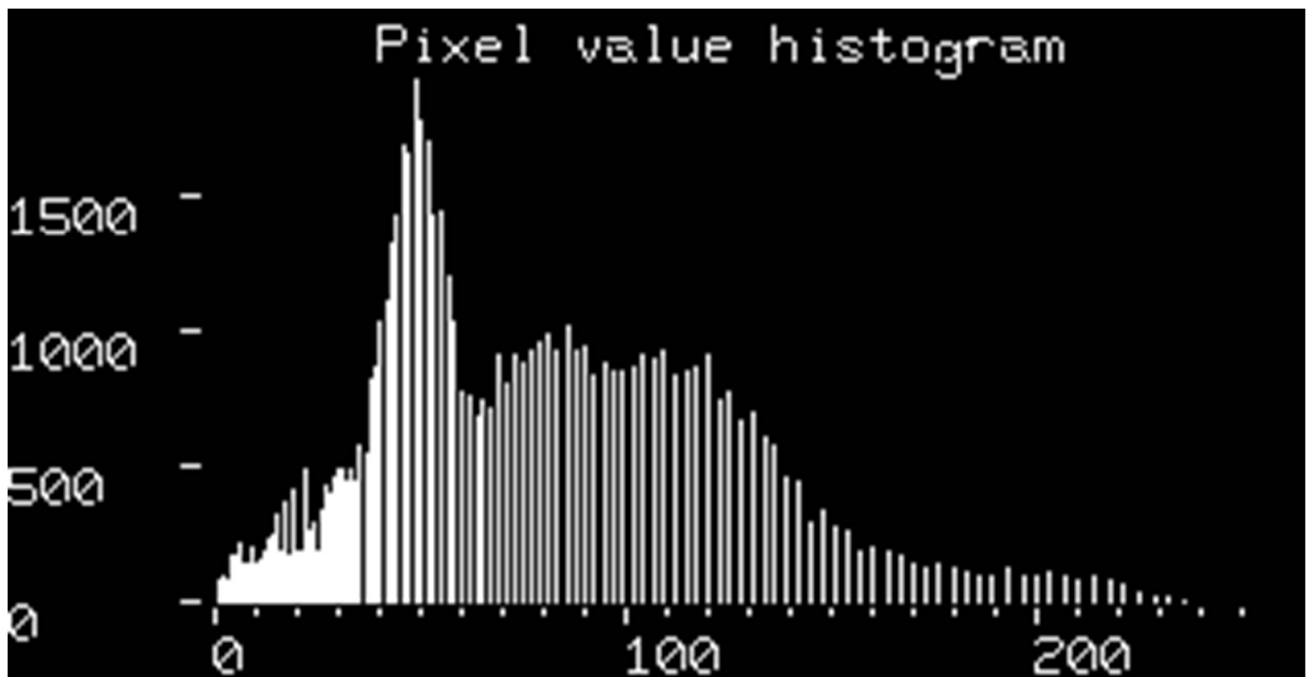


Implementatieplan onderzoek RGBtoIntensity algoritmes



Jeremy Ruizenaar 20-06-17

Doel onderzoek

Een RGBimage is een kleurenfoto bestaande uit drie kleurkanalen respectievelijk rood, groen en blauw. Voor een pc zijn deze drie kleurkanalen eigenlijk overbodige informatie. Het is namelijk makkelijker en sneller om bruikbare data uit een grijsintens foto te halen. Een grijsintensfoto oftewel een intensityimage bevat 255 mogelijke opties van grijsintens, dit omdat een pixel een waarde ter grote van een byte heeft. Het is dus meestal handig om een RGBimage om te zetten naar een intensityimage. Om dit te doen moet er dus een algoritme zijn wat op basis van een RGB pixel een intensity pixel kan maken.

Het doel van dit onderzoek is uitzoeken welke algoritmes er beschikbaar zijn om een RGBimage om te zetten naar een intensity image. Hierbij moet rekening gehouden worden met de snelheid, kwaliteit en bruikbaarheid van de afbeelding.

Bruikbare algoritmes

Er zijn verschillende bruikbare algoritmes beschikbaar voor de conversie van een RGBimage naar een intensityimage:

1. Averaging

Bij averaging wordt er per pixel een linear gemiddelde genomen tussen het rode groene en blauwe kleurkanaal. Dit wordt gedaan door de waardes van de drie kanalen op tellen en te delen door drie.

2. Luminance correction

Bij luminance correction worden per pixel de drie kleur kanalen gewogen naar gevoeligheid voor het menselijk zicht. De coëfficiënten waarmee de kleur kanalen gewogen worden zijn $0.3 \cdot R + 0.59 \cdot g + 0.11 \cdot b$, deze formule heeft als uitkomst die intensiteit. De reden waarom de kleurkanalen andere coëfficiënten gekozen worden is omdat het menselijk oog meer gevoeliger is voor de kleur groen. Om de juiste kleur te weergeven moet het groen dus zwaarder gewogen worden.

3. Desaturation

Bij desaturation wordt per pixel een RGB waarde omgezet in een HSL waarde. De hue wordt gezet op een willekeurige kleur en de saturation wordt op nul gezet. De lightness kan berekend worden door het volgende algoritme $lightness = (\max(R, G, B) + \min(R, G, B)) / 2$. Oftewel het gemiddelde tussen de maximum en minimum waarde van het RGB paar

4. Decomposition

Bij decomposition kan er onderscheid gemaakt worden tussen maximum en minimum decomposition. Bij maximum decomposition wordt per pixel de intensity gelijk aan $\max(R, G, B)$, en bij minimum decomposition wordt de intensity gelijk aan $\min(R, G, B)$. Dit kan of hele donkere of hele lichte afbeeldingen opleveren.

5. Single color channel

Bij single color channel wordt de intensiteit per pixel gelijk aan de waarde van één van de drie kleur kanalen. Dit kan dus het rode, groene, of blauwe kanaal zijn. In de praktijk word vaak van het groene kanaal uitgegaan. Dit kan of hele donkere of hele lichte afbeeldingen opleveren afhankelijk van het gekozen kanaal om de waardes uit te kopiëren.

.

Keuze

Er is gekozen voor het single-color-channel algoritme. Dit algoritme zou het snelst moeten zijn omdat alleen de instructie ($\text{gray} = \text{pixel.g}$) uitgevoerd wordt. Dit is eigenlijk gewoon een variabele kopiëren zonder enige rekenkundige operaties en kan dus zeer snel uitgevoerd worden door een cpu.

ook is er gekozen voor het average algoritme omdat dit ook een vrij simpel algoritme is. hierbij wordt een lineair gemiddelde genomen tussen de RGB waarden. Dit algoritme bevat wat rekenkundige operaties, maar dit is een eenvoudige optelling van drie cijfers met vervolgens een deling.

Verder komt het luminance-correction algoritme aan de orde. Dit is een algoritme wat ook snel zou moeten werken maar voor een cpu iets complexer is. Dit vanwege meerdere floating-point operations per instructie. Dit kost dus meer tijd bij het berekenen.

Deze drie algoritmes zouden elk voldoende moeten zijn om een kwalitatief goede intensityimage te verkrijgen uit een RGBimage. De voornaamste reden dat het luminance-correction-algoritme en het average-algoritme gekozen zijn is dat deze beiden met schaling dan wel gemiddeldes werken, dit zou een beter kwalitatieve afbeelding moeten opleveren. Het single-color-channel algoritme is gekozen met de groene pixel omdat het menselijk oog daar het meest gevoelig voor is. Ook de snelheid waarmee dit algoritme uitgevoerd kan worden is door het zeer beperkte aantal stappen een belangrijke factor geweest.

Implementatie

Single color channel algoritme

```
for ( int x = 0; x < image.getWidth(); x++ ) {
    for (int y = 0; y < image.getHeight(); y++) {
        RGB pixel = image.getPixel(x, y);
        int grayValue = pixel.g;
        IntensityImg->setPixel( x*image.getHeight()+y, Intensity(grayValue)
    );
    }
}
```

Luminance correction algoritme

```
for ( int x = 0; x < image.getWidth(); x++ ) {
    for ( int y = 0; y < image.getHeight(); y++ ) {
        RGB pixel = image.getPixel( x, y );
        int grayValue = ( 0.3*pixel.r + 0.59*pixel.g + 0.11*pixel.b );
        IntensityImg->setPixel( x*image.getHeight() +y,
Intensity(grayValue) );
    }
}
```

Average algoritme

```
for ( int x = 0; x < image.getWidth(); x++ ) {
    for ( int y = 0; y < image.getHeight(); y++ ) {
        RGB pixel = image.getPixel( x, y );
        int grayValue = ( pixel.r + pixel.g + pixel.b ) / 3 );
        IntensityImg->setPixel( x*image.getHeight() +y,
Intensity(grayValue) );
    }
}
```

Evaluatie

De tijd die het algoritme in beslag neemt om een image om te zetten zal gemeten worden over 100000 iteraties. Deze tijd zal gemeten worden met behulp van een extra timer-klasse. De uitkomst van deze metingen zal in een tabel geplaatst worden. Vervolgens zal er aan de hand van de data in de tabel een conclusie getrokken kunnen worden welk algoritme het snelste uitgevoerd wordt.

Verder wordt de kwaliteit van de image vergeleken met behulp van histogrammen en het menselijk oog. Hiervoor zullen er twee afbeeldingen genomen worden die elk door de algoritmes omgezet zullen worden. Deze afbeeldingen zullen naast elkaar gezet worden en samen met hun bijbehorende histogram vergelekt worden. Door analyse van de histogrammen kan er een conclusie getrokken welke afbeelding het meeste contrast heeft en dus kwalitatief het meest hoogwaardig is.

Ook zal er gekeken worden of de afbeelding bruikbaar is in vervolg stappen van het face-recognition process. Dit wordt gedaan door elke afbeelding om te zetten met een van de algoritmes en vervolgens de rest van het process door te lopen. De uitkomst van deze test zal uitgezet worden in een tabel. Op basis van deze tabel kan aangetoond worden welke algoritmes bruikbaar zijn voor een face-recognition systeem.