EE5356- PROJECT 3A- COLOR TRANSFORMATION

NAME: PAVAI ARCHIMEDES

ST ID: 1001233996

**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 down sampling in YCbCr domain to form 4:2:2 format, then apply luminance and chrominance decimation filters followed by interpolation filters and up sampling to original resolution and inverse conversion to R’, B’, G’**

**Show the reconstructed images.**

**Y: Luminance Cb, Cr: Color difference signals**

**MATLAB PROGRAM:**

clc;

clear all;

close all;

og\_im = imread('C:\Users\PAVAI ARCHIMEDES\Desktop\flowers.bmp');

imshow(og\_im);

title('Original Image-Flowers.bmp(500\*362) ');

figure;

R = double(og\_im(:,:,1)); % R component

G = double(og\_im(:,:,2)); % G component

B = double(og\_im(:,:,3)); % B component

og\_im1 = zeros(362,500,3);

org\_image2 = zeros(362,500,3);

org\_image3 = zeros(362,500,3);

og\_im1(:,:,1) = R;

figure(2);

subplot(3,1,1);

image(uint8(og\_im1));

title('Red Component');

org\_image2(:,:,2) = G;

figure(3);

subplot(3,1,2);

image(uint8(org\_image2));

title('Green Component');

org\_image3(:,:,3) = B;

figure(4);

subplot(3,1,3);

image(uint8(org\_image3));

title('Blue Component');

Y=((0.257\*R)+(0.504\*G)+(0.098\*B)+16); %Luminance

Cb=((-0.148\*R)+(-0.291\*G)+(0.439\*B)+128); %Color Difference Signals

Cr=((0.439\*R)+(-0.368\*G)+(-0.071\*B)+128); %Color Difference Signals

figure;

subplot(3,4,1);

imshow(uint8(Y));

title('Y');

subplot(3,4,5);

imshow(uint8(Cb));

title('Cb'); %Display Color Difference Signal Cb

subplot(3,4,9);

imshow(uint8(Cr));

title('Cr'); %Display Color Difference Signal Cr

% horizontal downsampling Cb and Cr by a factor of 2

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

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

%Decimation filter used for luminance

Y\_Decimation\_Fiter = [-29 0 88 138 88 0 -29]./256;

Y\_filter = imfilter(double(Y),Y\_Decimation\_Fiter,'circular','conv'); %Apply decimation filter on luminance

Y\_horiz\_vert = downsample(downsample(Y\_filter,2)',2)'; % Horizontal and Vertical downsampling on Luminance

%Decimation filter used for chrominance

CbCr\_Decimation\_Filter = [1 3 3 1]./8;

Cr\_horiz\_filter = imfilter(double(Cr\_horiz),CbCr\_Decimation\_Filter,'circular','conv');

Cr\_horiz\_vert = downsample(downsample(Cr\_horiz\_filter,2)',2)';

Cb\_horiz\_filter = imfilter(double(Cb\_horiz),CbCr\_Decimation\_Filter,'circular','conv');

Cb\_horiz\_vert = downsample(downsample(Cb\_horiz\_filter,2)',2)';

%Display downsampled versions of Y,Cb,Cr

subplot(3,4,2);

imshow(uint8(Y\_horiz\_vert));

title('Decimation -Y');

subplot(3,4,6);

imshow(uint8(Cr\_horiz\_vert));

title('Decimation -Cr');

subplot(3,4,10);

imshow(uint8(Cb\_horiz\_vert));

title('Decimation -Cb');

Y\_Inter\_Filter = [-12 0 140 256 140 0 -12]./256;

Y\_hori\_verti\_Up = upsample(upsample(Y\_horiz\_vert,2)',2)'; % Horizontal and Vertical Upsampling of decimated version of luminance

Y\_horiz\_verti\_Up\_filter = imfilter(double(Y\_hori\_verti\_Up),Y\_Inter\_Filter,'circular','conv'); %Apply interpolation filter on upsampled luminance

Y\_filter = imfilter(double(Y\_horiz\_verti\_Up\_filter),Y\_Inter\_Filter','circular','conv');

%Interpolation filter for chrominance

Cb\_Cr\_inter\_filter = [ 1 0 3 8 3 0 1]./8;

%For Chrominance Cr

Cr\_horiz\_vert\_Up = upsample(upsample(Cr\_horiz\_vert,2)',2)'; % Horizontal and Vertical Upsampling decimated version of chrominance Cr

Cr\_horiz\_vert\_Up\_filter = imfilter(Cr\_horiz\_vert\_Up,Cb\_Cr\_inter\_filter','circular','conv'); % Apply interpolation filter on upsampled chrominance Cr

Cr\_filter = imfilter(Cr\_horiz\_vert\_Up\_filter,Cb\_Cr\_inter\_filter,'circular','conv'); %Apply filter to upsampled value of Cr

Cr\_horiz\_vert\_Up1 = upsample(Cr\_filter',2)'; % Horizontal and Vertical Upsampling of upsampled version of chrominance Cr

[r c] = size(Cr\_horiz\_vert\_Up1);

Cr\_horiz\_vert\_Up1(:,c) = Cr\_horiz\_vert\_Up1(:,c-1);

for count = 2:2:c-1

Cr\_horiz\_vert\_Up1(:,count) = (Cr\_horiz\_vert\_Up1(:,count-1) + Cr\_horiz\_vert\_Up1(:,count+1))/2;

end

%For Chrominanc Cb

Cb\_horiz\_vert\_Up = upsample(upsample(Cb\_horiz\_vert,2)',2)'; % Horizontal and Vertical Upsampling decimated version of chrominance Cb

Cb\_horiz\_vert\_Up\_filter = imfilter(double(Cb\_horiz\_vert\_Up),Cb\_Cr\_inter\_filter,'circular','conv'); % Apply interpolation filter on upsampled chrominance Cb

Cb\_filter = imfilter(double(Cb\_horiz\_vert\_Up\_filter),Cb\_Cr\_inter\_filter','circular','conv'); %Apply filter to upsampled value of Cb

Cb\_horiz\_vert\_Up1 = upsample(Cb\_filter',2)';

[r c] = size(Cb\_horiz\_vert\_Up1);

Cb\_horiz\_vert\_Up1(:,c) = Cb\_horiz\_vert\_Up1(:,c-1);

for count = 2:2:c-1

Cb\_horiz\_vert\_Up1(:,count) = (Cb\_horiz\_vert\_Up1(:,count-1) + Cb\_horiz\_vert\_Up1(:,count+1))/2;

end

%Display upsampled versions of Y,Cb,Cr

subplot(3,4,3);

imshow(uint8(Y\_filter));

title('Interpolation -Y '); %Y= 500\*362

subplot(3,4,7);

imshow(uint8(Cr\_horiz\_vert\_Up1));

title('Interpolation -Cr'); %Cr= 500\*362

subplot(3,4,11);

imshow(uint8(Cb\_horiz\_vert\_Up1));

title('Interpolation -Cb'); %Cb= 500\*362

%Inverse Conversion from upsampled values of Y,Cb,Cr to R1,G1,B1

R1 = 1.164\*(Y\_filter-16) + 1.596\*(Cr\_horiz\_vert\_Up1-128);

G1 = 1.164\*(Y\_filter-16) - 0.813\*(Cr\_horiz\_vert\_Up1-128) - 0.392\*(Cb\_horiz\_vert\_Up1-128);

B1 = 1.164\*(Y\_filter-16) + 2.017\*(Cb\_horiz\_vert\_Up1-128);

%Reconstruction of image

recon\_image1 = zeros(362,500,3);

recon\_image2 = zeros(362,500,3);

recon\_image3 = zeros(362,500,3);

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

recon\_image1(:,:,1) = R1;

subplot(3,4,4);

image(uint8(recon\_image1));

title('R Component');

recon\_image2(:,:,2) = G1;

subplot(3,4,8);

image(uint8(recon\_image2));

title('G Component');

recon\_image3 (:,:,3) = B1;

subplot(3,4,12);

image(uint8(recon\_image3));

title('B Component');

recon\_image(:,:,1)=recon\_image1(:,:,1);

recon\_image(:,:,2)=recon\_image2(:,:,2);

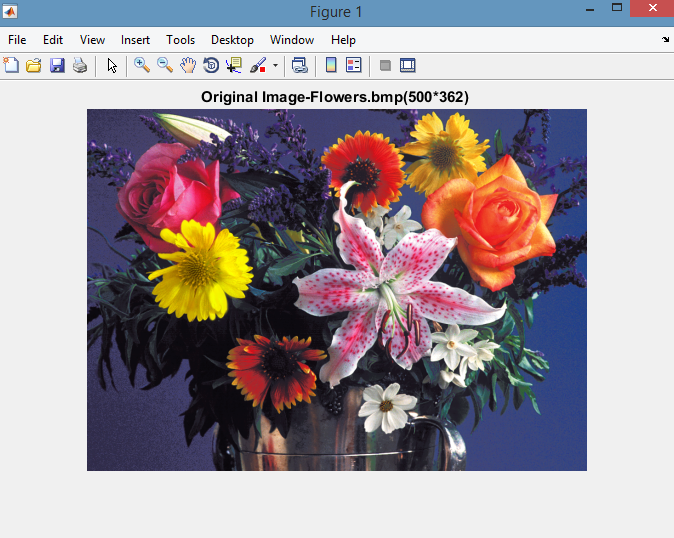
recon\_image(:,:,3)=recon\_image3(:,:,3);

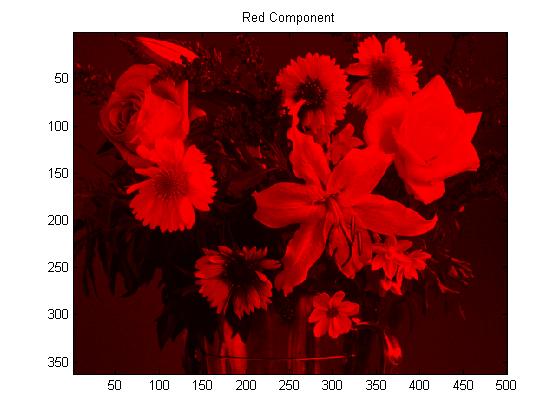
figure(6);

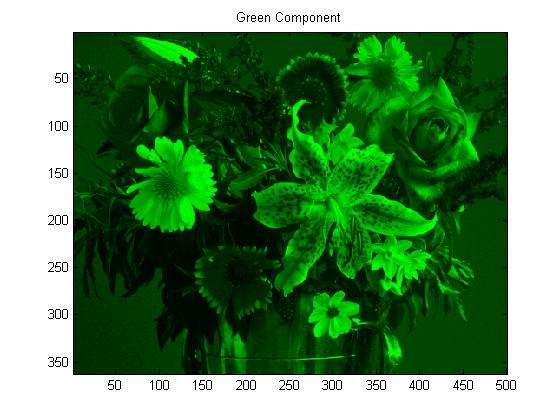
image(uint8(recon\_image));

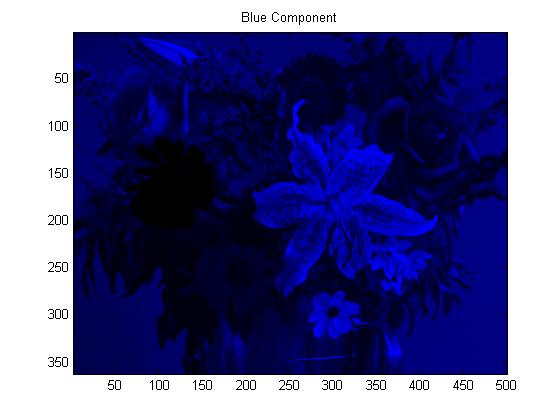
title('Reconstructed Image-Flowers.bmp(500\*362)');

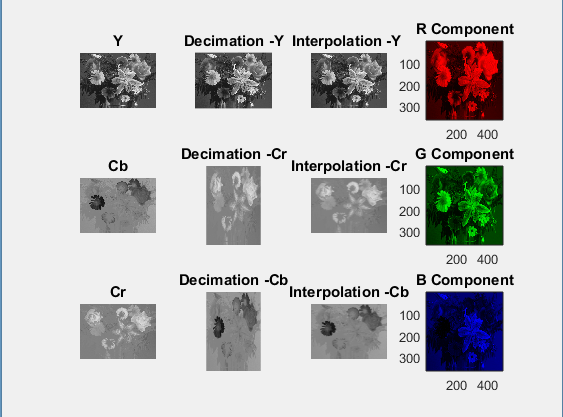
OUTPUT:

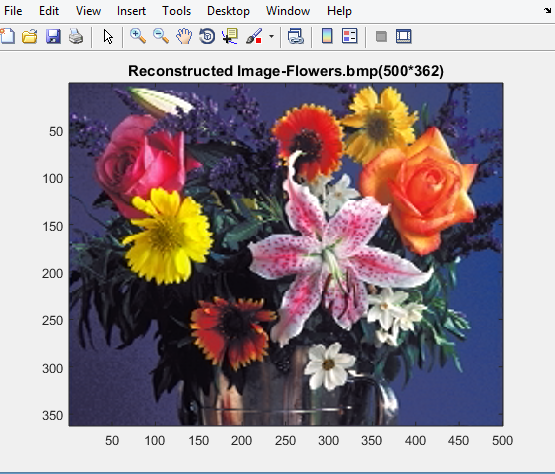












**CONCLUSION:**

The original color image flowers.bmp is initially separated into Red, Green and Blue components.