



**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:

Original Image (500*362)



Reconstructed Image(500*362)



Red Channel



Green Channel



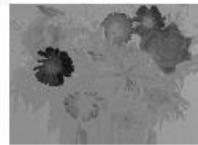
Blue Channel



Y



Cb



Cr



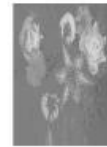
Y Decimated



Cb Decimated



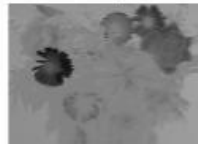
Cr Decimated



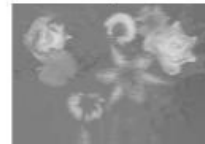
Y Interpolated



Cb Interpolated



Cr Interpolated





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.