PROJECT-4

COLOR TRANSFORMATION
PART - 2

EE5356 Digital Image Processing
Dr. K. R. Rao

Submitted By:
Ashutosh Desai
UTA ID:1001388602
Email:ashutoshrajesh.desai@mavs.uta.edu
Feb 27th 2018

EE5356 Project Assignment 3A: Color Transformation.

The purpose of this project is to obtain knowledge of image representation in color space, get hands on experiences of color conversion between different color spaces, and understand their application in video compression. The test image, flowers.bmp (500x362), for this project is available at the class website. Note that in video/image coding international standards this is required.

Question*:*

1. Read the given color image (flowers.tiff) into MATLAB and decompose it into R,G,B components and show those images respectively in your report.

2. Perform color space conversion from R, G, B to Y,Cb,Cr, and show each component image respectively in your report.

3. Perform downsampling in YCbCr domain to form 4:2:2 format, then apply luminance and chrominance decimation filters followed by interpolation filters and upsampling to original resolution and inverse conversion to $R^{'}$,$G^{'}$, $B^{'}$, show the reconstructed images.

Y: Luminance

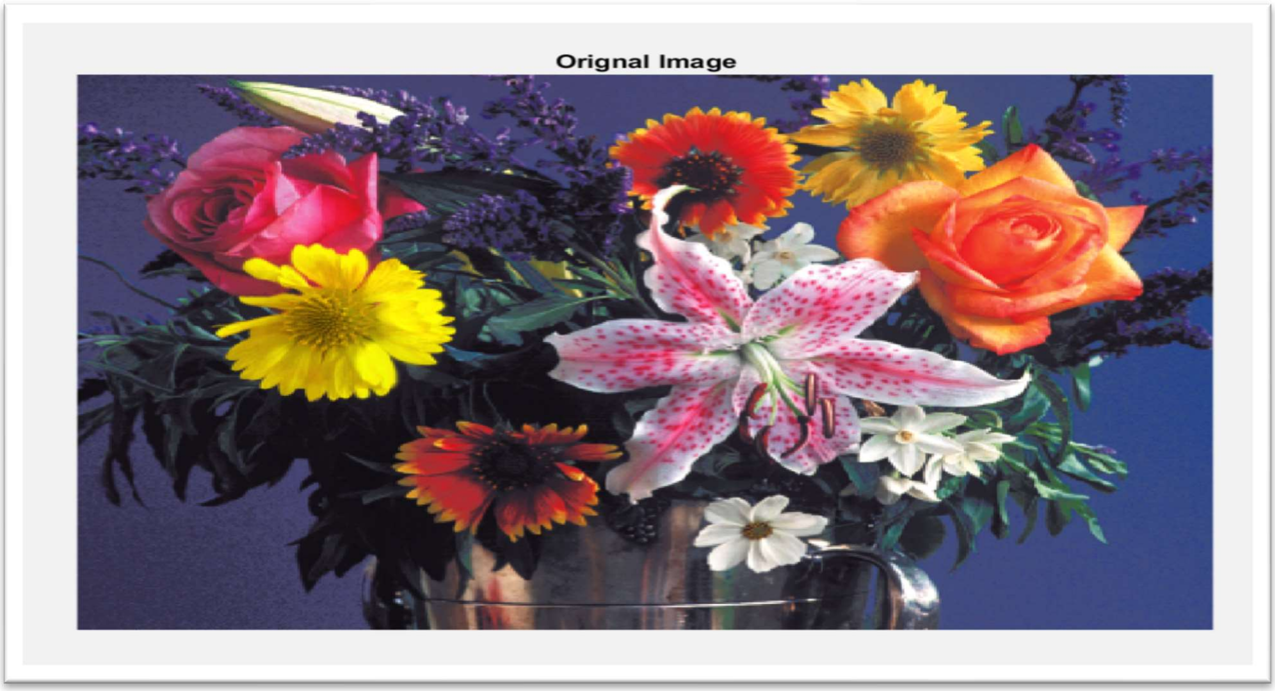
Cb, Cr: Color difference signals

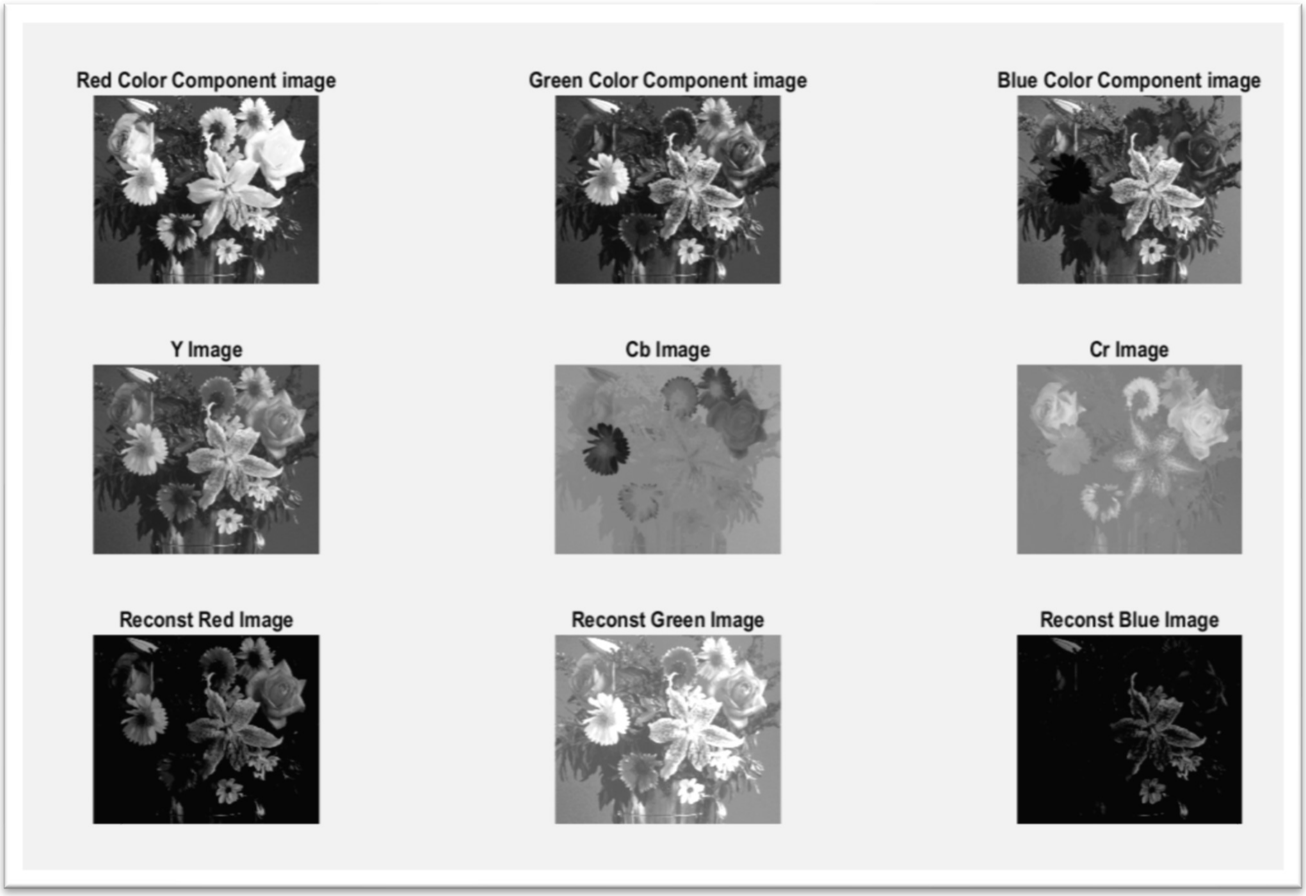
MATLAB SCRIPT:-

```
clc;
clear all;
close all;

Image= double(imread('D:\STUDY\DIP\Test img\flowers.bmp'));
figure(1);
imshow('D:\STUDY\DIP\Test img\flowers.bmp');
title('Orignal Image');

RedColor_component=Image(:,:,1);
GreenColor_component=Image(:,:,2);
BlueColor_component=Image(:,:,3);

figure(2);
subplot(3,3,1);
imshow(uint8(RedColor_component));
title('Red Color Component image');
subplot(3,3,2);
imshow(uint8(GreenColor_component));
title('Green Color Component image');
subplot(3,3,3);
imshow(uint8(BlueColor_component));
title('Blue Color Component image');

Y_Luminance=RedColor_component*0.257 +
GreenColor_component*0.504 + BlueColor_component*0.098 +16;
Cb_Chrominance= RedColor_component*-0.148 +
GreenColor_component*-0.291 + BlueColor_component*0.439+128;
Cr_Chrominance= RedColor_component*0.439 +
GreenColor_component*-0.368 + BlueColor_component*-0.071+128;

subplot(3,3,4);
imshow(uint8(Y_Luminance));
title('Y Image');
subplot(3,3,5);
imshow(uint8(Cb_Chrominance));
title('Cb Image');
subplot(3,3,6);
imshow(uint8(Cr_Chrominance));
title('Cr Image');

filter=1/8*[1 3 3 1];
for x=1:362
```

```
Cb_deci(x,:)=conv(Cb_Chrominance(x,:),filter);
Cr_deci(x,:)=conv(Cr_Chrominance(x,:),filter);
end

%Image downsampling.
Down_Samp_Cb=Cb_Chrominance(:,1:2:500);
Down_Samp_Cr=Cr_Chrominance(:,1:2:500);

%Image upsampling.
up_Samp_Cb=zeros(362,500);
up_Samp_Cr=zeros(362,500);
up_Samp_Cb(:,1:2:500)=Down_Samp_Cb;
up_Samp_Cr(:,1:2:500)=Down_Samp_Cr;

%Image Interpolation,horizontal filtering.
for x=1:362
Cb_up(x,:)=conv(up_Samp_Cb(x,:),filter);
Cr_up(x,:)=conv(up_Samp_Cr(x,:),filter);
end

Cb_up=Cb_up(:,1:500);
Cr_up=Cr_up(:,1:500);

%Image Reconstruction
Red_reconst=1.164*(Y_Luminance-16)+1.596*(Cr_up-128);
Green_reconst=1.164*(Y_Luminance-16)-0.813*(Cr_up-128)-
0.392*(Cb_up-128);
Blue_reconst=1.164*(Y_Luminance-16)+2.017*(Cb_up-128);
subplot(3,3,7);
imshow(uint8(Red_reconst));
title('Reconst Red Image');
subplot(3,3,8);
imshow(uint8(Green_reconst));
title('Reconst Green Image');
subplot(3,3,9);
imshow(uint8(Blue_reconst));
title('Reconst Blue Image')
```

OUTPUT:-


Orignal Image


Red Color Component image
Green Color Component image
Blue Color Component image
Y Image
Cb Image
Cr Image
Reconst Red Image
Reconst Green Image
Reconst Blue Image

PROCEDURE:

- A color image usually composed of Red, Green and Blue color components. Initially, we will separate each of the color components from the original image i.e. separate the Red, Green and Blue color components by setting the value of other two-color components as 0 (Zero) when finding a specific color. For example, if we want to extract the red color component within an image we will set the value of Green and Blue color component as zero.

➢ **MATLAB CODE FOR EXTRACTING THE COLOR COMPONENTS: -**

```
RedColor_component=Image(:,:,1);
GreenColor_component=Image(:,:,2);
BlueColor_component=Image(:,:,3);
```

- After obtaining the individual color components we will then find the Luminance(Y) and Chrominance (Cb , Cr) components of the image.

➢ **MATLAB CODE FOR OBTAINING Y,Cb and Cr**

```
Y_Luminance=RedColor_component*0.257 + GreenColor_component*0.504 +
BlueColor_component*0.098 +16;
Cb_Chrominance= RedColor_component*-0.148 + GreenColor_component*-
0.291 + BlueColor_component*0.439+128;
Cr_Chrominance=  RedColor_component*0.439  +  GreenColor_component*-
0.368 + BlueColor_component*-0.071+128;
```

- After obtaining the respective chrominance and luminance values we perform Down-sampling of the Chrominance component.

➢ **MATLAB CODE FOR DOWNSAMPLING: -**

```
Down_Samp_Cb=Cb_Chrominance(:,1:2:500);
Down_Samp_Cr=Cr_Chrominance(:,1:2:500);
```

- Thereafter we perform the decimation process of the resultant image by a factor of 2.

➢ **MATLAB CODE:**

```
filter=1/8*[1 3 3 1];
for x=1:362
Cb_deci(x,:)=conv(Cb_Chrominance(x,:),filter);
Cr_deci(x,:)=conv(Cr_Chrominance(x,:),filter);
end
```

- Then it is followed by the Interpolation process and then Up-sampling.

➢ **MATLAB CODE FOR UP-SAMPLING:**

```
up_Samp_Cb=zeros(362,500);
up_Samp_Cr=zeros(362,500);
up_Samp_Cb(:,1:2:500)=Down_Samp_Cb;
up_Samp_Cr(:,1:2:500)=Down_Samp_Cr;
```

- At last the reconstructed image is obtained by combining the final values of Luminance and Chrominance.

- **MATLAB CODE FOR IMAGE RECONSTRUCTION:**

```
Red_reconst=1.164*(Y_Luminance-16)+1.596*(Cr_up-128);
Green_reconst=1.164*(Y_Luminance-16)-0.813*(Cr_up-128)- 0.392*(Cb_up-
128);
Blue_reconst=1.164*(Y_Luminance-16)+2.017*(Cb_up-128);
subplot(3,3,7);
```

CONCLUSION:

From the above results, it can be inferred that the image obtained after reconstruction is different as compared to the original image. So the above procedure is not 100% efficient. Still it is used in some video compression area thus in order to reduce the bandwidth occupied during transmission.