EE5351\_ DIGITAL VIDEO CODING

1:2 SUB SAMPLING AND 2:1 UP SAMPLING

NAME:PAVAI ARCHIMEDES

ST ID: 1001233996

PROGRAM:

clc;

close all;

clear all;

Image=double(imread('lena512.bmp'));

figure;

imshow(uint8(Image));

title('ACTUAL IMAGE OF LENA');

Avg\_sub=subavg(Image);

Avg\_linear=linear\_inter(Avg\_sub);

MSE\_avg\_linear\_inter=(sum(sum((Image-Avg\_linear).^2)))/(512\*512);

PSNR\_avg\_linear\_inter= 10\*log10((255^2)/MSE\_avg\_linear\_inter);

SSIM\_avg\_linear\_inter=SSIM\_Index(Image,Avg\_linear);

figure;

subplot(1,3,1);

imshow(uint8(Avg\_linear));

title([{'AVERAGE SUBSAMPLING/LINEAR INTERPOLATION MSE= ',MSE\_avg\_linear\_inter,' PSNR= ',PSNR\_avg\_linear\_inter}]);

Avg\_filter=linear\_filter(Avg\_sub);

MSE\_avg\_linear\_filter=(sum(sum((Image-Avg\_filter).^2)))/(512\*512);

PSNR\_avg\_linear\_filter=10\*log10((255^2)/MSE\_avg\_linear\_filter);

SSIM\_avg\_linear\_filter=SSIM\_Index(Image,Avg\_filter);

subplot(1,3,2);

imshow(uint8(Avg\_filter));

title([{' AVERAGE SUBSAMPLING/LINEAR FILTER MSE= ',MSE\_avg\_linear\_filter,' PSNR= ',PSNR\_avg\_linear\_filter}]);

Avg\_DCT=DCT\_up(Avg\_sub);

MSE\_avg\_DCT=(sum(sum((Image-Avg\_DCT).^2)))/(512\*512);

PSNR\_avg\_\_DCT= 10\*log10((255^2)/MSE\_avg\_DCT);

SSIM\_avg\_DCT=SSIM\_Index(Image,Avg\_DCT);

subplot(1,3,3);

imshow(uint8(Avg\_DCT));

title([{'AVERAGE SUBSAMPLING/DCT UPSAMPLING MSE= ',MSE\_avg\_DCT,' PSNR= ',PSNR\_avg\_\_DCT}]);

Decimation\_filter=Decimation(Image);

Deci\_linear=linear\_inter(Decimation\_filter);

MSE\_deci\_linear\_inter=(sum(sum((Image-Deci\_linear).^2)))/(512\*512);

PSNR\_deci\_linear\_inter= 10\*log10((255^2)/MSE\_deci\_linear\_inter);

SSIM\_deci\_linear\_inter=SSIM\_Index(Image,Deci\_linear);

subplot(1,3,1);

imshow(uint8(Deci\_linear));

title([{'DECIMATION/LINEAR INTERPOLATION MSE= ',MSE\_deci\_linear\_inter,' PSNR',PSNR\_deci\_linear\_inter}]);

Deci\_filter=linear\_filter(Decimation\_filter);

MSE\_deci\_linear\_filter=(sum(sum((Image-Deci\_filter).^2)))/(512\*512);

PSNR\_deci\_linear\_filter= 10\*log10((255^2)/MSE\_deci\_linear\_filter);

SSIM\_deci\_linear\_filter=SSIM\_Index(Image,Deci\_filter);

subplot(1,3,2);

imshow(uint8(Deci\_filter));

title([{'DECIMATION/LINEAR FILTER MSE= ',MSE\_deci\_linear\_filter,' PSNR',PSNR\_deci\_linear\_filter}]);

Deci\_DCT=DCT\_up(Decimation\_filter);

Avg\_DCT=DCT\_up(Avg\_sub);

MSE\_deci\_DCT=(sum(sum((Image-Deci\_DCT).^2)))/(512\*512);

PSNR\_deci\_DCT= 10\*log10((255^2)/MSE\_deci\_DCT);

SSIM\_deci\_DCT=SSIM\_Index(Image,Deci\_DCT);

subplot(1,3,3);

imshow(uint8(Deci\_DCT));

title([{'DECIMATION/DCT UPSAMPLING MSE= ',MSE\_deci\_DCT,' PSNR',PSNR\_deci\_DCT}]);

sub\_DCT=DCT\_sub(Image);

subDCT\_linear=linear\_inter(sub\_DCT);

MSE\_DCT\_linear\_inter=(sum(sum((Image-subDCT\_linear).^2)))/(512\*512);

PSNR\_DCT\_linear\_inter= 10\*log10((255^2)/MSE\_DCT\_linear\_inter);

SSIM\_DCT\_linear\_inter=SSIM\_Index(Image,subDCT\_linear);

figure;

subplot(1,3,1);

imshow(uint8(subDCT\_linear));

title([{'DCT SUBSAMPLING/LINEAR INTERPOLATION MSE= ',MSE\_DCT\_linear\_inter,' PSNR',PSNR\_DCT\_linear\_inter}]);

% =========================================================================

% DCT SUBSAMPLING/LINEAR FILTER

subDCT\_filter=linear\_filter(sub\_DCT); % CALLS THE linear\_filter FUNCTION

MSE\_DCT\_linear\_filter=(sum(sum((Image-subDCT\_filter).^2)))/(512\*512);

% MSE IS CALCULATED USING THIS FORMULA

PSNR\_DCT\_linear\_filter= 10\*log10((255^2)/MSE\_DCT\_linear\_filter);

% PSNR IS CALCULATED USING THIS FORMULA

SSIM\_DCT\_linear\_filter=SSIM\_Index(Image,subDCT\_filter);

% CALLS THE SSIM\_Index FUNCTION

% PLOTTING THE DCT SUBSAMPLING/LINEAR FILTER

subplot(1,3,2);

imshow(uint8(subDCT\_filter));

title([{'DCT SUBSAMPLING/LINEAR FILTER MSE= ',MSE\_DCT\_linear\_filter,' PSNR',PSNR\_DCT\_linear\_filter}]);

% =========================================================================

% DCT SUBSAMPLING/DCT UPSAMPLING

subDCT\_DCT=DCT\_up(sub\_DCT); % CALLS THE DCT\_up FUNCTION

MSE\_DCT\_DCT=(sum(sum((Image-subDCT\_DCT).^2)))/(512\*512);

% MSE IS CALCULATED USING THIS FORMULA

PSNR\_DCT\_DCT= 10\*log10((255^2)/MSE\_DCT\_DCT);

% PSNR IS CALCULATED USING THIS FORMULA

SSIM\_DCT\_DCT=SSIM\_Index(Image,subDCT\_DCT);

% CALLS THE SSIM\_Index FUNCTION

% PLOTTING THE DCT SUBSAMPLING/DCT UPSAMPLING

subplot(1,3,3);

imshow(uint8(subDCT\_DCT));

title([{'DCT SUBSAMPLING/DCT UPSAMPLING MSE= ',MSE\_DCT\_DCT,' PSNR',PSNR\_DCT\_DCT}]);;

% =========================================================================

% SUBSAMPLING IMAGES

% PLOTTING THE 2D DCT SUBSAMPLING IMAGE

figure;

subplot(1,3,2);

imshow(uint8(sub\_DCT));

title('2D DCT SUBSAMPLING IMAGE');

% PLOTTING THE AVERAGE PIXEL LENA IMAGE

subplot(1,3,1);

imshow(uint8(Avg\_sub));

title('AVERAGE PIXEL LENA IMAGE');

% PLOTTING THE DECIMATION FILTER IMAGE

subplot(1,3,3);

imshow(uint8(Decimation\_filter));

title('DECIMATION FILTER IMAGE');

subavg.m :

% THIS IS A subavg FUNCTION WHICH WILL BE CALLED BY THE CALLING

% FUNCTION IN THE project15.m

function [fig\_avg1]=subavg(Image)

fig\_avg=zeros(512,512);

for i=1:2:512

for j=1:2:512

fig\_avg(i,j) = (Image(i,j)+Image(i,j+1)+Image(i+1,j)+Image(i+1,j+1))/4;

end

end

for i=1:256

for j=1:256

fig\_avg1(i,j) = fig\_avg(1+2\*(i-1), 1+2\*(j-1));

end

end

linear\_inter.m :

% THIS IS A linear\_inter FUNCTION WHICH WILL BE CALLED BY THE CALLING

% FUNCTION IN THE project15.m

function [linear\_up\_filter]= linear\_inter(down\_samp)

linear\_up\_filter=zeros(512,512);

k=0;

for i=1:2:512

k=k+1;l=0;

for j=1:2:512

l=l+1;

linear\_up\_filter(i,j)=down\_samp(k,l);

end

end

C=zeros(513,513);

C(1:512,1:512)=linear\_up\_filter;

for i=1:2:512

for j=2:2:512

linear\_up\_filter(i,j)=(C(i,j-1)+C(i,j+1))/2;

end

end

C(1:512,1:512)=linear\_up\_filter;

for i=2:2:512

for j=1:512

linear\_up\_filter(i,j)=(C(i-1,j)+C(i+1,j))/2;

end

end

linear\_filter.m :

% THIS IS A linear\_filter FUNCTION WHICH WILL BE CALLED BY THE CALLING

% FUNCTION IN THE project15.m

function [up\_linear\_filter]=linear\_filter(down\_samp)

B=zeros(256,512);

B(1:256,1:2:512)=down\_samp;

C=zeros(256,517);

filter = [-12 0 140 256 140 0 -12];

C(1:256,3:514)=B;

C(1:256,1)=B(1:256,3);

C(1:256,2)=B(1:256,2);

C(1:256,515)=B(1:256,511);

C(1:256,516)=B(1:256,510);

C(1:256,517)=B(1:256,509);

a=0;

for i=1:256

a=a+1;b=0;

for j=4:2:515

b=b+2;

B(a,b)=((C(i,j-3)\*filter(1,1))+(C(i,j-2)\*filter(1,2))+(C(i,j-1)\*filter(1,3))+(filter(1,4)\*C(i,j))+(filter(1,5)\*C(i,j+1))+(filter(1,6)\*C(i,j+2))+(filter(1,7)\*C(i,j+3)))/256;

end

end

C=zeros(517,512);

up\_linear\_filter=zeros(512,512);

up\_linear\_filter(1:2:512,1:512)=B;

up\_linear\_filter(2:2:512,1:512)=zeros;

C(3:2:514,1:512)=B;

Filter\_v=filter';

C(1,1:512)=up\_linear\_filter(3,1:512);

C(2,1:512)=up\_linear\_filter(2,1:512);

C(515,1:512)=up\_linear\_filter(511,1:512);

C(516,1:512)=up\_linear\_filter(510,1:512);

C(517,1:512)=up\_linear\_filter(509,1:512);

a=0;

for i=4:2:515

a=a+2;b=0;

for j=1:512

b=b+1;

up\_linear\_filter(a,b)=((C(i-3,j)\*Filter\_v(1,1))+(C(i-2,j)\*Filter\_v(2,1))+(C(i-1,j)\*Filter\_v(3,1))+(Filter\_v(4,1)\*C(i,j))+(Filter\_v(5,1)\*C(i+1,j))+(Filter\_v(6,1)\*C(i+2,j))+(Filter\_v(7,1)\*C(i+3,j)))/256;

end

end

Decimation.m :

% THIS IS A Decimation FUNCTION WHICH WILL BE CALLED BY THE CALLING

% FUNCTION IN THE project15.m

function [decimation\_filter]=Decimation(img)

B=zeros(512,518);

B(1:512,4:515)=img;

a=4;

for j=1:3

B(1:512,j)=img(1:512,a);

a=a-1;

end

a=514;

for j=516:518

B(1:512,j)=B(1:512,a);

a=a-1;

end

filter=[-1 0 9 16 9 0 -1];

a=0;

for i=1:512

a=a+1;b=0;

for j=4:515

b=b+1;

A(a,b)=((B(i,j-3)\*filter(1,1))+(B(i,j-2)\*filter(1,2))+(B(i,j-1)\*filter(1,3))+(B(i,j)\*filter(1,4))+(B(i,j+1)\*filter(1,5))+(B(i,j+2)\*filter(1,6))+(B(i,j+3)\*filter(1,7)))/32;

end

end

B=zeros(518,512);

Filter\_v=filter';

B(4:515,1:512)=A(1:512,1:512);

B(1,1:512)=A(4,1:512);

B(2,1:512)=A(3,1:512);

B(3,1:512)=A(2,1:512);

B(516,1:512)=A(511,1:512);

B(517,1:512)=A(510,1:512);

B(518,1:512)=A(509,1:512);

a=0;

for i=4:515

a=a+1;b=0;

for j=1:512

b=b+1;

A(a,b)=((B(i-3,j)\*Filter\_v(1,1))+(B(i-2,j)\*Filter\_v(2,1))+(B(i-1,j)\*Filter\_v(3,1))+(B(i,j)\*Filter\_v(4,1))+(B(i+1,j)\*Filter\_v(5,1))+(B(i+2,j)\*Filter\_v(6,1))+(B(i+3,j)\*Filter\_v(7,1)))/32;

end

end

decimation\_filter=zeros(256,256);

a=0;

for i=1:2:512

b=0;a=a+1;

for j=1:2:512

b=b+1;

decimation\_filter(a,b)=A(i,j);

end

end

DCT\_up.m :

% THIS IS A DCT\_up FUNCTION WHICH WILL BE CALLED BY THE CALLING

% FUNCTION IN THE project15.m

function [up\_DCT]=DCT\_up(down\_samp)

img\_DCT=dct2(down\_samp);

Image=zeros(512,512);

Image(1:256,1:256)=img\_DCT;

up\_DCT=idct2(Image);

DCT\_sub.m :

% THIS IS A DCT\_sub FUNCTION WHICH WILL BE CALLED BY THE CALLING

% FUNCTION IN THE project15.m

function [DCT\_down]=DCT\_sub(img)

for i=1:16:512

for j=1:16:512

Img\_dct(i:i+7, j:j+7) = dct2(img(i:i+7, j:j+7));

Img\_dct(i:i+7, j+8:j+15) = dct2(img(i:i+7, j+8:j+15));

Img\_dct(i+8:i+15, j:j+7) = dct2(img(i+8:i+15, j:j+7));

Img\_dct(i+8:i+15, j+8:j+15) = dct2(img(i+8:i+15, j+8:j+15));

end

end

for i=1:16:512

for j=1:16:512

i1=(i/2)+0.5;

j1=(j/2)+0.5;

DCT\_down(i1:i1+3, j1:j1+3) = idct2(Img\_dct(i:i+3, j:j+3));

DCT\_down(i1:i1+3, j1+4:j1+7) = idct2(Img\_dct(i:i+3, j+8:j+11));

DCT\_down(i1+4:i1+7, j1:j1+3) = idct2(Img\_dct(i+8:i+11, j:j+3));

DCT\_down(i1+4:i1+7, j1+4:j1+7) = idct2(Img\_dct(i+8:i+11, j+8:j+11));

end

end

SSIM\_Index.m ;

if (nargin < 2 || nargin > 5)

ssim\_index = -Inf;

ssim\_map = -Inf;

return;

end

if (size(img1) ~= size(img2))

ssim\_index = -Inf;

ssim\_map = -Inf;

return;

end

[M N] = size(img1);

if (nargin == 2)

if ((M < 11) || (N < 11))

ssim\_index = -Inf;

ssim\_map = -Inf;

return

end

window = fspecial('gaussian', 11, 1.5); %

K(1) = 0.01; % default settings

K(2) = 0.03; %

L = 255; %

end

if (nargin == 3)

if ((M < 11) || (N < 11))

ssim\_index = -Inf;

ssim\_map = -Inf;

return

end

window = fspecial('gaussian', 11, 1.5);

L = 255;

if (length(K) == 2)

if (K(1) < 0 || K(2) < 0)

ssim\_index = -Inf;

ssim\_map = -Inf;

return;

end

else

ssim\_index = -Inf;

ssim\_map = -Inf;

return;

end

end

if (nargin == 4)

[H W] = size(window);

if ((H\*W) < 4 || (H > M) || (W > N))

ssim\_index = -Inf;

ssim\_map = -Inf;

return

end

L = 255;

if (length(K) == 2)

if (K(1) < 0 || K(2) < 0)

ssim\_index = -Inf;

ssim\_map = -Inf;

return;

end

else

ssim\_index = -Inf;

ssim\_map = -Inf;

return;

end

end

if (nargin == 5)

[H W] = size(window);

if ((H\*W) < 4 || (H > M) || (W > N))

ssim\_index = -Inf;

ssim\_map = -Inf;

return

end

if (length(K) == 2)

if (K(1) < 0 || K(2) < 0)

ssim\_index = -Inf;

ssim\_map = -Inf;

return;

end

else

ssim\_index = -Inf;

ssim\_map = -Inf;

return;

end

end

C1 = (K(1)\*L)^2;

C2 = (K(2)\*L)^2;

window = window/sum(sum(window));

img1 = double(img1);

img2 = double(img2);

mu1 = filter2(window, img1, 'valid');

mu2 = filter2(window, img2, 'valid');

mu1\_sq = mu1.\*mu1;

mu2\_sq = mu2.\*mu2;

mu1\_mu2 = mu1.\*mu2;

sigma1\_sq = filter2(window, img1.\*img1, 'valid') - mu1\_sq;

sigma2\_sq = filter2(window, img2.\*img2, 'valid') - mu2\_sq;

sigma12 = filter2(window, img1.\*img2, 'valid') - mu1\_mu2;

if (C1 > 0 & C2 > 0)

ssim\_map = ((2\*mu1\_mu2 + C1).\*(2\*sigma12 + C2))./((mu1\_sq + mu2\_sq + C1).\*(sigma1\_sq + sigma2\_sq + C2));

else

numerator1 = 2\*mu1\_mu2 + C1;

numerator2 = 2\*sigma12 + C2;

denominator1 = mu1\_sq + mu2\_sq + C1;

denominator2 = sigma1\_sq + sigma2\_sq + C2;

ssim\_map = ones(size(mu1));

index = (denominator1.\*denominator2 > 0);

ssim\_map(index) = (numerator1(index).\*numerator2(index))./(denominator1(index).\*denominator2(index));

index = (denominator1 ~= 0) & (denominator2 == 0);

ssim\_map(index) = numerator1(index)./denominator1(index);

end

mssim = mean2(ssim\_map);

OUTPUT:

****

****

****

****

****