemiddelde van de RGB waarden. De berekening hiervoor ziet er als volgt uit:

$$\text{Intensity} = (R + G + B) / 3$$

Desaturation

De tweede manier neemt de minst en meest intensieve kleur een deelt dit door twee. De berekening bij deze methode ziet er als volgt uit:

$$\text{Intensity} = (\text{Max}(R, G, B) + \text{Min}(R, G, B)) / 2$$

Luminosity

De derde en laatste methode waar we naar gekeken hebben houdt rekening met hoe het oog kleuren ziet. Voor deze methode zijn verschillende berekeningen beschikbaar, maar wij hebben gekozen voor de berekening volgens ITU-R (zie BT.709 <https://www.itu.int/rec/R-REC-BT.709/en>). Deze berekening ziet er als volgt uit:

$$\text{Intensity} = R * 0.2126 + G * 0.7152 + B * 0.0722$$

1.4. Keuze

Op basis van de gegeven methoden en de tests die hiermee gedaan zijn, hebben we gekozen voor de derde methode: Luminosity. Voor elke methode is een afbeelding van RGB naar Intensity omgezet. Daarna zijn deze afbeeldingen met elkaar vergeleken om zo een keuze te kunnen maken. Deze afbeeldingen zijn te vinden in bijlage 1. Uiteindelijk bleek dat er weinig verschil te zien was tussen deze afbeeldingen, dus is er nog meer onderzoek gedaan. Uiteindelijk bleek dat Luminosity vaak gebruikt wordt, zelfs door bepaalde programma’s voor het bewerken van afbeeldingen, zoals GIMP (<https://www.gimp.org/>). Dit gaf dan ook de doorslag om voor Luminosity te gaan.

1.5. Implementatie

Voor de implementatie is in de klasse `IntensityImageStudent` in één functie geschreven. Deze functie krijgt een `RGBImageStudent` mee en hiervan worden de RGB pixels geconverteerd naar `Intensity`. Deze functie wordt aangeroepen door zowel een constructor als een setter van `IntensityImageStudent`. Deze twee krijgen een `RGBImageStudent` mee als parameter.

```
void IntensityImageStudent::convertRGBImageStudentToIntensityImageStudent(const RGBImageStudent &input) {  
    // Values for calculating Intensity as defined in BT.709 by ITU-R  
    // See https://www.itu.int/rec/R-REC-BT.709/en for more info  
    static const double rVal = 0.2126;  
    static const double gVal = 0.7152;  
    static const double bVal = 0.0722;  
  
    int amountOfPixels = input.getWidth() * input.getHeight();  
    for (int i = 0; i < amountOfPixels; i++) {  
        RGB rgb = input.getPixel(i);  
        setPixel(i, Intensity( (rVal * rgb.r) + (gVal * rgb.g) + (bVal * rgb.b) ));  
    }  
}
```

1.6. Evaluatie

Om de implementatie te testen zullen we kijken naar de snelheid ten opzichte van de standaardmethode. Daarnaast zal ook het resultaat van de conversie vergeleken worden met het resultaat van de standaardmethode.

1.7. Bijlage: Afbeeldingen

De afbeelding waarop de verschillende methoden zijn uitgevoerd:



Hieronder zijn de afbeeldingen te vinden voor iedere methode waarmee de RGB waarden naar Intensity zijn omgezet.

