

Implementatieplan Vision

Door Stefan Dijkman & Floris Rijker

12-06-17

Doel

Het doel is om een RGB image om te zetten naar een Intensity image.

Methoden

We hebben meerdere formules gevonden waarmee we de RGB waarden naar grijswaarden kunnen converteren.

Lightness (lichtheid)	:	$\text{gray} = (\max(R, G, B) + \min(R, G, B)) / 2.$
Average (gemiddelde)	:	$\text{gray} = (R + G + B) / 3.$
Luminosity (helderheid)	:	$\text{gray} = 0.21 * R + 0.72 * G + 0.07 * B$

Keuze

Wij denken dat de Average methode de snelste manier is om een RGB image naar GrayScale om te zetten. Dit denken we omdat deze maar 3 berekeningen hoeft te doen waar Luminosity 5 berekeningen moet doen en de Lightness methode moet nog 2 andere functies aanroepen.

Implementatie

We hebben de volgende code gebruikt om het image om te zetten.

```
void IntensityImageStudent::set(const RGBImage &other) {
    IntensityImage::set(other.getWidth(), other.getHeight());
    int maxSize = other.getWidth() * other.getHeight();
    pixelStorage = new Intensity[maxSize];
    for (int i = 0; i < maxSize; i++) {
        RGB rgb = other.getPixel(i);
        pixelStorage[i] = (rgb.r + rgb.g + rgb.b) / 3;
        //pixelStorage[i] = (std::max(rgb.r, std::max(rgb.g, rgb.b)) + std::min(rgb.r, std::min(rgb.g, rgb.b))) / 2;
        //pixelStorage[i] = 0.21 * rgb.r + 0.72 * rgb.g + 0.07 * rgb.b;
    }
}
```

De andere formules staan erbij zodat deze ook getest kunnen worden om te kijken of de Average methode wel echt de snelste is. Om dit nauwkeurig te meten hebben we ook een timer geïmplementeerd.

Evaluatie

We zullen de RGB Image omzetten in een Intensity image, we zullen meten welke methode het snelste is, dat zal terug te vinden zijn in een meetrapport. Als we dit hebben gedaan zullen we kijken of de snelste methode sneller is als de default methode. Ook hiervoor zal een meetrapport komen