

**THE UNIVERSITY OF TEXAS AT ARLINGTON, TEXAS  
DEPARTMENT OF ELECTRICAL ENGINEERING**

**EE 5356**

**DIGITAL IMAGE PROCESSING**

**PROJECT # 4**

**by**

**SOUTRIK MAITI**

**1001569883**

**Presented to**

**Dr. K.R.RAO**

**Mar 1, 2019**

**Color Transformation 3A**

MATLAB Code:

%% Read the image

img\_origin = double(imread('flowers.bmp'));

%% The R,G,B components are as follows:

R\_img = img\_origin(:,:,1);

G\_img = img\_origin(:,:,2);

B\_img = img\_origin(:,:,3);

%% Reconstruction of image from the components:

I(:,:,1) = R\_img;

I(:,:,2) = G\_img;

I(:,:,3) = B\_img;

%% Display original Image

figure(1)

imshow(uint8(img\_origin));

title('Original Image (500\*362)');

%% Display the RGB components of the image

figure(2)

subplot(4,3,1)

imshow(uint8(R\_img));

title('Red Channel');

subplot(4,3,2)

imshow(uint8(G\_img));

title('Green Channel');

subplot(4,3,3)

imshow(uint8(B\_img));

title('Blue Channel');

%% Color Transform

Y = R\_img \* 0.257 + G\_img \* 0.504 + B\_img \* 0.098 + 16;

Cb = R\_img \* (-0.148) + G\_img \* (-0.291) + B\_img \* 0.439 + 128;

Cr = R\_img \* 0.439 + G\_img \* (-0.368) + B\_img \* (-0.071) + 128;

%% Display Y,Cb,Cr components of the image

subplot(4,3,4)

imshow(uint8(Y))

title('Y');

subplot(4,3,5)

imshow(uint8(Cb))

title('Cb');

subplot(4,3,6)

imshow(uint8(Cr))

title('Cr');

%% Decimation filters for Y, Cb and Cr

Y\_decim = [-29 0 88 138 88 0 -29] / 256;

C\_decim = [1 3 3 1] / 8;

%% Performing decimation

Cr\_down = downsample(Cr',2)';

Cb\_down = downsample(Cb',2)';

Y\_filter = imfilter(Y, Y\_decim, 'circular', 'conv');

Y\_down = downsample(downsample(Y\_filter,2)',2)';

Cr\_Int = imfilter(Cr\_down, C\_decim, 'circular', 'conv');

Cr\_down = downsample(downsample(Cr\_Int,2)',2)';

Cb\_filter = imfilter(Cb\_down, C\_decim, 'circular', 'conv');

Cb\_down = downsample(downsample(Cb\_filter,2)',2)';

%% Displaying the decimated values of Y,Cb,Cr

subplot(4,3,7)

imshow(uint8(Y\_down))

title('Y Decimated');

subplot(4,3,8)

imshow(uint8(Cb\_down))

title('Cb Decimated');

subplot(4,3,9)

imshow(uint8(Cr\_down))

title('Cr Decimated');

%% Interpolation filter for Y,Cb,Cr

Y\_Interpol = [-12 0 140 256 140 0 -12] / 256;

C\_Interpol = [ 1 0 3 8 3 0 1] / 8;

%% Performing decimation

Y\_Up = upsample(upsample(Y\_down,2)',2)';

Y\_Up\_filter = imfilter(Y\_Up,Y\_Interpol,'circular','conv');

Y\_Int = imfilter(Y\_Up\_filter,Y\_Interpol','circular','conv');

Cb\_Up = upsample(upsample(Cb\_down,2)',2)';

Cb\_Up\_filter = imfilter(Cb\_Up,C\_Interpol','circular','conv');

Cb\_Int = imfilter(Cb\_Up\_filter,C\_Interpol,'circular','conv');

Cb\_Up\_Up = upsample(Cb\_Int',2)';

c1 = size(Cb\_Up\_Up,2);

Cb\_Up\_Up(:,c1) = Cb\_Up\_Up(:,c1-1);

for cnt = 2:2:c1-1

Cb\_Up\_Up(:,cnt) = (Cb\_Up\_Up(:,cnt-1) + Cb\_Up\_Up(:,cnt+1))/2;

end

Cr\_Up = upsample(upsample(Cr\_down,2)',2)';

Cr\_Up\_filter = imfilter(Cr\_Up,C\_Interpol','circular','conv');

Cr\_Int = imfilter(Cr\_Up\_filter,C\_Interpol,'circular','conv');

Cr\_Up\_Up = upsample(Cr\_Int',2)';

c2 = size(Cr\_Up\_Up,2);

Cr\_Up\_Up(:,c2) = Cr\_Up\_Up(:,c2-1);

for cnt = 2:2:c2-1

Cr\_Up\_Up(:,cnt) = (Cr\_Up\_Up(:,cnt-1) + Cr\_Up\_Up(:,cnt+1))/2;

end

%% Displaying the decimation values

subplot(4,3,10)

imshow(uint8(Y\_Int))

title('Y Interpolated');

subplot(4,3,11)

imshow(uint8(Cb\_Up\_Up))

title('Cb Interpolated');

subplot(4,3,12)

imshow(uint8(Cr\_Up\_Up))

title('Cr Interpolated');

saveas(gca,'Components','jpg');

%% Reconstructing R,G,B from interpolated Y,Cr,Cb

r\_recon = 1.164\*(Y\_Int-16) + 1.596\*(Cr\_Up\_Up-128);

g\_recon = 1.164\*(Y\_Int-16) - 0.813\*(Cr\_Up\_Up - 128) - 0.392 \* (Cb\_Up\_Up-128);

b\_recon = 1.164\*(Y\_Int-16) + 2.017\*(Cb\_Up\_Up-128);

recon\_R = zeros(362,500,3);

recon\_G = zeros(362,500,3);

recon\_B = zeros(362,500,3);

recon\_R(:,:,1) = r\_recon;

recon\_G(:,:,2) = g\_recon;

recon\_B (:,:,3) = b\_recon;

recon\_img = zeros(362,500,3);

recon\_img(:,:,1)=recon\_R(:,:,1);

recon\_img(:,:,2)=recon\_G(:,:,2);

recon\_img(:,:,3)=recon\_B(:,:,3);

%% Displaying reconstructed R,G,B components

figure(3);

subplot(1,3,1);

imshow(uint8(recon\_R));

title('R Reconstructed');

subplot(1,3,2);

imshow(uint8(recon\_G));

title('G Reconstructed');

subplot(1,3,3);

imshow(uint8(recon\_B));

title('B Reconstructed');

saveas(gca,'recon\_comp','jpg');

%% Display reconstructed image

figure(4);

imshow(uint8(recon\_img));

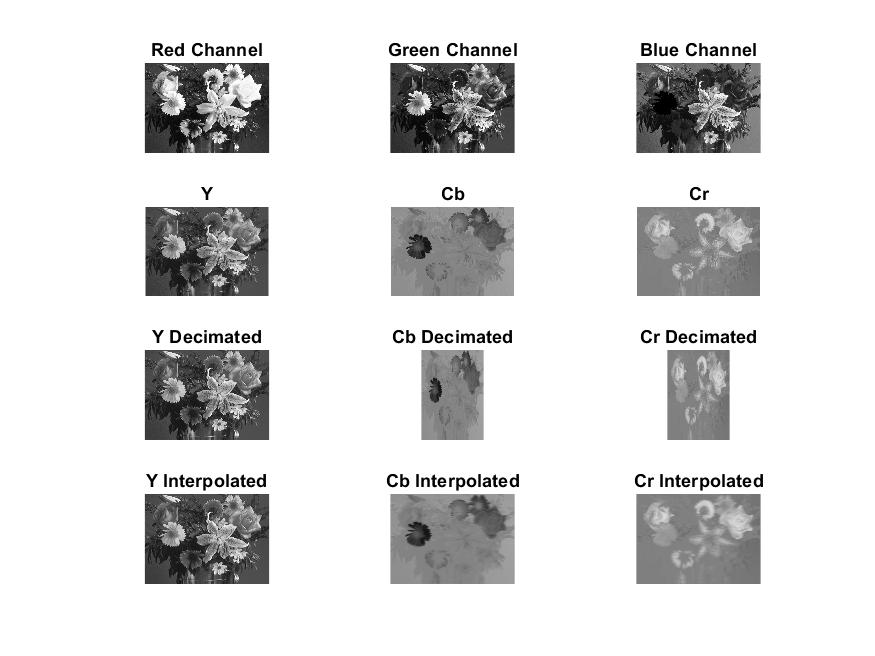
title('Reconstructed Image(500\*362)');

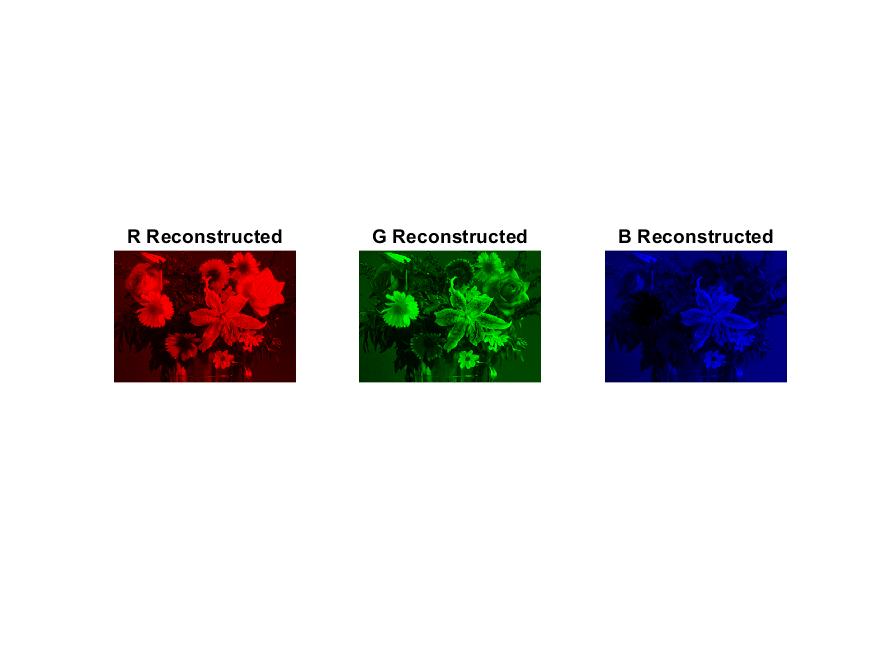
saveas(gca,'recon\_img','jpg');

**Results:**

****

****

****

****

**Procedure:**

* The image is decomposed into its RGB components.
* It is then converted to YCbCr components with application of the conversion formula.
* The YCbCr components are down sampled in the ratio 4:2:2.
* To this down sampled YCbCr components, the process of Decimation filtering is performed.
* To bring back the components to the original resolution a process called up sampling is performed.
* The up sampled image is further done by Interpolation filtering.
* YCbCr domain is obtained after Conversion from RGB domain.
* From RGB domain, the reconstructed image is obtained.