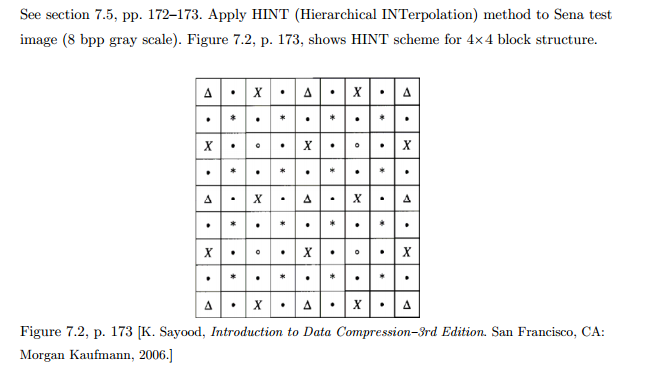
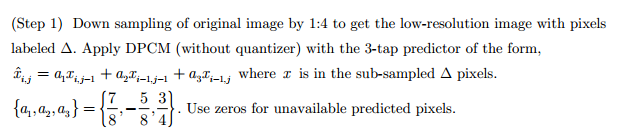
EE5351\_DIGITAL VIDEO CODING

ASSIGNMENT\_HINT

NAME: PAVAI ARCHIMEDES

ST ID: 1001233996





PROGRAM:

img = fopen('sena.img', 'r');

O\_Img = fread(img,[256,256]);

O\_Img=O\_Img';

E=entropy(uint8(O\_Img));

[c,x]=imhist(uint8(O\_Img));

figure(1);

subplot(3,3,1);

stem(x,c);

title('Histogram- Sena image');

O\_Img1=O\_Img;

delta\_pix(64,64)=0;

for i\_delta=1:64

for j\_delta=1:64

delta\_pix(i\_delta, j\_delta)= O\_Img(1+4\*(i\_delta-1), 1+4\*(j\_delta-1));

end

end

delta\_pix1(64,64)=delta\_pix(64,64);

for i\_delta=2:64

for j\_delta=2:64

delta\_pix1(i\_delta, j\_delta)= (7/8)\*delta\_pix(i\_delta, j\_delta-1)- (5/8)\*delta\_pix(i\_delta-1, j\_delta-1)+ (3/4)\*delta\_pix(i\_delta-1, j\_delta) ;

end

end

err=abs(delta\_pix-delta\_pix1);

E1=entropy(uint8(err));

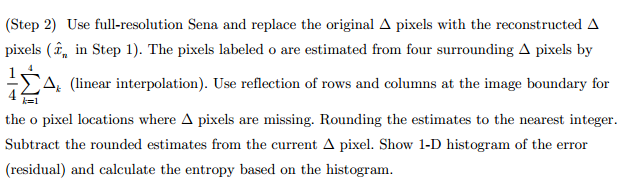
[c1,x1]=imhist(uint8(err));

figure(1);

subplot(3,3,2);

stem(x1,c1);

title('Histogram- Delta pixel errors');



for i\_delta=1:64

for j\_delta=1:64

O\_Img1(1+4\*(i\_delta-1), 1+4\*(j\_delta-1))= delta\_pix1(i\_delta, j\_delta);

end

end

O\_Img1(257,1:256)=O\_Img1(256,1:256);

O\_Img1(1:256,257)=O\_Img1(1:256,256);

O\_Img1(257,257)=O\_Img1(256,256);

O\_Img1(258,1:257)=O\_Img1(257,1:257);

O\_Img1(1:257,258)=O\_Img1(1:257,257);

O\_Img1(258,258)=O\_Img1(257,257);

O\_Img1(259,1:258)=O\_Img1(258,1:258);

O\_Img1(1:258,259)=O\_Img1(1:258,258);

O\_Img1(259,259)=O\_Img1(258,258);

O\_Img\_org=O\_Img1;

for i\_o=1:64

for j\_o=1:64

O\_Img1(3+4\*(i\_o-1), 3+4\*(j\_o-1)) = round((O\_Img1(1+4\*(i\_o-1), 1+4\*(j\_o-1))+ O\_Img1(1+4\*(i\_o-1), 1+4\*(j\_o))+O\_Img1(1+4\*(i\_o), 1+4\*(j\_o-1))+ O\_Img1(1+4\*(i\_o), 1+4\*(j\_o)))/4);

end

end

for i\_o=1:64

for j\_o=1:64

o\_64(i\_o,j\_o)= O\_Img1(3+4\*(i\_o-1), 3+4\*(j\_o-1));

end

end

err2=abs(delta\_pix1-o\_64);

E2=entropy(uint8(err2));

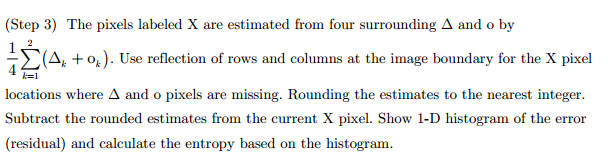
[c2,x2]=imhist(uint8(err2));

figure(1);

subplot(3,3,3);

stem(x2,c2);

title('Histogram- O pixel errors');



OImg\_x(3:261,3:261)= O\_Img1(1:259, 1:259);

OImg\_x(1:2,1:2)=O\_Img1(1:2,1:2);

OImg\_x(262:267,262:267) = OImg\_x(256:261,256:261);

f=1;

for i\_x=3:66

for j\_x=3:66

if(f~=0)

OImg\_x(3+4\*(i\_x-1),5+4\*(j\_x-1))= round((OImg\_x(3+4\*(i\_x-1),3+4\*(j\_x-1))+ OImg\_x(3+4\*(i\_x-1),7+4\*(j\_x-1))+OImg\_x(1+4\*(i\_x-1),3+4\*(j\_x-1))+ OImg\_x(5+4\*(i\_x-1),7+4\*(j\_x-1)))/4);

else

OImg\_x(5+4\*(i\_x-1),3+4\*(j\_x-1))= round((OImg\_x(5+4\*(i\_x-1),1+4\*(j\_x-1))+OImg\_x(5+4\*(i\_x-1),5+4\*(j\_x-1))+OImg\_x(3+4\*(i\_x-1),3+4\*(j\_x-1))+ OImg\_x(5+4\*(i\_x-1),3+4\*(j\_x-1)))/4);

end

end

f = ~f;

end

f=1;

for i\_x=1:64

for j\_x=1:64

if(f~=0)

OImg\_x64(i\_x,j\_x)= OImg\_x(1+4\*(i\_x-1), 3+4\*(j\_x-1));

else

OImg\_x64(i\_x,j\_x)= OImg\_x(3+4\*(i\_x-2), 1+4\*(j\_x-1));

end

end

f = ~f;

end

f=1;

for i\_x=1:64

for j\_x=1:64

if(f~=0)

OImg\_64(i\_x,j\_x)= O\_Img1(1+4\*(i\_x-1), 3+4\*(j\_x-1));

else

OImg\_64(i\_x,j\_x)= O\_Img1(3+4\*(i\_x-2), 1+4\*(j\_x-1));

end

end

f = ~f;

end

err3=abs(OImg\_x64-OImg\_64);

E3=entropy(uint8(err3));

[c3,x3]=imhist(uint8(err3));

figure(1);

subplot(3,3,4);

stem(x3,c3);

title('Histogram- X pixel errors');

f=1;

for i\_x=1:64

for j\_x=1:64

if(f~=0)

O\_Img1(1+4\*(i\_x-1), 3+4\*(j\_x-1))= OImg\_x64(i\_x, j\_x);

else

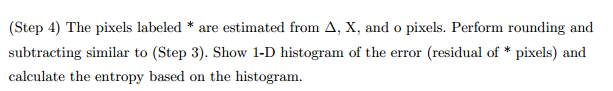
O\_Img1(3+4\*(i\_x-2), 1+4\*(j\_x-1))= OImg\_x64(i\_x, j\_x);

end

end

f = ~f;

end



OImg\_s(1:257,1:257)=O\_Img1(1:257,1:257);

for i\_ast=1:128

for j\_ast=1:128

OImg\_s(2\*i\_ast,2\*j\_ast)= round((OImg\_s(2\*i\_ast-1,2\*j\_ast-1)+OImg\_s(2\*i\_ast-1,2\*j\_ast+1)+OImg\_s(2\*i\_ast+1,2\*j\_ast-1)+OImg\_s(2\*i\_ast+1,2\*j\_ast+1))/4);

end

end

for i\_ast=1:128

for j\_ast=1:128

P\_s(i\_ast,j\_ast)=OImg\_s(2\*i\_ast,2\*j\_ast);

OImg\_sf(i\_ast,j\_ast)=O\_Img1(2\*i\_ast,2\*j\_ast);

end

end

err4=abs(OImg\_sf-P\_s);

E4=entropy(uint8(err4));

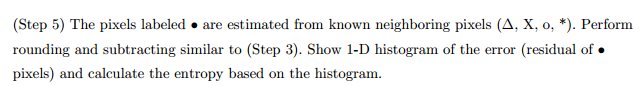
[c4,x4]=imhist(uint8(err4));

figure(1);

subplot(3,3,5);

stem(x4,c4);

title('Histogram- \* pixel errors');



OImg\_d(2:257,2:257)=O\_Img1(1:256, 1:256);

OImg\_d(1:256,1)=OImg\_d(2:257,1);

OImg\_d(1,1:256)=OImg\_d(1,1:256);

OImg\_d(257,1)=OImg\_d(256,1);

OImg\_d(1,257)=OImg\_d(1,256);

OImg\_d(258,1:257)=OImg\_d(257,1:257);

OImg\_d(1:257,258)=OImg\_d(1:257,257);

OImg\_d(258,258)=OImg\_d(257,257);

OImg\_d(259,1:258)=OImg\_d(258,1:258);

OImg\_d(1:258,259)=OImg\_d(1:258,258);

OImg\_d(259,259)=OImg\_d(258,258);

f=1;

for i\_d=1:256

for j\_d=1:128

if(f~=0)

OImg\_d(2+(i\_d-1),3+2\*(j\_d-1))= round((OImg\_d(1+(i\_d-1),2+2\*(j\_d-1))+OImg\_d(1+(i\_d-1),3+2\*(j\_d-1))+OImg\_d(3+(i\_d-1),2+2\*(j\_d-1))+OImg\_d(3+(i\_d-1),4+2\*(j\_d-1)))/4);

else

OImg\_d(3+(i\_d-1),2+2\*(j\_d-1))= round((OImg\_d(2+(i\_d-1),1+2\*(j\_d-1))+OImg\_d(2+(i\_d-1),3+2\*(j\_d-1))+OImg\_d(4+(i\_d-1),1+2\*(j\_d-1))+OImg\_d(4+(i\_d-1),3+2\*(j\_d-1)))/4);

end

end

f = ~f;

end

f=1;

for i\_d=1:256

for j\_d=1:128

if(f~=0)

OImg\_df(i\_d,j\_d)=OImg\_d(2+(i\_d-1),3+2\*(j\_d-1));

else

OImg\_df(i\_d,j\_d)=OImg\_d(3+(i\_d-1),2+2\*(j\_d-1));

end

end

f = ~f;

end

f=1;

for i\_d=1:256

for j\_d=1:128

if(f~=0)

OImg\_v(i\_d,j\_d)=O\_Img1(2+(i\_d-1),3+2\*(j\_d-1));

else

OImg\_v(i\_d,j\_d)=O\_Img1(3+(i\_d-1),2+2\*(j\_d-1));

end

end

f = ~f;

end

err5=abs(OImg\_df-OImg\_v);

E5=entropy(uint8(err5));

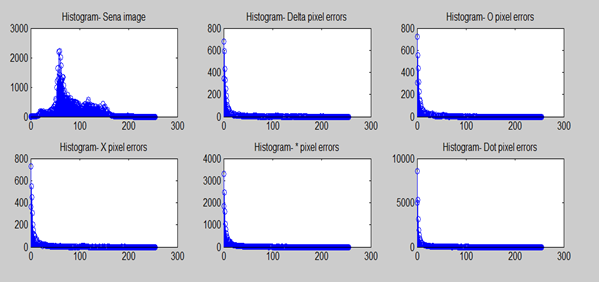
[c5,x5]=imhist(uint8(err5));

figure(1);

subplot(3,3,6);

stem(x5,c5);

title('Histogram- Dot pixel errors');



Conclusion:

This results gives pyramid like representation of image with each layer of pyramid serving as prediction model for the layer below it and technique such is HINT-Hierarchical Interpolation.