

1. Implementatieplan RGB naar grijswaarde

1.1. Namen en datum

Ramon van Bommel
Cris van der Nolle
21-02-2019

1.2. Doel

Het doel is om een foto met RGB waardes, om te zetten naar Grayscale / intensity.

1.3. Methoden

De eerste methode(average) is de R, G en B-waardes optellen, dan delen door 3. Hierbij krijg je een gemiddelde waarde wat de grijswaarde zou voorstellen.

De tweede methode (hierna luminosity genoemd) is RGB to YUV conversion(luma/luminosity). Hierbij worden de R, G en B-waardes niet helemaal gelijk meegenomen zoals bij de eerste methode. De rood waarden (bij de YUV-conversie) telt bijvoorbeeld minder mee dan de groen en blauwwaarden. Het doel hiervan is dat het zwart-wit resultaat beter afgesteld is voor het menselijk zicht. In onze code wordt rood voor 21% meegenomen, blauw voor 7% en groen voor 71%.

De derde methode wordt in de onderstaande bron Lightness methode genoemd. Hierbij worden de hoogste en laagste RGB-waardes bij elkaar gemiddeld genomen.

1.4. Keuze

Uit de conclusie van bovenstaande bron wordt verteld dat luma/luminosity, in t algemeen, de beste resultaten heeft. De tweede methode gaan wij dan ook als eerst implementeren. Uiteindelijk zal de inhoudelijke implementatie net iets anders zijn dan bij andere methodes. Dus de andere methodes willen we ook zeker proberen t.b.v. vergelijking.

1.5. Implementatie

De implementatie gebeurt in de eerste preprocessing stap. Hiervoor hebben we de volgende files aangepast:

- StudentPreProcessing

```

5  IntensityImage * StudentPreProcessing::stepToIntensityImage(const RGBImage &image) const {
6      IntensityImageStudent *grayScaleImage = new IntensityImageStudent(image.getWidth(), image.getHeight());
7      RGB pixel = { 0,0,0 };
8
9      for (unsigned int i = 0; i < image.getWidth(); i++) {
10         for (unsigned int j = 0; j < image.getHeight(); j++) {
11             pixel = image.getPixel(i, j);
12             if (0) { // luminosity method
13                 Intensity luminosityValue = (pixel.b * 0.07) + (pixel.g * 0.72) + (pixel.r * 0.21);
14                 grayScaleImage->setPixel(j, i, luminosityValue);
15             }
16             if (0) { // average method
17                 grayScaleImage->setPixel(j, i, (pixel.b + pixel.g + pixel.r) / 3);
18             }
19             if (1) { // lightness method takes the highest R, G, or B value and adds this with the lowest value.
20                 unsigned char maxColorValue = pixel.getBiggestValue();
21                 unsigned char minColorValue = pixel.getSmallestValue();
22                 grayScaleImage->setPixel(j, i, (maxColorValue + minColorValue) / 2);
23             }
24         }
25     }
26     return grayScaleImage;
27 }

```

De daadwerkelijke implementatie met de 3 verschillende methodes, die één grijswaardefoto teruggeeft.

- RGBImageStudent - implementatie RGB image klasse
- intensityImageStudent - implementatie Intensity Image klasse

- PixelType.h - methodes om de hoogste en laagste waarde van de RGB struct te geven.

```
21  unsigned char getSmallestValue() {
22      unsigned char smallestValue = r;
23      if (g < smallestValue) {
24          smallestValue = g;
25      }
26      if (b < smallestValue) {
27          smallestValue = b;
28      }
29      return smallestValue;
30  }
31
32  unsigned char getBiggestValue() {
33      unsigned char biggestValue = r;
34      if (g > biggestValue) {
35          biggestValue = g;
36      }
37      if (b > biggestValue) {
38          biggestValue = b;
39      }
40      return biggestValue;
41  }
42  };
```

1.6. Evaluatie

De resultaten zullen meerdere grijswaarde foto's zijn. Voor elke methode één. Deze zullen wij vergelijken en We willen meerdere foto's per test meenemen, waarin verschillende methodes worden gebruikt om te testen of alle kleuren realistisch omgezet kunnen worden.

1.7. Bronnen

Cook, J. O. H. N. (2009, 24 augustus). Converting color to grayscale. Geraadpleegd op 14 april 2019, van <https://www.johndcook.com/blog/2009/08/24/algorithms-convert-color-grayscale/>

Wikipedia contributors. (2019, 24 maart). color space typically used as part of a color image pipeline. Geraadpleegd op 14 april 2019, van <https://en.wikipedia.org/wiki/YUV>

Cambridge in Colour. (z.d.). Understanding Digital Camera Histograms: Luminosity and Color. Geraadpleegd op 14 april 2019, van <https://www.cambridgeincolour.com/tutorials/histograms2.htm>