

Implementatieplan Intensity Image

Nico van Bentum, Gianluca Piccardo 27-02-2019

Doel

Het maken van een functie die een RGB afbeelding converteert naar een intensiteit/grijswaarden afbeelding.

Methoden

Er zijn een aantal manieren onderzocht om individuele pixels naar intensiteit waarden te converteren. Hierbij hebben wij drie verschillende algoritmes geëvalueerd. Het eerste 'lightness' algoritme kijkt naar de RGB channels, kiest de hoogste en de laagste en neemt daar het gemiddelde van. Het 'average' algoritme neemt alle drie RGB waarden en berekent een gemiddelde. De laatste 'Luminosity' methode is ongeveer hetzelfde als de 'average' methode maar geeft elke waarde een gewicht/modifier mee, waarbij groen het zwaarste mee telt.

Ook zijn er methodes die meer dan de aangewezen pixel bekijken, bijvoorbeeld een algoritme dat kijkt naar het een gegeneerd histogram tot aan het huidige punt en gebruikt het om een nieuwe waarde te berekenen. Ook zijn er verscheidene algoritmes te vinden die neighbours – aanliggende pixels van de geselecteerde pixel – meenemen in de berekening van een nieuwe waarde.

Keuze

Wij hebben gekozen voor de 'Luminosity' methode, deze geeft in de meeste gevallen het beste resultaat van de algoritmes die naar enkele pixels kijken. Zo geeft de 'lightness' methode vaak een 'flat' image terug en gaat er veel detail verloren. We hebben niet gekozen voor algoritmes die grotere oppervlaktes/meerder pixels bekijken omdat deze overbodig complex zijn voor de opgave en afdoen aan de snelheid.

Implementatie

De implementatie vermenigvuldigt elke channel waarde met zichzelf en vervolgens met het gewicht. Deze drie waardes worden bij elkaar opgeteld en de wortel van berekent als eindresultaat.

```
for (int i = image.getHeight()*image.getWidth(); i > -1; i--)
{
    auto pixel = image.getPixel(i);
    Intensity i_pixel = sqrt( (pow(pixel.r, 2) * .241)
                             + (pow(pixel.g, 2) * .691)
                             + (pow(pixel.b, 2) * .068) );

    intensity_img->setPixel(i, i_pixel);
}
```

Evaluatie

Onze metingen bestaan uit het testen van de nieuwe gegenereerde grayscale waarden in de GUI. Hierbij kijken we of het resultaat overeenkomt en/of beter is dan de standaard implementatie.

Daarnaast voegen we als bijlage een grayscale image van zowel de standaard implementatie als onze implementatie bij om te kijken welke scherper is.

Bronnen:

Cook, J. (2009, 24 augustus). Converting color to grayscale. Geraadpleegd op 21 februari 2019, van <https://www.johndcook.com/blog/2009/08/24/algorithms-convert-color-grayscale/>

Dobovizki, N. (2008b, 27 april). Calculating the Perceived Brightness of a Color [Blogpost]. Geraadpleegd op 27 februari 2019, van <http://www.nbdtech.com/Blog/archive/2008/04/27/Calculating-the-Perceived-Brightness-of-a-Color.aspx>