

# DSP-implementatie - Imageprocessing in MATLAB

Andries Gert-Jan  
Dejager Xavier  
Steen Nick

Master of Science in de industriële  
ingenieurswetenschappen:  
elektronica-ICT, optie Elektronica

**Docent:**  
ing. Dries Debouvere

© Copyright KU Leuven

Zonder voorafgaande schriftelijke toestemming van zowel de docent als de auteurs is overnemen, kopiëren, gebruiken of realiseren van deze uitgave of gedeelten ervan verboden. Voor aanvragen tot of informatie i.v.m. het overnemen en/of gebruik en/of realisatie van gedeelten uit deze publicatie, wend u tot IIW, Zeedijk 101, B-8400 Oostende, +32-59-569000 of via e-mail [iiw.kulab.oostende@kuleuven.be](mailto:iiw.kulab.oostende@kuleuven.be).

Voorafgaande schriftelijke toestemming van de docent is eveneens vereist voor het aanwenden van de in dit verslag beschreven (originele) methoden, producten, schakelingen en programma's voor industrieel of commercieel nut en voor de inzending van deze publicatie ter deelname aan wetenschappelijke prijzen of wedstrijden.

# Inhoudsopgave

<b>Lijst van figuren en tabellen</b>	<b>ii</b>
<b>Lijst van gebruikte afkortingen</b>	<b>iii</b>
<b>1 Inleiding</b>	<b>1</b>
1.1 Doelstelling . . . . .	1
1.2 Achtergrond . . . . .	1
<b>2 Resultaten</b>	<b>3</b>
2.1 Opgave 1 . . . . .	3
2.2 Opgave 2 . . . . .	7
<b>3 Algemeen besluit</b>	<b>12</b>
<b>A Matlab Code</b>	<b>14</b>
A.1 Opgave 1 Deel 1 . . . . .	14
A.2 Opgave 1 Deel 2 . . . . .	15
A.3 Opgave 2 Deel 1 . . . . .	16
A.4 Opgave 2 Deel 2 . . . . .	17

# Lijst van figuren en tabellen

## Lijst van figuren

2.1	Resultaat kleursverandering Smurf: eerste aanpak . . . . .	5
2.2	Resultaat kleursverandering Smurf: tweede aanpak . . . . .	7
2.3	Resultaat achtergrond veranderen . . . . .	9
2.4	Resultaat achtergrond veranderen - uitbreiding . . . . .	11

## Lijst van tabellen

2.1	Grenswaarden voor blauwe kleur . . . . .	3
2.2	Offset voor conversie blauwe kleur naar gele kleur . . . . .	5
2.3	Grenswaarden voor blauwe kleur . . . . .	8

# Lijst van gebruikte afkortingen

## Afkortingen

DSP	Digital Signal Processing
tv	Televisie
rgb	Rood Groen Blauw

# Hoofdstuk 1

## Inleiding

### 1.1 Doelstelling

Het doel van deze opdracht is om via het Matlab pakket kennis te maken met de basisprincipes van videoprocessing. Deze kennismaking is de basis van de vervolgoopdracht waarbij gebruikt gemaakt wordt van een DSPprocessor om live videobeelden realtime te bewerken. Een videosignaal bestaat uit verschillende frames. Wanneer men dus een videosignaal wil bewerken dient men dit frame per frame aan te pakken. Elk frame kan gezien worden als een afbeelding waarop één of meerdere bewerkingen gebeuren. Aan de hand van het Matlab pakket wordt kennis verworven in DSP-technieken om kleuren in afbeeldingen te detecteren en veranderen.

In het eerste deel van deze opdracht is het de bedoeling om aan de hand van het programma Matlab inzicht te verwerven in het bewerken van afbeeldingen (frames). Nadien zal in het tweede deel hetzelfde principe realtime toegepast worden op videobeelden aan de hand van een DSP-processor, namelijk de BlackFin BF-561.

### 1.2 Achtergrond

#### 1.2.1 Chromakey

Chromakey, ook wel kleurwaarde genoemd is een techniek die véél gebruikt wordt in de tv- en filmwereld. Tijdens het filmen wordt als achtergrond een scherm gebruikt dat een bepaalde effen kleur heeft. Vervolgens wordt via een computer of DSP-processor deze achtergrond veranderd door een andere achtergrond die afkomstig kan zijn van een andere videobron of afbeelding. De meest gebruikte kleuren voor deze achtergrond zijn blauw en groen. De reden hiervoor is dat de kleuren blauw en groen het meest verschillen in tint (hue) van de kleur van de menselijke huid.

De chromakey techniek wordt al gebruikt sinds de jaren '90 om speciale effecten in films mogelijk te maken. Ook bij het weerbericht en het journaal maakt men vaak

gebruik van chromakey om extra informatie achter de presentator weer te geven. Het is zowel mogelijk om live te achtergrond te wijzigen als nadien.

De chromakey techniek brengt ook een aantal problemen met zich mee. Zo speelt de belichting van de set een belangrijke rol tijdens het filmen. Het vormen van schaduw op het scherm kan ervoor zorgen dat dit van kleur veranderd, namelijk het wordt donkerder waardoor het door het bewerkingsprogramma niet meer herkend wordt als de kleur die dient gewijzigd te worden. Een ander probleem dat zich voordoet is dat te dunne objecten (bewegende haren) moeilijker kunnen worden waargenomen door de computer. Dit kan zorgen voor een ongewenst effect. Tot slot dient er ook op gelet te worden dat de presentator zijn kleding geen element bevat in de kleur van het scherm dat gebruikt wordt. Anders zou dit ook vervangen worden, net als de achtergrond.

Naast het gebruik chromakey in de achtergrond is het ook mogelijk om in een videofragment een bepaalde kleur te vervangen door een andere kleur. Deze techniek kan toegepast worden om oude zwart-wit films om te vormen naar een minimalistische kleurenfilm. Dit kan gedaan worden door aan een bepaalde grijswaarde een bepaalde kleur toe te kennen. Hierbij speelt de belichting en schaduwen vaak een nadelige rol.

### 1.2.2 Matlab

Matlab is een computerprogramma dat het mogelijk maakt om allerhande wiskundige toepassingen te simuleren. Aan de basis hiervan ligt de programmeertaal M-code. Via Matlab is het mogelijk om afbeeldingen in te lezen en vervolgens hierop bewerkingen uit te voeren. Vervolgens kan de bewerkte afbeelding terug worden weergegeven.

## Hoofdstuk 2

# Resultaten

### 2.1 Opgave 1

#### 2.1.1 Doelstelling

Deze opgave bestaat eruit om de afbeelding van een Smurf zodanig te bewerken dat de blauwe kleur wordt veranderd in een gele kleur. Er dient opgemerkt te worden dat de blauwe kleur waaruit de Smurf bestaat niet één type blauw is, maar bestaat uit verschillende tinten. Om het blauw naar het geel om te vormen moet dus rekening gehouden worden met een bepaald bereik wat de kleur betreft.

#### 2.1.2 Werkwijze en resultaat

##### Eerste aanpak

Bij de eerste aanpak werd gekozen om alle kleuren die tussen bepaalde grenswaarden vallen te vervangen door een gele kleur. De grenswaarden werden vastgesteld door middel van het programma ColorPic. Dit programma maakt het mogelijk om aan de hand van een colorpicker de kleur van een bepaalde pixel op het scherm te bepalen. Proefondervindelijk werden zo volgende grenzen bepaald:

Kleur	Ondergrens	Bovengrens
Rood	0	115
Groen	100	200
Blauw	0	255

TABEL 2.1: Grenswaarden voor blauwe kleur



Om de blauwe kleur naar geel te veranderen werd elke pixel van de afbeelding doorlopen door middel van een for-structuur. Elke pixel wordt gecontroleerd of de kleur tussen de grenswaarden valt. Indien dit het geval is wordt de kleur veranderd naar geel. Om de kleur van elke pixel te controleren moest afzonderlijk naar de rood-waarde, groen-waarde en blauw-waarde van elke pixel gekeken worden. Deze waarden liggen elk tussen een waarde van 0 en 255. Als gele kleur werd gekozen voor het geel dat overeenstemt met rgb-waarde 255,255,0. De Matlabimplementatie is weergegeven in onderstaand codefragment:

```
DATA = imread('C:\Users\Gert-Jan\Desktop\DSPmatlab\GroteSmurf.jpg');

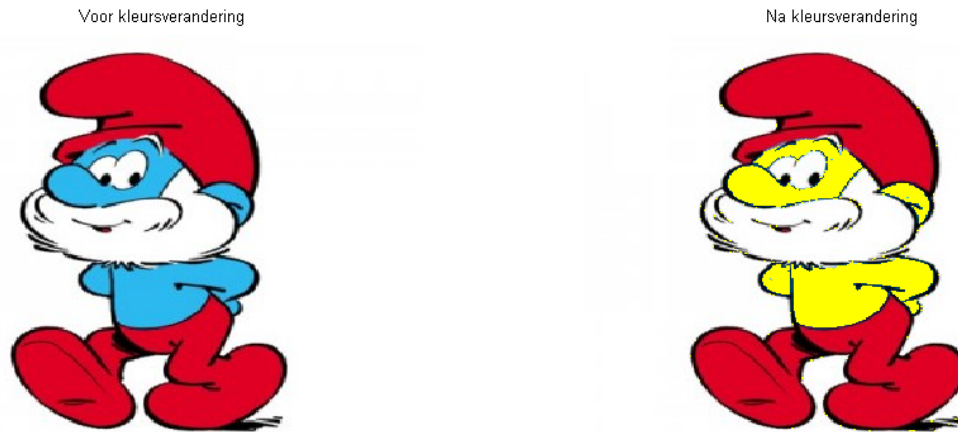
figure
subplot(1,2,1);
image(DATA)
title('Voor kleursverandering')
axis off;

for y=1:300
    for x=1:300
        if DATA(y,x,1)>0 && DATA(y,x,1)<115 && DATA(y,x,2)>100 &&
DATA(y,x,2)<200 && DATA(y,x,3)>0 && DATA(y,x,3)<255
            DATA(y,x,1) = 255;
            DATA(y,x,2) = 255;
            DATA(y,x,3) = 0;
        end
    end
end

subplot(1,2,2);
image(DATA)
title('Na kleursverandering')
axis off;
```

LISTING 2.1: Matlab-code eerste aanpak

Het resultaat van deze code is te zien in onderstaande figuur:



FIGUUR 2.1: Resultaat kleursverandering Smurf: eerste aanpak

Er is op te merken dat dit een vrij drastische verandering is. Het blauw bestaat namelijk niet uit één type blauw, maar uit een mix van verschillende tinten blauw en het wordt vervangen door slechts één type geel. Dit vormt geen aangenaam gevoel voor het menselijk oog. Men neemt dit waar als een niet-natuurlijk beeld. Om dit probleem op te lossen, en het eindresultaat er natuurlijker laten uit te zien werd een tweede aanpak angewend.

### Tweede aanpak

Het resultaat van de eerste aanpak was voor het menselijk oog niet zo goed. Om dit te verbeteren werd gekozen om bij de kleursverandering niet voor een vaste kleur geel te kiezen, maar om de rgb-waarden van de gele kleur te laten afhangen van de blauwe kleur. Dit wordt gedaan door bij de blauwe kleur een bepaalde waarde op te tellen of af te trekken. Om negatieve waarden te vermijden wordt steeds de absolute waarde genomen. Deze waarden werden experimenteel bepaald en leverden volgend resultaat:

Kleur	Offset
Rood	255 (vast)
Groen	+54
Blauw	-215

TABEL 2.2: Offset voor conversie blauwe kleur naar gele kleur

Ook hier wordt elke pixel één voor één overlopen en gecontroleerd of de rgb-waarden binnen de grenswaarden vallen. Indien dit het geval is wordt er bij de huidige waarde van de pixel een offset bijgeteld zoals is weergegeven in bovenstaande tabel. De Matlab-code van deze aanpak wordt hieronder weergegeven:

```
DATA = imread('C:\Users\Gert-Jan\Desktop\DSPmatlab\GroteSmurf.jpg');

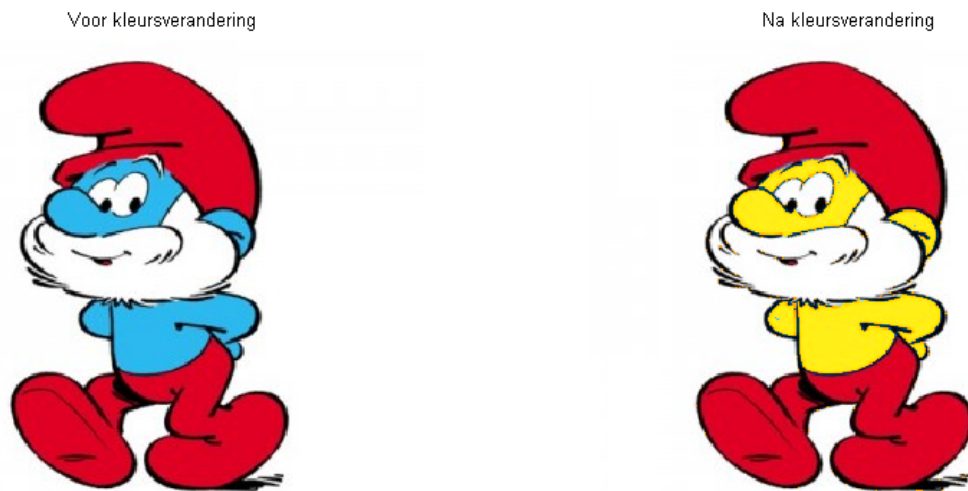
figure
subplot(1,2,1);
image(DATA)
title('Voor kleursverandering')
axis off;

for y=1:300
    for x=1:300
        if DATA(y,x,1)>0 && DATA(y,x,1)<115 && DATA(y,x,2)>100 &&
DATA(y,x,2)<200 && DATA(y,x,3)>0 && DATA(y,x,3)<255
            DATA(y,x,1) = 255;
            DATA(y,x,2) = DATA(y,x,2)+54;
            DATA(y,x,3) = abs(DATA(y,x,3)-215);
        end
    end
end

subplot(1,2,2);
image(DATA)
title('Na kleursverandering')
axis off;
```

LISTING 2.2: Matlab-code tweede aanpak

Het resultaat van deze aanpak is te zien in volgende figuur:



FIGUUR 2.2: Resultaat kleursverandering Smurf: tweede aanpak

Er valt op te merken dat bovenstaande afbeelding veel natuurlijker aanvoelt voor het oog. Dit komt omdat het niet enkel uit één tint geel bestaat, maar uit meerdere tinten geel wat een véél rustiger effect geeft.

## 2.2 Opgave 2

### 2.2.1 Kleursverandering

#### Doelstelling

in deze opgave is het de bedoeling om het groene scherm in de achtergrond te veranderen naar een grijze achtergrond. Net zoals bij de afbeelding van de Smurf bestaat het groen hier niet uit één tint groen, maar meerde tinten groen. Om het groen naar grijs om te vormen moet dus rekening gehouden worden met een bepaald kleurbereik wat de kleur betreft.

#### Werkwijze en resultaat

Net zoals bij de opgave met de Smurf-afbeelding wordt hier ook eerst experimenteel het groene kleurbereik vastgesteld. Dit levert volgend resultaat:

Kleur	Ondergrens	Bovengrens
Rood	0	115
Groen	100	200
Blauw	0	255

TABEL 2.3: Grenswaarden voor blauwe kleur

Net als bij de opgave waarbij de Smurf van kleur veranderd moest worden, wordt ook hier pixel per pixel overlopen of de kleur van in het kleurbereik van groen valt. Indien dit het geval is wordt deze pixel vervangen door een grijze kleur met rgb-waarde 180,180,180. Onderstaande matlabcode werd gegenereerd om de kleursverandering toe te passen:

```
DATA = imread('C:\Users\Gert-Jan\Desktop\DSPmatlab\
    achtergrondveranderen.jpg');

figure
subplot(1,2,1);
image(DATA)
title('Voor kleursverandering')
axis off;

for y=1:600
    for x=1:800
        if DATA(y,x,1)>45 && DATA(y,x,1)<145
            if DATA(y,x,2)>90 && DATA(y,x,2)<185
                if DATA(y,x,3)>29 && DATA(y,x,3)<108
                    DATA(y,x,1) = 180;
                    DATA(y,x,2) = 180;
                    DATA(y,x,3) = 180;
                end
            end
        end
    end
end

subplot(1,2,2);
image(DATA)
title('Na kleursverandering')
axis off;
```

LISTING 2.3: Matlab-code achtergrond veranderen

Het resultaat van deze aanpak is te zien in volgende figuur:



FIGUUR 2.3: Resultaat achtergrond veranderen

Uit bovenstaande figuur is af te leiden dat niet alleen het groene scherm in de achtergrond veranderd is, maar ook enkele andere pixels in de achtergrond. Dit komt omdat er sommige stukken in de achtergrond nog een groene kleur bevat. Alles wat in het bereik van de groene kleur valt zal veranderd worden naar grijs.

### 2.2.2 Kleursverandering: uitbreiding

#### Doelstelling

In de vorige opgave werd steeds iedere pixel in de afbeelding vergeleken om na te gaan of deze in het bereik valt van de groene kleur. Dit heeft als nadeel dat het bereik experimenteel bepaald moet worden en vervolgens handmatig in de code dient ingesteld te worden. Het zou opteren voor een minder omslachtig proces dat minstens een even goed resultaat aflevert. Wanneer men in een tv-studio gebruikt maakt van een groen scherm, wil men dit doorgaans niet gebruiken om het door één andere kleur te vervangen. Meestal gebruikt men dit om de achtergrond te veranderen. In deze uitbreiding zal ook geprobeerd worden om de groene achtergrond te veranderen naar een afbeelding met wolkjes.

### Werkwijze en resultaat

De werkwijze is compleet verschillend met de werkwijze die gevolgd werd in bovenstaande opgaven. In de eerste plaats worden beide afbeeldingen ingeladen in de code. vervolgens wordt ook al plaats gereserveerd om het eindresultaat in op te slaan. Vervolgens gaan we de kleurwaarden per pixel omzetten naar percentages. Dit wil zeggen dat we de rgb-waarden die variëren van 0 tot 255 gaan delen door 255 zodat we het percentage bekomen, een getal tussen 0 en 1.

Vervolgens gaan we de groenwaarde van elke pixel. dit doen we door middel van de kleurkanalen met elkaar te vergelijken. Dit gebeurt aan de hand van volgende formule:  $\text{groenwaarde} = \text{groen} \times (\text{groen} - \text{rood}) \times (\text{groen} - \text{blauw})$ . Hierna wordt ook rekening gehouden met een bepaalde threshold, zodat we niet 1 tint groen bekijken, maar een aantal tinten groen die afgeleid zijn van het gemiddelde. Om na te gaan of een pixel groen is stellen we opnieuw een matrix op met daarin een 0 als de pixel niet groen is, en een 1 als de pixel wel groen is. De beslissing of een pixel al dan niet groen is wordt genomen op basis van de threshold. Wanneer de groenwaarde groter is dan de threshold, dan zal de pixel als groen bestempeld worden.

Nu zullen voor elke kleur (R,G en B) de afbeeldingen overlopen worden. Per kleur zullen de pixels van het studiobeeld overschreven worden met die van de achtergrondaafbeelding indien er in de IsGroen matrix een 1 staat op de plaats met de overeenstemmende pixel. Het resultaat zal opgeslagen worden in een nieuwe variabale. De Matlabcode die bij deze werkwijze hoort, wordt hieronder weergegeven:

```
StudioBeeld = imread('C:\Users\Gert-Jan\Desktop\DSPmatlab\
    achtergrondveranderen.jpg');
Achtergrond = imread('C:\Users\Gert-Jan\Desktop\DSPmatlab\wolken.
    jpg');

Eindresultaat = StudioBeeld;

figure

subplot(1,3,1);
image(Achtergrond)
title('Achtergrondaafbeelding')
axis off;

subplot(1,3,2);
image(StudioBeeld)
title('Studiobeeld voor verandering')
axis off;

%zet kleurwaarden om naar percentage (schaal van 0-255 --> 0-1)
KleurPercentage = double( StudioBeeld)/255;

%groenheid berekenen van 3 kleurkanalen Groen * (groen-rood) * (
    groen-blauw)
```

```

GroenWaarde = KleurPercentage(:,:,2).*(KleurPercentage(:,:,2)-
    KleurPercentage(:,:,1)).*(KleurPercentage(:,:,2)-
    KleurPercentage(:,:,3));
%threshold om beter eindresultaat te bereiken. --> er is niet 1
    tint groen,
%maar meerdere.
Threshold = 0.8*mean(GroenWaarde(GroenWaarde>0));
IsGroen = GroenWaarde > Threshold;

for i = 1:3
    % sla de rgb waarde van de achtergrond op
    rgbValAchtergrond = Achtergrond(:,:,i);
    % sla de rgb waarde van het studiobeeld op
    rgbValStudioBeeld = StudioBeeld(:,:,i);
    %logische vergelijking. Als het studiobeeld groen is dan moet
    de
    %achtergrond op het studiobeeld geplaatst worden
    rgbValStudioBeeld(IsGroen) = rgbValAchtergrond(IsGroen);
    % sla op in eindresultaat
    Eindresultaat(:,:,i) = rgbValStudioBeeld;
end

subplot(1,3,3);
image(Eindresultaat)
title('Studiobeeld na verandering')
axis off;

```

LISTING 2.4: Matlab-code achtergrond veranderen

Het resultaat van deze aanpak is te zien in volgende figuur:



FIGUUR 2.4: Resultaat achtergrond veranderen - uitbreiding

Men kan opmerken dat inderdaad de groene achtergrond veranderd is naar de wolkjes. Men dient wel op te merken dat beide afbeeldingen even groot moeten zijn om bovenstaande code correct te kunnen uitvoeren. De code is dus nog niet optimaal, maar maakt het wel al mogelijk de achtergrond te veranderen zonder dat er een kleurenbereik ingesteld dient te worden.



## Hoofdstuk 3

# Algemeen besluit

Na deze verschillende opdrachten uit te voeren, en zelf ook wat variaties op de oefeningen met Matlab geoefend te hebben is het begrip DSP op vlak van video toch heel wat opgeklaard. We kunnen reeds een verzameling aan kleuren binnen vooropgestelde rgb-grenzen vervangen door een andere kleur. Hoe nauwer deze grenzen, hoe minder de afbeelding zal worden beïnvloed. Als dit mogelijk is voor een enkel frame, rest ons enkel nog de stap naar real-time processing. Dit houdt in dat er continue frames binnekomen die dan ook één voor één ver- en bewerkt moeten worden. Als we elk zo'n frame als een enkele statische afbeelding kunnen bewerken, kunnen we de reeds opgedane ervaring inzetten om sneller tot ons gewenste resultaat te komen. Indien de overstap van Matlab naar DSP++ niet te groot is, dan hebben we ons dus reeds voorbereid op de uiteindelijke opgave.

# Bijlagen

# Bijlage A

## Matlab Code

### A.1 Opgave 1 Deel 1

```
DATA = imread('C:\Users\Gert-Jan\Desktop\DSPmatlab\GroteSmurf.jpg');  
);  
  
figure  
subplot(1,2,1);  
image(DATA)  
title('Voor kleursverandering')  
axis off;  
  
for y=1:300  
    for x=1:300  
        if DATA(y,x,1)>0 && DATA(y,x,1)<115 && DATA(y,x,2)>100 &&  
DATA(y,x,2)<200 && DATA(y,x,3)>0 && DATA(y,x,3)<255  
            DATA(y,x,1) = 255;  
            DATA(y,x,2) = 255;  
            DATA(y,x,3) = 0;  
        end  
    end  
end  
  
subplot(1,2,2);  
image(DATA)  
title('Na kleursverandering')  
axis off;
```

LISTING A.1: Matlab-code eerste aanpak

## A.2 Opgave 1 Deel 2

```
DATA = imread('C:\Users\Gert-Jan\Desktop\DSPmatlab\GroteSmurf.jpg');  
);  
  
figure  
subplot(1,2,1);  
image(DATA)  
title('Voor kleursverandering')  
axis off;  
  
for y=1:300  
    for x=1:300  
        if DATA(y,x,1)>0 && DATA(y,x,1)<115 && DATA(y,x,2)>100 &&  
DATA(y,x,2)<200 && DATA(y,x,3)>0 && DATA(y,x,3)<255  
            DATA(y,x,1) = 255;  
            DATA(y,x,2) = DATA(y,x,2)+54;  
            DATA(y,x,3) = abs(DATA(y,x,3)-215);  
        end  
    end  
end  
  
subplot(1,2,2);  
image(DATA)  
title('Na kleursverandering')  
axis off;
```

LISTING A.2: Matlab-code tweede aanpak

## A.3 Opgave 2 Deel 1

```
DATA = imread('C:\Users\Gert-Jan\Desktop\DSPmatlab\
    achtergrondveranderen.jpg');

figure
subplot(1,2,1);
image(DATA)
title('Voor kleursverandering')
axis off;

for y=1:600
    for x=1:800
        if DATA(y,x,1)>45 && DATA(y,x,1)<145
            if DATA(y,x,2)>90 && DATA(y,x,2)<185
                if DATA(y,x,3)>29 && DATA(y,x,3)<108
                    DATA(y,x,1) = 180;
                    DATA(y,x,2) = 180;
                    DATA(y,x,3) = 180;
                end
            end
        end
    end
end

subplot(1,2,2);
image(DATA)
title('Na kleursverandering')
axis off;
```

LISTING A.3: Matlab-code achtergrond veranderen

## A.4 Opgave 2 Deel 2

```

StudioBeeld = imread('C:\Users\Gert-Jan\Desktop\DSPmatlab\
    achtergrondveranderen.jpg');
Achtergrond = imread('C:\Users\Gert-Jan\Desktop\DSPmatlab\wolken.
    jpg');

Eindresultaat = StudioBeeld;

figure

subplot(1,3,1);
image(Achtergrond)
title('Achtergrondaafbeelding')
axis off;

subplot(1,3,2);
image(StudioBeeld)
title('Studiobeeld voor verandering')
axis off;

%zet kleurwaarden om naar percentage (schaal van 0-255 --> 0-1)
KleurPercentage = double( StudioBeeld)/255;

%groenheid berekenen van 3 kleurkanalen Groen * (groen-rood) * (
    groen-blauw)
GroenWaarde = KleurPercentage(:,:,2).*(KleurPercentage(:,:,2)-
    KleurPercentage(:,:,1)).*(KleurPercentage(:,:,2)-
    KleurPercentage(:,:,3));
%threshold om beter eindresultaat te bereiken. --> er is niet 1
    tint groen,
%maar meerdere.
Threshold = 0.8*mean(GroenWaarde(GroenWaarde>0));
IsGroen = GroenWaarde > Threshold;

for i = 1:3
    % sla de rgb waarde van de achtergrond op
    rgbValAchtergrond = Achtergrond(:,:,i);
    % sla de rgb waarde van het studiobeeld op
    rgbValStudioBeeld = StudioBeeld(:,:,i);
    %logische vergelijking. Als het studiobeeld groen is dan moet
    de
    %achtergrond op het studiobeeld geplaatst worden
    rgbValStudioBeeld(IsGroen) = rgbValAchtergrond(IsGroen);
    % sla op in eindresultaat
    Eindresultaat(:,:,i) = rgbValStudioBeeld;
end

subplot(1,3,3);
image(Eindresultaat)
title('Studiobeeld na verandering')
axis off;

```

LISTING A.4: Matlab-code achtergrond veranderen