



Odisee Hogeschool  
Gebroeders de Smetstraat 1, 9000 Gent

# **Applied Programming Project 1:**

## Histogram Stretching

Steven Impens



# Inhoudsopgave

Codefragmentenlijst.....	3
Figurenlijst .....	4
Inleiding .....	6
1 Onderzoek .....	7
1.1 Kleuren Modellen .....	7
1.1.1 RGB .....	7
1.1.2 CMYK.....	7
1.1.3 RGB converteren naar CMYK.....	8
1.2 Histogram.....	9
1.2.1 Kleuren Histogram.....	9
1.2.2 Histogram Stretching .....	10
2 Uitwerking .....	12
2.1 Grafische User Interface .....	12
2.2 Afbeelding naar histogram data .....	13
2.3 Implementatie Stretching .....	15
2.4 Implementatie RGB naar CMYK .....	16
3 Besluit .....	17
4 Perfectioneren van het programma .....	18
Literatuurlijst.....	19

## Codefragmentenlijst

Codefragment 2-1: Itereren over pixels van een afbeelding .....	13
Codefragment 2-2: Toevoegen kleurwaardes aan dictionary .....	13
Codefragment 2-3: Som van het aantal pixels per helderheidswaarde berekenen.....	14
Codefragment 2-4: Itereren over elke pixel en nieuwe waardes toewijzen .....	15
Codefragment 2-5: Converteren van RGB naar CMYK.....	16

# Figurenlijst

Figuur 1-1: RGB Model.....	7
Figuur 1-2: CMYK Model.....	7
Figuur 1-3: Histogram.....	9
Figuur 1-4: Stretching histogram.....	10
Figuur 2-1: Interface .....	12

# Inleiding

Het doel van dit project is een programma ontwerpen die histogram stretching toepast op kleuren afbeeldingen. In dit verslag zal duidelijk worden wat een histogram is, hoe deze kan worden gestretcht en wat de impact is op de originele foto. Er wordt rekening gehouden met 2 kleuren modellen waaronder RGB en CMYK voor het maken van de histogrammen en het stretchen wordt enkel met RGB-waarden uitgevoerd. Voor het schrijven van het programma werd gebruik gemaakt van Visual Studio 2017 en C# met als achtergrond doel een zo gebruiksvriendelijke interface te ontwerpen zodat men gemakkelijk kan experimenteren met verschillende componenten van de kleuren modellen en waarden van de histogrammen.

# 1 Onderzoek

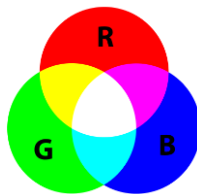
## 1.1 Kleuren Modellen

### 1.1.1 RGB

RGB staat voor Rood-Groen-Blauw en is een model van het additieve kleurmenging type. Een kleur ontstaat door de 3 primaire kleuren met elkaar te mengen zoals in **Figuur 1-1**.

Elke Primaire kleur wordt meestal uitgedrukt in een getal van 8 bits. Zo heeft deze een bereik van 0 tot en met 255, dit is de intensiteit van de kleur. Het is tevens ook mogelijk om RGB voor te stellen met meer bits waaronder 12, 16 of meer. Dit wordt vaak gebruikt waar hogere kwaliteit nodig is.

Een samengemengde kleur wordt uitgedrukt in 3 waarden van 0 tot en met 255 of meer naargelang de gekozen kwaliteit.

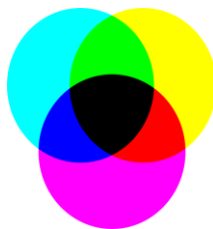


*Figuur 1-1: RGB Model*

### 1.1.2 CMYK

CMYK staat voor Cyan-Magenta-Yellow-Key en is een model van het subtractieve kleurmenging type. Het verschil tussen RGB en CMYK is dat bij dit model een kleur ontstaat door de primaire kleuren van elkaar af te trekken. Beginnend bij wit en door een of meer van de primaire kleuren weg te halen zal een kleur ontstaan zoals in **Figuur1-2**. In de praktijk gebruikt met nog een vierde component, de Key meer bekend als Zwart.

Een samengemengde kleur wordt uitgedrukt in 4 percentages.



*Figuur 1-2: CMYK Model*

### 1.1.3 RGB converteren naar CMYK

Om een RGB-kleur te converteren naar een CMYK-kleur zijn er verschillende formules nodig.

$$R' = \frac{R}{255} \quad G' = \frac{G}{255} \quad B' = \frac{B}{255}$$

Waarbij R, G en B respectievelijk de kleuren componenten Rood, Groen en Blauw zijn van de initiële RGB-kleur.

Voor de primaire kleuren van het CMYK-model te bereken is de Key waarde nodig.

$$K = 1 - \max(R', G', B')$$

Nu dat het Key component bekend is kunnen de andere componenten berekend worden.

$$C = \frac{(1 - R' - K)}{(1 - K)}$$

$$M = \frac{(1 - G' - K)}{(1 - K)}$$

$$Y = \frac{(1 - B' - K)}{(1 - K)}$$



## 1.2 Histogram

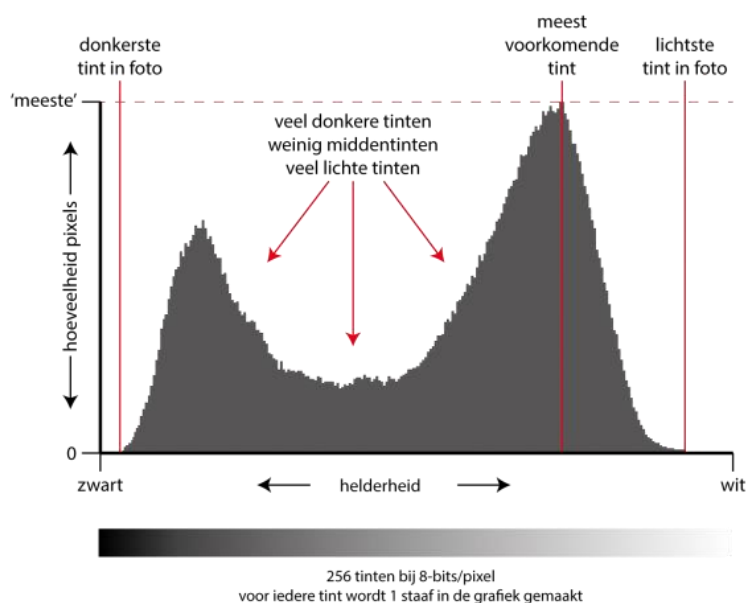
### 1.2.1 Kleuren Histogram

Een Kleuren histogram is een grafiek waarbij voor elke helderheidswaarde het aantal keer dezelfde waarde weergegeven wordt.

#### **Figuur 1-3:**

De x-as van het histogram bevat de helderheidswaarde.

De y-as bevat het aantal pixels. 1 staaf op de grafiek zijn het aantal pixels die dezelfde helderheidswaarde hebben.



*Figuur 1-3: Histogram*

Het RGB-kleuren model zal in totaal 3 verschillende histogrammen bevatten, een voor elke primaire kleur. Voor Rood, Groen en Blauw. Telkens van 0 tot en met 255.

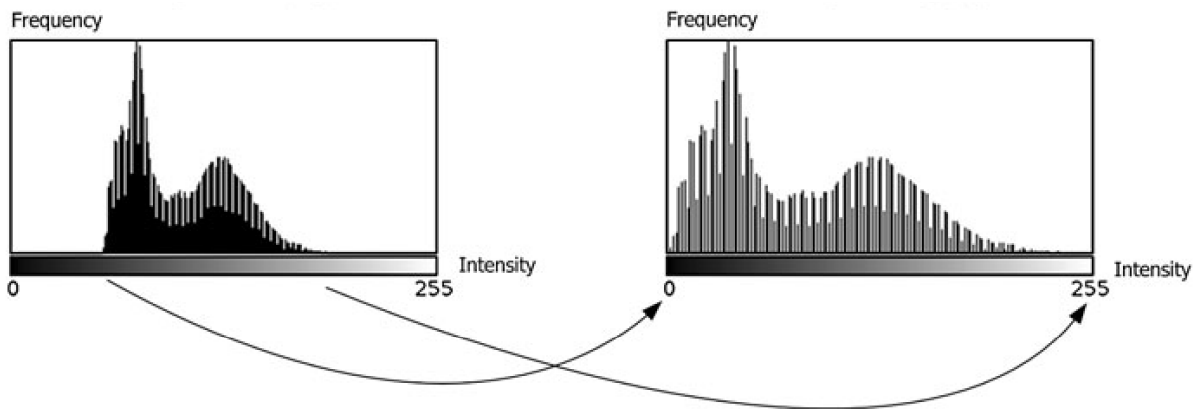
Hetzelfde kan met het CMYK-kleuren model, waarbij de x-as een bereik heeft van 0% tot en met 100%.

Net zoals bij RGB zal elk component een eigen histogram verkrijgen.

## 1.2.2 Histogram Stretching

Histogram Stretching is een techniek waarbij het bereik van de helderheid gestretcht wordt. Dit is vooral nuttig bij histogrammen die niet het hele bereik innemen.

Voorbeeld op **Figuur 1-4** waar een histogram met een helderheidsbereik van 65 tot en met 195 zal gestretcht worden naar een bereik van 0 tot en met 255. Het volledig bereik van een RGB-kleuren component.



*Figuur 1-4: Stretching histogram*

Doordat het histogram wordt gespreid, zal er een groter contrast ontstaan tussen de helderheden in elk kleuren component.

Een grote drawback van deze techniek is dat wanneer er zich pixels bevinden aan de uiterste waarden van de helderheid dan zal het histogram niet verder gestretcht worden. Dit kan verholpen worden door grenzen te definiëren, deze grenzen zullen dan de minimum en maximum-helderheidswaarde zijn die worden gestretcht naar respectievelijk 0 en 255. Een grens kiezen kan op het aantal pixels dat eronder of erboven moet liggen. Bijvoorbeeld 5% en 95% voor de onderste en bovenste grens dan zullen 5% van de pixels onder de onderste grens liggen en 5% van de pixels boven de bovenste grens. Hierdoor worden “outliers” genegeerd bij het stretchen, deze pixels zullen dan toegevoegd worden aan de onderste en bovenste grens van de gestretchte histogram.

Histogram stretching wordt toegepast aan de hand van volgende formule waarbij “ $P_{out}$ ” en “ $P_{in}$ ” de nieuwe en originele kleur zijn. “ $a$ ” en “ $b$ ” het gewenst bereik is ( $a=0$ ,  $b=255$ ). “ $d$ ” en “ $c$ ” het oude bereik is dat ook kan afhangen van eventuele outliers.

$$P_{out} = (P_{in} - c) * \left(\frac{b - a}{d - c}\right) + a$$

Berekende waarde kleiner dan 0 worden 0 en waarde groter dan 255 worden 255.

## 2 Uitwerking

### 2.1 Grafische User Interface

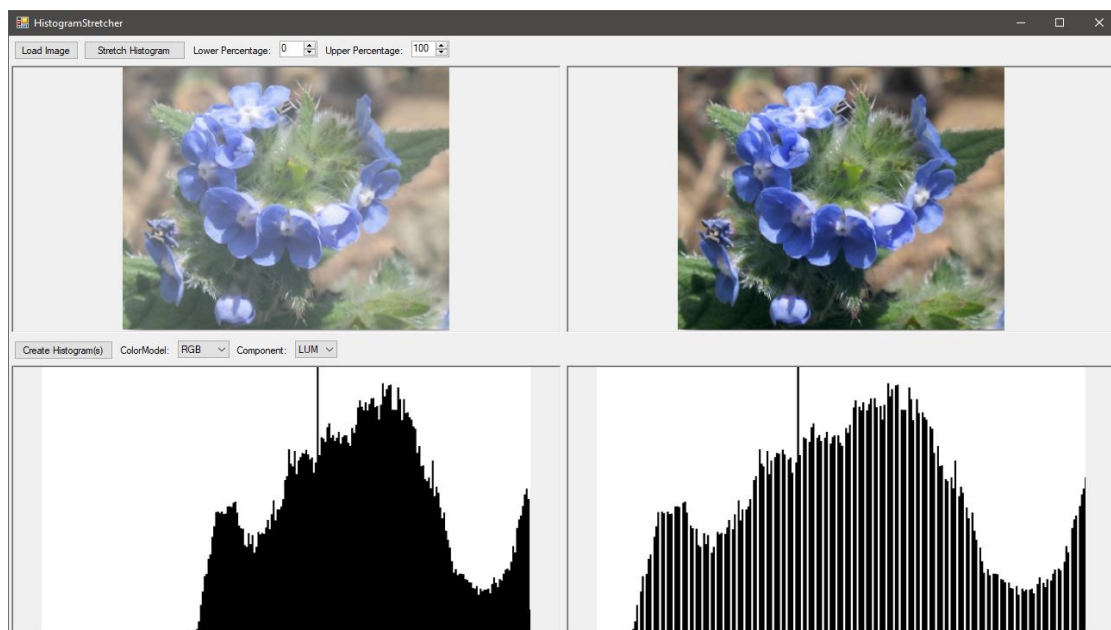
De Grafische User Interface afgekort GUI bevat 2 delen zoals te zien in **Figuur 2-1**.

Het eerste deel heeft als doel om een afbeelding in te laden en weer te geven. Daarnaast is er plaats voor de afbeelding waarvan het histogram is gestretcht. Dit deel bevat alle nodige opties om een afbeelding te kunnen stretchen, waaronder een knop om de afbeelding in te laden, een knop voor diezelfde afbeelding zijn histogram te stretchen en opties om eventuele outliers weg te werken bij het stretchen, de lower percentage en de upper percentage.

Het tweede deel is voorbehouden voor de histogrammen. Een histogram representeert de afbeelding erboven. Hier zijn 2 opties beschikbaar waaronder het ColorModel waarvan het histogram gebruik van maakt en het Component van het geselecteerde kleurenmodel die het histogram zal weergeven.

Mogelijke Kleuren Modellen zijn:

- RGB met componenten: Helderheid (LUM), Gemiddelde (AVG), Rood (R), Groen (G), Blauw (B).
- CMYK met componenten: Gemiddelde (AVG), Cyaan (C), Magenta (M), Yellow (Y), Key (K).



*Figuur 2-1: Interface*

## 2.2 Afbeelding naar histogram data

De code zal over elke pixel van de afbeelding gaan en het aantal keer de waarde van een kleur component voorkomt bijhouden. De componenten zijn gedefinieerd als “sleutels” in een variabele van het type dictionary.

```
for (int i = 0; i < OriginalImage.Width; i++)
{
    for (int j = 0; j < OriginalImage.Height; j++)
    {
        Color color = this.OriginalImage.GetPixel(i, j);

        AddRgbColor(color);
        AddCmykColor(color);
    }
}
```

*Codefragment 2-1: Itereren over pixels van een afbeelding*

Voorbeeld component R van RGB: In de dictionary bij de sleutel “R” zal een array met 256 waarden staan. De index van die array is de helderheidswaarde gaande van 0 tot en met 255 en hebben als waarde de hoeveelheid pixels die de index als waarde hebben.

Voorbeeld component C van CMYK: In de dictionary bij de sleutel “C” zal een array met 1001 waarden staan. De index van die array is de helderheidswaarde gaande van 0 tot en met 1000 en hebben als waarde de hoeveelheid pixels die de index als waarde maal 1000 hebben.

Tevens is er ook nog de Luminosity sleutel bij het RGB-model, Die waarden worden berekend met volgende Formule:  $0.3 * R + 0.59 * G + 0.11 * B$ .

```
private void AddRgbColor(Color color)
{
    this.ValueCollectionRGB["R"][color.R]++;
    this.ValueCollectionRGB["G"][color.G]++;
    this.ValueCollectionRGB["B"][color.B]++;
    this.ValueCollectionRGB["LUM"]
        [(int)(0.3 * color.R + 0.59 * color.G + 0.11 * color.B)]++;
}
```

*Codefragment 2-2: Toevoegen kleurwaarden aan dictionary*

Voor een histogram waar alle kleurcomponenten in meegerekend zijn is er een aparte sleutel in de dictionary. Genoemd naar zijn kleurenmodel. Deze omvat de som van het aantal pixel per helderheidswaarde.

```
for (int i = 0; i <= 255; i++)
{
    this.ValueCollectionRGB["RGB"][i] =
        this.ValueCollectionRGB["R"][i] +
        this.ValueCollectionRGB["G"][i] +
        this.ValueCollectionRGB["B"][i];
}
```

*Codefragment 2-3: Som van het aantal pixels per helderheidswaarde berekenen*

## 2.3 Implementatie Stretching

Voor het stretchen wordt gebruik gemaakt van de formules in “1.2.2 Histogram Stretching”. De code zal eerst een kopie van de originele afbeelding maken en een minimum en maximum-waarde zoeken voor “c” en “d” voor de formule respectievelijk “min” en “max” waarna over elke pixel van de afbeelding zal worden geïtereerd. Tijdens dit proces wordt de kleur van de pixels vervangen met de nieuwe waarden berekend door de formules.

```
for (int i = 0; i < stretched.Width; i++)
{
    for (int j = 0; j < stretched.Height; j++)
    {
        Color color = stretched.GetPixel(i, j);

        //formulas: https://homepages.inf.ed.ac.uk/rbf/HIPR2/stretch.htm
        int r = (int)((color.R - min) * (255 / (float)(max - min)));
        int g = (int)((color.G - min) * (255 / (float)(max - min)));
        int b = (int)((color.B - min) * (255 / (float)(max - min)));

        if (r > 255) r = 255;
        if (r < 0) r = 0;
        if (g > 255) g = 255;
        if (g < 0) g = 0;
        if (b > 255) b = 255;
        if (b < 0) b = 0;

        Color colorNew = Color.FromArgb(r, g, b);
        stretched.SetPixel(i, j, colorNew);
    }
}
```

*Codefragment 2-4: Itereren over elke pixel en nieuwe waardes toewijzen*

## 2.4 Implementatie RGB naar CMYK

Voor het converteren van RGB naar CMYK worden de formules van “1.1.3 RGB converteren naar CMYK” gebruikt. De waarden worden afgerond en vermenigvuldigd met 1000 zodat deze net zoals bij RGB in een array kunnen worden opgeteld.

```
private int[] ConvertRgbToCmyk(Color color)
{
    //formulas: http://www.rapidtables.com/convert/color/rgb-to-cmyk.htm
    double red = color.R / 255.0;
    double green = color.G / 255.0;
    double blue = color.B / 255.0;

    double k = 1 - Math.Max(red, Math.Max(green, blue));
    double c = (1 - red - k) / (1 - k);
    double m = (1 - green - k) / (1 - k);
    double y = (1 - blue - k) / (1 - k);

    int key = (int)(Math.Round(k, 3) * 1000);
    int cyan = (int)(Math.Round(c, 3) * 1000);
    int magenta = (int)(Math.Round(m, 3) * 1000);
    int yellow = (int)(Math.Round(y, 3) * 1000);
    return new int[] { cyan, magenta, yellow, key};
}
```

*Codefragment 2-5: Converteren van RGB naar CMYK*



### 3 Besluit

Tijdens het onderzoeken van verschillende onderwerpen over histogrammen is het duidelijk geworden dat er geen eenheid is voor het maken van een histogram dat data bevat van meerdere kleurencomponenten. Zo kan men een histogram van meerdere kleuren componenten genereren door deze met elkaar op te tellen, enkel de hoogste waarde te gebruiken of het gemiddelde van de componenten te gebruiken. Het stretchen van een histogram maakt een verschil in het contrast van de afbeelding, zo wordt de afbeelding kleurrijker en minder dof maar outliers hebben een groot impact op het stretchen en moet er hiervoor gecompenseerd worden. De pixels behouden hun originele kleur maar hebben tegenover elkaar een groter contrast. Op de histogrammen is duidelijk te zien dat de data verder uit elkaar ligt bij het stretchen.

## 4 Perfectioneren van het programma

Er is nog veel mogelijkheid in het uitbreiden van het programma.

De histogrammen worden getoond zonder assen. Het bijvoegen van assen zouden duidelijker maken wat er juist veranderd aan de histogrammen.

Bij grote afbeeldingen moet de code veel werk verrichtten, het zou beter zijn als de calculaties gebeuren in de achtergrond waardoor het programma nog bruikbaar is voor eventueel opties te veranderen.

Extra opties voor het tweaken van de histogrammen en het stretchen voor nauwkeurigheid en dat er meer geëxperimenteerd kan worden.

## Literatuurlijst

- [1] "RGB color model," Wikipedia, 8 November 2017. [Online]. Available: [https://en.wikipedia.org/wiki/RGB\\_color\\_model](https://en.wikipedia.org/wiki/RGB_color_model).
- [2] "Image Histogram," Wikipedia, 21 March 2017. [Online]. Available: [https://en.wikipedia.org/wiki/Image\\_histogram](https://en.wikipedia.org/wiki/Image_histogram).
- [3] "RGB to CMYK color conversion," RapidTables, 2017. [Online]. Available: <http://www.rapidtables.com/convert/color/rgb-to-cmyk.htm>.
- [4] "CMYK," Wikipedia, 17 August 2017. [Online]. Available: <https://nl.wikipedia.org/wiki/CMYK>.
- [5] "Hoe gebruik je een histogram?," EMDAY, [Online]. Available: <http://emday.nl/2013/03/hoe-lees-je-een-histogram/>.
- [6] "Histogram Dialog," GIMP, [Online]. Available: <https://docs.gimp.org/en/gimp-histogram-dialog.html>.
- [7] "Histogram," emday, [Online]. Available: [http://emday.nl/files/2013/03/histogram\\_B\\_1368px-560x460.png](http://emday.nl/files/2013/03/histogram_B_1368px-560x460.png).
- [8] WhatWhenHow, [Online]. Available: <http://what-when-how.com/wp-content/uploads/2012/07/tmp26dc99.png>.
- [9] R. Fisher, S. Perkins, A. Walker and E. Wolfart, "Contrast Stretching," HIPR2, 2003. [Online]. Available: <https://homepages.inf.ed.ac.uk/rbf/HIPR2/stretch.htm>.
- [10] G. Bakker, "Histograms," geraldbakker, 29 June 2017. [Online]. Available: <http://geraldbakker.nl/psnumbers/histograms-1.html>.