**Sourabh Tirodkar**

**EE569- Digital Image Processing**

**3589406164**

**Codes:**

Problem 1A : Sobel Edge Detector

% Sourabh Tirodkar

% 3589406164

% tirodkar@usc.edu

% Submission Date- 16th Feb,20

% PROBLEM 1A

% SOBEL EDGE DETECTOR

% DOG IMAGE

N=481;

M=321;

%I = readraw\_color('C:\Users\user\Desktop\USC Classes\Spring 2020\EE569- Digital Image Processing\MATLAB Files\EE569\_Spring\_2020\_HW2\_Materials\Problem1\Dogs.raw',N,M);

I = imread('C:\Users\user\Desktop\USC Classes\Spring 2020\EE569- Digital Image Processing\Homeworks\HW #2\EE569\_Spring\_2020\_HW2\_Materials\Problem1\Dogs.jpg');

I=double(I);

figure(1)

imshow(uint8(I));

%[row col]= size(I);

% row=321;

% col=481;

%%padding

% II=zeros(row+2,col+2,3);

% for i=1:1:row

% for j=1:1:col

% II(i+1,j+1)=I(i,j);

% end

% end

%

% for j=1:1:col

% II(1,j+1)=I(1,j);

% II(row+2,j+1)=I(row,j);

% end

%

% for i=1:1:row+2

% II(i,1)=II(i,2);

% II(i,col+2)=II(i,col+1);

% end

R=I(:,:,1);

G=I(:,:,2);

B=I(:,:,3);

Y=0.2989\*R + 0.5870\*G + 0.1140\*B;

figure(2)

imshow(uint8(Y))

for i=2:1:320

for j=2:1:480

Gx(i,j)=(-1\*Y(i-1,j-1)+ 0 + 1\*Y(i-1,j+1) -2\*Y(i,j-1)+ 0 + 2\*Y(i,j+1) -1\*Y(i+1,j-1)+ 0 + 1\*Y(i+1,j+1));

end

end

for j=2:1:480

for i=2:1:320

Gy(i,j)=(1\*Y(i-1,j-1)+ 2\*Y(i-1,j) + 1\*Y(i-1,j+1)+ 0 + 0+ 0 -1\*Y(i+1,j-1) -2\*Y(i+1,j)-1\*Y(i+1,j+1));

end

end

YY=zeros(320,480);

YY=sqrt((Gx).^2+(Gy).^2);

for i=2:1:320

for j=2:1:480

Gx1(i,j)= 255\*(Gx(i,j) - min(Gx))/(max(Gx) - min(Gx));

Gy1(i,j)= 255\*(Gy(i,j) - min(Gy))/(max(Gy) - min(Gy));

YY1(i,j)= 255\*(YY(i,j) - min(YY))/(max(YY) - min(YY));

end

end

% X gradient

figure(3)

imshow(uint8(Gx1))

% Y gradient

figure(4)

imshow(uint8(Gy1))

% gradient magnitude

figure(5)

imshow(uint8(YY))

%%

T=80;

% thre=0.85;

% abc=thre\*(320\*480);

% abc=count;

count=0;

for i=1:1:320

for j=1:1:480

%0.85\*(320\*480)/100

if YY1(i,j)<T

zz(i,j)=255;

count=count+1;

else

zz(i,j)=0;

end

end

end

figure(6)

imshow(uint8(zz))

thre=(count/(320\*480))\*100;

title("Thresholding to invert the image- Pixel val:60, Threshold %="+(thre));

% Sourabh Tirodkar

% 3589406164

% tirodkar@usc.edu

% Submission Date- 16th Feb,20

% PROBLEM 1A

% SOBEL EDGE DETECTOR

% GALLERY IMAGE

N=481;

M=321;

%I = readraw\_color('C:\Users\user\Desktop\USC Classes\Spring 2020\EE569- Digital Image Processing\MATLAB Files\EE569\_Spring\_2020\_HW2\_Materials\Problem1\Dogs.raw',N,M);

I = imread('C:\Users\user\Desktop\USC Classes\Spring 2020\EE569- Digital Image Processing\Homeworks\HW #2\EE569\_Spring\_2020\_HW2\_Materials\Problem1\gallery.jpg');

I=double(I);

figure(1)

imshow(uint8(I));

%[row col]= size(I);

% row=321;

% col=481;

%%padding

% II=zeros(row+2,col+2,3);

% for i=1:1:row

% for j=1:1:col

% II(i+1,j+1)=I(i,j);

% end

% end

%

% for j=1:1:col

% II(1,j+1)=I(1,j);

% II(row+2,j+1)=I(row,j);

% end

%

% for i=1:1:row+2

% II(i,1)=II(i,2);

% II(i,col+2)=II(i,col+1);

% end

%%

R=I(:,:,1);

G=I(:,:,2);

B=I(:,:,3);

Y=0.2989\*R + 0.5870\*G + 0.1140\*B;

figure(2)

imshow(uint8(Y))

for i=2:1:320

for j=2:1:480

Gx(i,j)=(-1\*Y(i-1,j-1)+ 0 + 1\*Y(i-1,j+1) -2\*Y(i,j-1)+ 0 + 2\*Y(i,j+1) -1\*Y(i+1,j-1)+ 0 + 1\*Y(i+1,j+1));

end

end

for j=2:1:480

for i=2:1:320

Gy(i,j)=(1\*Y(i-1,j-1)+ 2\*Y(i-1,j) + 1\*Y(i-1,j+1)+ 0 + 0+ 0 -1\*Y(i+1,j-1) -2\*Y(i+1,j)-1\*Y(i+1,j+1));

end

end

YY=zeros(320,480);

YY=sqrt((Gx).^2+(Gy).^2);

for i=2:1:320

for j=2:1:480

Gx1(i,j)= 255\*(Gx(i,j) - min(Gx))/(max(Gx) - min(Gx));

Gy1(i,j)= 255\*(Gy(i,j) - min(Gy))/(max(Gy) - min(Gy));

YY1(i,j)= 255\*(YY(i,j) - min(YY))/(max(YY) - min(YY));

end

end

% X gradient

figure(3)

imshow(uint8(Gx1))

% Y gradient

figure(4)

imshow(uint8(Gy1))

% gradient magnitude

figure(5)

imshow(uint8(YY))

T=50;

count=0;

for i=1:1:320

for j=1:1:480

%0.85\*(320\*480)/100

if YY1(i,j)<T

zz(i,j)=255;

count=count+1;

else

zz(i,j)=0;

end

end

end

figure(6)

imshow(uint8(zz))

thre=(count/(320\*480))\*100;

title("Thresholding to invert the image- Pixel val:60, Threshold %="+(thre));

Problem 1B : Canny Edge Detector

Source Code

Problem 1 C : Structured Edge

Source Code

Problem 1 D : Performance Evaluation

Source Code

Problem 2 A : Dithering

1. Fixed Thresholding

% Sourabh Tirodkar

% 3589406164

% tirodkar@usc.edu

% Submission Date- 16th Feb,20

% PROBLEM 2A\_1

% HALFTONING- DITHERING

% LIGHT HOUSE IMAGE

% FIXED THRESHOLDING

%Input image

N=750;

M=500;

I = readraw\_grey('C:\Users\user\Desktop\USC Classes\Spring 2020\EE569- Digital Image Processing\Homeworks\HW #2\EE569\_Spring\_2020\_HW2\_Materials\Problem2\LightHouse.raw', N,M);

figure(1)

imshow(uint8(I));

I = double (I);

T=128;

for i=1:1:500

for j=1:1:750

if I(i,j)<T

II(i,j)=0;

else

II(i,j)=255;

end

end

end

figure(2)

imshow(uint8(II))

title('Fixed Thresholding, T=128');

1. Random Thresholding

% Sourabh Tirodkar

% 3589406164

% tirodkar@usc.edu

% Submission Date- 16th Feb,20

% PROBLEM 2A\_1

% HALFTONING- DITHERING

% LIGHT HOUSE IMAGE

% RANDOM THRESHOLDING

%Input image

N=750;

M=500;

I = readraw\_grey('C:\Users\user\Desktop\USC Classes\Spring 2020\EE569- Digital Image Processing\Homeworks\HW #2\EE569\_Spring\_2020\_HW2\_Materials\Problem2\LightHouse.raw', N,M);

figure(1)

imshow(uint8(I));

I = double (I);

T=rand(M,N)\*255;

for i=1:1:500

for j=1:1:750

if I(i,j)<T(i,j)

II(i,j)=0;

else

II(i,j)=255;

end

end

end

figure(2)

imshow(uint8(II))

title('Random Thresholding');

1. Dithering Matrix

% Sourabh Tirodkar

% 3589406164

% tirodkar@usc.edu

% Submission Date- 16th Feb,20

% PROBLEM 2A\_1

% HALFTONING- DITHERING

% LIGHT HOUSE IMAGE

% DITHERING MATRIX

%Input image

N=750;

M=500;

I = readraw\_grey('C:\Users\user\Desktop\USC Classes\Spring 2020\EE569- Digital Image Processing\Homeworks\HW #2\EE569\_Spring\_2020\_HW2\_Materials\Problem2\LightHouse.raw', N,M);

figure(1)

imshow(uint8(I));

I = double (I);

n2=2;

n8=8;

n16=16;

n32=32;

D2=[1 2;3 0];

D4=[4\*D2+1 4\*D2+2;4\*D2+3 4\*D2];

D8=[4\*D4+1 4\*D4+2;4\*D4+3 4\*D4];

D16=[4\*D8+1 4\*D8+2;4\*D8+3 4\*D8];

D32=[4\*D16+1 4\*D16+2;4\*D16+3 4\*D16];

T2=((D2+0.5)/(n2^2))\*255;

T8=((D8+0.5)/(n8^2))\*255;

T16=((D16+0.5)/(n16^2))\*255;

T32=((D32+0.5)/(n32^2))\*255;

for i=1:1:500

for j=1:1:750

if I(i,j)<T2(1+mod(i,n2),1+mod(j,n2))

II(i,j)=0;

else

II(i,j)=255;

end

end

end

figure(2)

imshow((II))

title('Dithering Matrix-2');

for i=1:1:500

for j=1:1:750

if I(i,j)<T8(1+mod(i,n8),1+mod(j,n8))

II(i,j)=0;

else

II(i,j)=255;

end

end

end

figure(3)

imshow((II))

title('Dithering Matrix-8');

for i=1:1:500

for j=1:1:750

if I(i,j)<T32(1+mod(i,n32),1+mod(j,n32))

II(i,j)=0;

else

II(i,j)=255;

end

end

end

figure(4)

imshow((II))

title('Dithering Matrix-32');

Problem 2 B : Error Diffusion

% Sourabh Tirodkar

% 3589406164

% tirodkar@usc.edu

% Submission Date- 16th Feb,20

% PROBLEM 2B

% HALFTONING- ERROR DIFFUSION

% LIGHT HOUSE IMAGE

% FLOYD, JJN AND STUCKI

N=750;

M=500;

I = readraw\_grey('C:\Users\user\Desktop\USC Classes\Spring 2020\EE569- Digital Image Processing\Homeworks\HW #2\EE569\_Spring\_2020\_HW2\_Materials\Problem2\LightHouse.raw', N,M);

figure(1)

imshow(uint8(I));

I = double(I);

%copying image into f

f=I;

%Floyd Steinberg Matrix

h=(1/16)\*[0 0 0;0 0 7;3 5 1];

h1=(1/16)\*[0 0 0;7 0 0;1 5 3];

T=128;

row=500;

col=750;

for i = 1 : row

% odd row

if mod(i,2) ~= 0

for j = 1 : col

if f(i,j)>T

b(i,j)=255;

else

b(i,j)=0;

end

e = f(i,j) - b(i,j);

if j==1

f(i,j+1)=f(i,j+1)+(7/16)\*e;

f(i+1,j+1)=f(i+1,j+1)+(1/16)\*e;

f(i+1,j)=f(i+1,j)+(5/16)\*e;

elseif j==col

f(i+1,j-1)=f(i+1,j-1)+(3/16)\*e;

f(i+1,j)=f(i+1,j)+(5/16)\*e;

else

f(i,j+1)=f(i,j+1)+(7/16)\*e;

f(i+1,j-1)=f(i+1,j-1)+(3/16)\*e;

f(i+1,j)=f(i+1,j)+(5/16)\*e;

f(i+1,j+1)=f(i+1,j+1)+(1/16)\*e;

end

end

end

% even row

if mod(i,2) == 0

if i~=row

for j = col:-1:1

if f(i,j)>T

b(i,j)=255;

else

b(i,j)=0;

end

e = f(i,j) - b(i,j);

if j==col

f(i,j-1)=f(i,j-1)+(7/16)\*e;

f(i+1,j-1)=f(i+1,j-1)+(1/16)\*e;

f(i+1,j)=f(i+1,j)+(5/16)\*e;

elseif j==1

f(i+1,j+1)=f(i+1,j+1)+(3/16)\*e;

f(i+1,j)=f(i+1,j)+(5/16)\*e;

else

f(i,j-1)=f(i,j-1)+(7/16)\*e;

f(i+1,j-1)=f(i+1,j-1)+(1/16)\*e;

f(i+1,j)=f(i+1,j)+(5/16)\*e;

f(i+1,j+1)=f(i+1,j+1)+(3/16)\*e;

end

end

end

end

end

figure(2);

imshow(uint8(b))

title('Floyd - Steinberg Algorithm')

%JJN

%copying image into f

f=I;

b=zeros(row,col);

for i=1:row

if(mod(i,2)~=0)

if i~= (row-1)

for j=1:col

if (f(i,j)>T)

b(i,j)=255;

else

b(i,j)=0;

end

e=f(i,j)-b(i,j);

if j==1

f(i,j+1)=f(i,j+1)+(7/48)\*e;

f(i,j+2)=f(i,j+2)+(5/48)\*e;

f(i+1,j+1)=f(i+1,j+1)+(5/48)\*e;

f(i+1,j+2)=f(i+1,j+2)+(3/48)\*e;

f(i+2,j+1)=f(i+2,j+1)+(3/48)\*e;

f(i+2,j+2)=f(i+2,j+2)+(1/48)\*e;

f(i+1,j)=f(i+1,j)+(7/48)\*e;

f(i+2,j)=f(i+2,j)+(5/48)\*e;

elseif j==2

f(i,j+1)=f(i,j+1)+(7/48)\*e;

f(i,j+2)=f(i,j+2)+(5/48)\*e;

f(i+1,j)=f(i+1,j)+(7/48)\*e;

f(i+2,j)=f(i+2,j)+(5/48)\*e;

f(i+1,j+1)=f(i+1,j+1)+(5/48)\*e;

f(i+1,j+2)=f(i+1,j+2)+(3/48)\*e;

f(i+2,j+1)=f(i+2,j+1)+(3/48)\*e;

f(i+2,j+2)=f(i+2,j+2)+(1/48)\*e;

f(i+1,j-1)=f(i+1,j-1)+(5/48)\*e;

f(i+2,j-1)=f(i+2,j-1)+(3/48)\*e;

elseif j==col

f(i+1,j-1)=f(i+1,j-1)+(5/48)\*e;

f(i+1,j-2)=f(i+1,j-2)+(3/48)\*e;

f(i+2,j-1)=f(i+2,j-1)+(3/48)\*e;

f(i+2,j-2)=f(i+2,j-2)+(1/48)\*e;

f(i+1,j)=f(i+1,j)+(7/48)\*e;

f(i+2,j)=f(i+2,j)+(5/48)\*e;

elseif j==col-1

f(i+1,j-1)=f(i+1,j-1)+(5/48)\*e;

f(i+1,j-2)=f(i+1,j-2)+(3/48)\*e;

f(i+2,j-1)=f(i+2,j-1)+(3/48)\*e;

f(i+2,j-2)=f(i+2,j-2)+(1/48)\*e;

f(i+1,j)=f(i+1,j)+(7/48)\*e;

f(i+2,j)=f(i+2,j)+(5/48)\*e;

f(i,j+1)=f(i,j+1)+(7/48)\*e;

f(i+1,j+1)=f(i+1,j+1)+(5/48)\*e;

f(i+2,j+1)=f(i+2,j+1)+(3/48)\*e;

else

f(i,j+1)=f(i,j+1)+(7/48)\*e;

f(i,j+2)=f(i,j+2)+(5/48)\*e;

f(i+1,j)=f(i+1,j)+(7/48)\*e;

f(i+2,j)=f(i+2,j)+(5/48)\*e;

f(i+1,j+1)=f(i+1,j+1)+(5/48)\*e;

f(i+1,j+2)=f(i+1,j+2)+(3/48)\*e;

f(i+2,j+1)=f(i+2,j+1)+(3/48)\*e;

f(i+2,j+2)=f(i+2,j+2)+(1/48)\*e;

f(i+1,j-1)=f(i+1,j-1)+(5/48)\*e;

f(i+1,j-2)=f(i+1,j-2)+(3/48)\*e;

f(i+2,j-1)=f(i+2,j-1)+(3/48)\*e;

f(i+2,j-2)=f(i+2,j-2)+(1/48)\*e;

end

end

else

for j=col:-1:1

if (f(i,j)>T)

b(i,j)=255;

else

b(i,j)=0;

end

e=f(i,j)-b(i,j);

if j==1

f(i,j+1)=f(i,j+1)+(7/48)\*e;

f(i,j+2)=f(i,j+2)+(5/48)\*e;

f(i+1,j)=f(i+1,j)+(7/48)\*e;

f(i+1,j+1)=f(i+1,j+1)+(5/48)\*e;

f(i+1,j+2)=f(i+1,j+2)+(3/48)\*e;

elseif j==2

f(i,j+1)=f(i,j+1)+(7/48)\*e;

f(i,j+2)=f(i,j+2)+(5/48)\*e;

f(i+1,j)=f(i+1,j)+(7/48)\*e;

f(i+1,j+1)=f(i+1,j+1)+(5/48)\*e;

f(i+1,j+2)=f(i+1,j+2)+(3/48)\*e;

f(i+1,j-1)=f(i+1,j-1)+(5/48)\*e;

elseif j==col

f(i+1,j-1)=f(i+1,j-1)+(5/48)\*e;

f(i+1,j-2)=f(i+1,j-2)+(3/48)\*e;

f(i+1,j)=f(i+1,j)+(7/48)\*e;

elseif j==col-1

f(i+1,j-1)=f(i+1,j-1)+(5/48)\*e;

f(i+1,j-2)=f(i+1,j-2)+(3/48)\*e;

f(i+1,j)=f(i+1,j)+(7/48)\*e;

f(i,j+1)=f(i,j+1)+(7/48)\*e;

f(i+1,j+1)=f(i+1,j+1)+(5/48)\*e;

else

f(i,j+1)=f(i,j+1)+(7/48)\*e;

f(i,j+2)=f(i,j+2)+(5/48)\*e;

f(i+1,j)=f(i+1,j)+(7/48)\*e;

f(i+1,j+1)=f(i+1,j+1)+(5/48)\*e;

f(i+1,j+2)=f(i+1,j+2)+(3/48)\*e;

f(i+1,j-1)=f(i+1,j-1)+(5/48)\*e;

f(i+1,j-2)=f(i+1,j-2)+(3/48)\*e;

end

end

end

else

if i~=row

for j=col:-1:1

if (f(i,j)>T)

b(i,j)=255;

else

b(i,j)=0;

end

e=f(i,j)-b(i,j);

if j==col

f(i,j-1)=f(i,j-1)+(7/48)\*e;

f(i,j-2)=f(i,j-2)+(5/48)\*e;

f(i+1,j)=f(i+1,j)+(7/48)\*e;

f(i+2,j)=f(i+2,j)+(5/48)\*e;

f(i+2,j-1)=f(i+2,j-1)+(3/48)\*e;

f(i+2,j-2)=f(i+2,j-2)+(1/48)\*e;

f(i+1,j-1)=f(i+1,j-1)+(5/48)\*e;

f(i+1,j-2)=f(i+1,j-2)+(3/48)\*e;

elseif j==col-1

f(i+1,j)=f(i+1,j)+(7/48)\*e;

f(i+2,j)=f(i+2,j)+(5/48)\*e;

f(i,j-1)=f(i,j-1)+(7/48)\*e;

f(i,j-2)=f(i,j-2)+(5/48)\*e;

f(i+2,j-1)=f(i+2,j-1)+(3/48)\*e;

f(i+2,j-2)=f(i+2,j-2)+(1/48)\*e;

f(i+1,j-1)=f(i+1,j-1)+(5/48)\*e;

f(i+1,j-2)=f(i+1,j-2)+(3/48)\*e;

f(i+2,j+1)=f(i+2,j+1)+(3/48)\*e;

f(i+1,j+1)=f(i+1,j+1)+(5/48)\*e;

elseif j==1

f(i+1,j)=f(i+1,j)+(7/48)\*e;

f(i+2,j)=f(i+2,j)+(5/48)\*e;

f(i+2,j+1)=f(i+2,j+1)+(3/48)\*e;

f(i+2,j+2)=f(i+2,j+2)+(1/48)\*e;

f(i+1,j+1)=f(i+1,j+1)+(5/48)\*e;

f(i+1,j+2)=f(i+1,j+2)+(3/48)\*e;

elseif j==2

f(i,j-1)=f(i,j-1)+(7/48)\*e;

f(i+2,j-1)=f(i+2,j-1)+(3/48)\*e;

f(i+1,j-1)=f(i+1,j-1)+(5/48)\*e;

f(i+1,j)=f(i+1,j)+(7/48)\*e;

f(i+2,j)=f(i+2,j)+(5/48)\*e;

f(i+2,j+1)=f(i+2,j+1)+(3/48)\*e;

f(i+2,j+2)=f(i+2,j+2)+(1/48)\*e;

f(i+1,j+1)=f(i+1,j+1)+(5/48)\*e;

f(i+1,j+2)=f(i+1,j+2)+(3/48)\*e;

else

f(i+1,j)=f(i+1,j)+(7/48)\*e;

f(i+2,j)=f(i+2,j)+(5/48)\*e;

f(i,j-1)=f(i,j-1)+(7/48)\*e;

f(i,j-2)=f(i,j-2)+(5/48)\*e;

f(i+2,j-1)=f(i+2,j-1)+(3/48)\*e;

f(i+2,j-2)=f(i+2,j-2)+(1/48)\*e;

f(i+1,j-1)=f(i+1,j-1)+(5/48)\*e;

f(i+1,j-2)=f(i+1,j-2)+(3/48)\*e;

f(i+2,j+1)=f(i+2,j+1)+(3/48)\*e;

f(i+2,j+2)=f(i+2,j+2)+(1/48)\*e;

f(i+1,j+1)=f(i+1,j+1)+(5/48)\*e;

f(i+1,j+2)=f(i+1,j+2)+(3/48)\*e;

end

end

else

for j=col:-1:1

if (f(i,j)>T)

b(i,j)=255;

else

b(i,j)=0;

end

e=f(i,j)-b(i,j);

if (j==2)

f(i,j-1)=f(i,j-1)+(7/48)\*e;

elseif (j==1)

else

f(i,j-1)=f(i,j-1)+(7/48)\*e;

f(i,j-2)=f(i,j-2)+(5/48)\*e;

end

end

end

end

end

figure(3);

imshow(uint8(b));

title('Jarvis Error Diffusion');

%Stucki

b=zeros(row,col);

f=I;

for i=1:row

if(mod(i,2)~=0)

if i~= (row-1)

for j=1:col

if (f(i,j)>T)

b(i,j)=255;

else

b(i,j)=0;

end

e=f(i,j)-b(i,j);

if j==1

f(i,j+1)=f(i,j+1)+(8/42)\*e;

f(i,j+2)=f(i,j+2)+(4/42)\*e;

f(i+1,j+1)=f(i+1,j+1)+(4/42)\*e;

f(i+1,j+2)=f(i+1,j+2)+(2/42)\*e;

f(i+2,j+1)=f(i+2,j+1)+(2/42)\*e;

f(i+2,j+2)=f(i+2,j+2)+(1/42)\*e;

f(i+1,j)=f(i+1,j)+(8/42)\*e;

f(i+2,j)=f(i+2,j)+(4/42)\*e;

elseif j==2

f(i,j+1)=f(i,j+1)+(8/42)\*e;

f(i,j+2)=f(i,j+2)+(4/42)\*e;

f(i+1,j)=f(i+1,j)+(8/42)\*e;

f(i+2,j)=f(i+2,j)+(4/42)\*e;

f(i+1,j+1)=f(i+1,j+1)+(4/42)\*e;

f(i+1,j+2)=f(i+1,j+2)+(2/42)\*e;

f(i+2,j+1)=f(i+2,j+1)+(2/42)\*e;

f(i+2,j+2)=f(i+2,j+2)+(1/42)\*e;

f(i+1,j-1)=f(i+1,j-1)+(4/42)\*e;

f(i+2,j-1)=f(i+2,j-1)+(2/42)\*e;

elseif j==col

f(i+1,j-1)=f(i+1,j-1)+(4/42)\*e;

f(i+1,j-2)=f(i+1,j-2)+(2/42)\*e;

f(i+2,j-1)=f(i+2,j-1)+(2/42)\*e;

f(i+2,j-2)=f(i+2,j-2)+(1/42)\*e;

f(i+1,j)=f(i+1,j)+(8/42)\*e;

f(i+2,j)=f(i+2,j)+(4/42)\*e;

elseif j==col-1

f(i+1,j-1)=f(i+1,j-1)+(4/42)\*e;

f(i+1,j-2)=f(i+1,j-2)+(2/42)\*e;

f(i+2,j-1)=f(i+2,j-1)+(2/42)\*e;

f(i+2,j-2)=f(i+2,j-2)+(1/42)\*e;

f(i+1,j)=f(i+1,j)+(8/42)\*e;

f(i+2,j)=f(i+2,j)+(4/42)\*e;

f(i,j+1)=f(i,j+1)+(8/42)\*e;

f(i+1,j+1)=f(i+1,j+1)+(4/42)\*e;

f(i+2,j+1)=f(i+2,j+1)+(2/42)\*e;

else

f(i,j+1)=f(i,j+1)+(8/42)\*e;

f(i,j+2)=f(i,j+2)+(4/42)\*e;

f(i+1,j)=f(i+1,j)+(8/42)\*e;

f(i+2,j)=f(i+2,j)+(4/42)\*e;

f(i+1,j+1)=f(i+1,j+1)+(4/42)\*e;

f(i+1,j+2)=f(i+1,j+2)+(2/42)\*e;

f(i+2,j+1)=f(i+2,j+1)+(2/42)\*e;

f(i+2,j+2)=f(i+2,j+2)+(1/42)\*e;

f(i+1,j-1)=f(i+1,j-1)+(4/42)\*e;

f(i+1,j-2)=f(i+1,j-2)+(2/42)\*e;

f(i+2,j-1)=f(i+2,j-1)+(2/42)\*e;

f(i+2,j-2)=f(i+2,j-2)+(1/42)\*e;

end

end

else

for j=col:-1:1

if (f(i,j)>T)

b(i,j)=255;

else

b(i,j)=0;

end

e=f(i,j)-b(i,j);

if j==1

f(i,j+1)=f(i,j+1)+(8/42)\*e;

f(i,j+2)=f(i,j+2)+(4/42)\*e;

f(i+1,j)=f(i+1,j)+(8/42)\*e;

f(i+1,j+1)=f(i+1,j+1)+(4/42)\*e;

f(i+1,j+2)=f(i+1,j+2)+(2/42)\*e;

elseif j==2

f(i,j+1)=f(i,j+1)+(8/42)\*e;

f(i,j+2)=f(i,j+2)+(4/42)\*e;

f(i+1,j)=f(i+1,j)+(8/42)\*e;

f(i+1,j+1)=f(i+1,j+1)+(4/42)\*e;

f(i+1,j+2)=f(i+1,j+2)+(2/42)\*e;

f(i+1,j-1)=f(i+1,j-1)+(4/42)\*e;

elseif j==col

f(i+1,j-1)=f(i+1,j-1)+(4/42)\*e;

f(i+1,j-2)=f(i+1,j-2)+(2/42)\*e;

f(i+1,j)=f(i+1,j)+(8/42)\*e;

elseif j==col-1

f(i+1,j-1)=f(i+1,j-1)+(4/42)\*e;

f(i+1,j-2)=f(i+1,j-2)+(2/42)\*e;

f(i+1,j)=f(i+1,j)+(8/42)\*e;

f(i,j+1)=f(i,j+1)+(8/42)\*e;

f(i+1,j+1)=f(i+1,j+1)+(4/42)\*e;

else

f(i,j+1)=f(i,j+1)+(8/42)\*e;

f(i,j+2)=f(i,j+2)+(4/42)\*e;

f(i+1,j)=f(i+1,j)+(8/42)\*e;

f(i+1,j+1)=f(i+1,j+1)+(4/42)\*e;

f(i+1,j+2)=f(i+1,j+2)+(2/42)\*e;

f(i+1,j-1)=f(i+1,j-1)+(4/42)\*e;

f(i+1,j-2)=f(i+1,j-2)+(2/42)\*e;

end

end

end

else

if i~=row

for j=col:-1:1

if (f(i,j)>T)

b(i,j)=255;

else

b(i,j)=0;

end

e=f(i,j)-b(i,j);

if j==col

f(i,j-1)=f(i,j-1)+(8/42)\*e;

f(i,j-2)=f(i,j-2)+(4/42)\*e;

f(i+1,j)=f(i+1,j)+(8/42)\*e;

f(i+2,j)=f(i+2,j)+(4/42)\*e;

f(i+2,j-1)=f(i+2,j-1)+(2/42)\*e;

f(i+2,j-2)=f(i+2,j-2)+(1/42)\*e;

f(i+1,j-1)=f(i+1,j-1)+(4/42)\*e;

f(i+1,j-2)=f(i+1,j-2)+(2/42)\*e;

elseif j==col-1

f(i+1,j)=f(i+1,j)+(8/42)\*e;

f(i+2,j)=f(i+2,j)+(4/42)\*e;

f(i,j-1)=f(i,j-1)+(8/42)\*e;

f(i,j-2)=f(i,j-2)+(4/42)\*e;

f(i+2,j-1)=f(i+2,j-1)+(2/42)\*e;

f(i+2,j-2)=f(i+2,j-2)+(1/42)\*e;

f(i+1,j-1)=f(i+1,j-1)+(4/42)\*e;

f(i+1,j-2)=f(i+1,j-2)+(2/42)\*e;

f(i+2,j+1)=f(i+2,j+1)+(2/42)\*e;

f(i+1,j+1)=f(i+1,j+1)+(4/42)\*e;

elseif j==1

f(i+1,j)=f(i+1,j)+(8/42)\*e;

f(i+2,j)=f(i+2,j)+(4/42)\*e;

f(i+2,j+1)=f(i+2,j+1)+(2/42)\*e;

f(i+2,j+2)=f(i+2,j+2)+(1/42)\*e;

f(i+1,j+1)=f(i+1,j+1)+(4/42)\*e;

f(i+1,j+2)=f(i+1,j+2)+(2/42)\*e;

elseif j==2

f(i,j-1)=f(i,j-1)+(8/42)\*e;

f(i+2,j-1)=f(i+2,j-1)+(2/42)\*e;

f(i+1,j-1)=f(i+1,j-1)+(4/42)\*e;

f(i+1,j)=f(i+1,j)+(8/42)\*e;

f(i+2,j)=f(i+2,j)+(4/42)\*e;

f(i+2,j+1)=f(i+2,j+1)+(2/42)\*e;

f(i+2,j+2)=f(i+2,j+2)+(1/42)\*e;

f(i+1,j+1)=f(i+1,j+1)+(4/42)\*e;

f(i+1,j+2)=f(i+1,j+2)+(2/42)\*e;

else

f(i+1,j)=f(i+1,j)+(8/42)\*e;

f(i+2,j)=f(i+2,j)+(4/42)\*e;

f(i,j-1)=f(i,j-1)+(8/42)\*e;

f(i,j-2)=f(i,j-2)+(4/42)\*e;

f(i+2,j-1)=f(i+2,j-1)+(2/42)\*e;

f(i+2,j-2)=f(i+2,j-2)+(1/42)\*e;

f(i+1,j-1)=f(i+1,j-1)+(4/42)\*e;

f(i+1,j-2)=f(i+1,j-2)+(2/42)\*e;

f(i+2,j+1)=f(i+2,j+1)+(2/42)\*e;

f(i+2,j+2)=f(i+2,j+2)+(1/42)\*e;

f(i+1,j+1)=f(i+1,j+1)+(4/42)\*e;

f(i+1,j+2)=f(i+1,j+2)+(2/42)\*e;

end

end

else

for j=col:-1:1

if (f(i,j)>T)

b(i,j)=255;

else

b(i,j)=0;

end

e=f(i,j)-b(i,j);

if (j==2)

f(i,j-1)=f(i,j-1)+(8/42)\*e;

elseif (j==1)

else

f(i,j-1)=f(i,j-1)+(8/42)\*e;

f(i,j-2)=f(i,j-2)+(4/42)\*e;

end

end

end

end

end

figure(4);

imshow(uint8(b));

title('Stucki Error Diffusion');

Problem 2 C : Color Halftoning with Error Diffusion

1. Separable Error Method

% Sourabh Tirodkar

% 3589406164

% tirodkar@usc.edu

% Submission Date- 16th Feb,20

% PROBLEM 2C

% COLOR HALFTONING

% ROSE IMAGE

% SEPARABLE ERROR METHOD

%read image

I = imread('C:\Users\user\Desktop\USC Classes\Spring 2020\EE569- Digital Image Processing\Homeworks\HW #2\EE569\_Spring\_2020\_HW2\_Materials\Problem2\Rose.jpg');

figure(1)

imshow(uint8(I));

I = double(I);

R=I(:,:,1);

G=I(:,:,2);

B=I(:,:,3);

%Calculating C,M,Y

C=255-R;

M=255-G;

Y=255-B;

CMY=cat(3,C,M,Y);

figure(2)

imshow(uint8(CMY))

%%

% %Floyd Steinberg Matrix

% h=(1/16)\*[0 0 0;0 0 7;3 5 1];

% h1=(1/16)\*[0 0 0;7 0 0;1 5 3];

T=127;

row=480;

col=640;

%%

%for cyan

for i = 1 : row

% odd row

if mod(i,2) ~= 0

for j = 1 : col

if C(i,j)>T

b1(i,j)=255;

else

b1(i,j)=0;

end

e = C(i,j) - b1(i,j);

if j==1

C(i,j+1)=C(i,j+1)+(7/16)\*e;

C(i+1,j+1)=C(i+1,j+1)+(1/16)\*e;

C(i+1,j)=C(i+1,j)+(5/16)\*e;

elseif j==col

C(i+1,j-1)=C(i+1,j-1)+(3/16)\*e;

C(i+1,j)=C(i+1,j)+(5/16)\*e;

else

C(i,j+1)=C(i,j+1)+(7/16)\*e;

C(i+1,j-1)=C(i+1,j-1)+(3/16)\*e;

C(i+1,j)=C(i+1,j)+(5/16)\*e;

C(i+1,j+1)=C(i+1,j+1)+(1/16)\*e;

end

end

end

% even row

if mod(i,2) == 0

if i~=row

for j = col:-1:1

if C(i,j)>T

b1(i,j)=255;

else

b1(i,j)=0;

end

e = C(i,j) - b1(i,j);

if j==col

C(i,j-1)=C(i,j-1)+(7/16)\*e;

C(i+1,j-1)=C(i+1,j-1)+(1/16)\*e;

C(i+1,j)=C(i+1,j)+(5/16)\*e;

elseif j==1

C(i+1,j+1)=C(i+1,j+1)+(3/16)\*e;

C(i+1,j)=C(i+1,j)+(5/16)\*e;

else

C(i,j-1)=C(i,j-1)+(7/16)\*e;

C(i+1,j-1)=C(i+1,j-1)+(1/16)\*e;

C(i+1,j)=C(i+1,j)+(5/16)\*e;

C(i+1,j+1)=C(i+1,j+1)+(3/16)\*e;

end

end

end

end

end

%for magenta

for i = 1 : row

% odd row

if mod(i,2) ~= 0

for j = 1 : col

if M(i,j)>T

b2(i,j)=255;

else

b2(i,j)=0;

end

e = M(i,j) - b2(i,j);

if j==1

M(i,j+1)=M(i,j+1)+(7/16)\*e;

M(i+1,j+1)=M(i+1,j+1)+(1/16)\*e;

M(i+1,j)=M(i+1,j)+(5/16)\*e;

elseif j==col

M(i+1,j-1)=M(i+1,j-1)+(3/16)\*e;

M(i+1,j)=M(i+1,j)+(5/16)\*e;

else

M(i,j+1)=M(i,j+1)+(7/16)\*e;

M(i+1,j-1)=M(i+1,j-1)+(3/16)\*e;

M(i+1,j)=M(i+1,j)+(5/16)\*e;

M(i+1,j+1)=M(i+1,j+1)+(1/16)\*e;

end

end

end

% even row

if mod(i,2) == 0

if i~=row

for j = col:-1:1

if M(i,j)>T

b2(i,j)=255;

else

b2(i,j)=0;

end

e = M(i,j) - b2(i,j);

if j==col

M(i,j-1)=M(i,j-1)+(7/16)\*e;

M(i+1,j-1)=M(i+1,j-1)+(1/16)\*e;

M(i+1,j)=M(i+1,j)+(5/16)\*e;

elseif j==1

M(i+1,j+1)=M(i+1,j+1)+(3/16)\*e;

M(i+1,j)=M(i+1,j)+(5/16)\*e;

else

M(i,j-1)=M(i,j-1)+(7/16)\*e;

M(i+1,j-1)=M(i+1,j-1)+(1/16)\*e;

M(i+1,j)=M(i+1,j)+(5/16)\*e;

M(i+1,j+1)=M(i+1,j+1)+(3/16)\*e;

end

end

end

end

end

%for yellow

for i = 1 : row

% odd row

if mod(i,2) ~= 0

for j = 1 : col

if Y(i,j)>T

b3(i,j)=255;

else

b3(i,j)=0;

end

e = Y(i,j) - b3(i,j);

if j==1

Y(i,j+1)=Y(i,j+1)+(7/16)\*e;

Y(i+1,j+1)=Y(i+1,j+1)+(1/16)\*e;

Y(i+1,j)=Y(i+1,j)+(5/16)\*e;

elseif j==col

Y(i+1,j-1)=Y(i+1,j-1)+(3/16)\*e;

Y(i+1,j)=Y(i+1,j)+(5/16)\*e;

else

Y(i,j+1)=Y(i,j+1)+(7/16)\*e;

Y(i+1,j-1)=Y(i+1,j-1)+(3/16)\*e;

Y(i+1,j)=Y(i+1,j)+(5/16)\*e;

Y(i+1,j+1)=Y(i+1,j+1)+(1/16)\*e;

end

end

end

% even row

if mod(i,2) == 0

if i~=row

for j = col:-1:1

if Y(i,j)>T

b3(i,j)=255;

else

b3(i,j)=0;

end

e = Y(i,j) - b3(i,j);

if j==col

Y(i,j-1)=Y(i,j-1)+(7/16)\*e;

Y(i+1,j-1)=Y(i+1,j-1)+(1/16)\*e;

Y(i+1,j)=Y(i+1,j)+(5/16)\*e;

elseif j==1

Y(i+1,j+1)=Y(i+1,j+1)+(3/16)\*e;

Y(i+1,j)=Y(i+1,j)+(5/16)\*e;

else

Y(i,j-1)=Y(i,j-1)+(7/16)\*e;

Y(i+1,j-1)=Y(i+1,j-1)+(1/16)\*e;

Y(i+1,j)=Y(i+1,j)+(5/16)\*e;

Y(i+1,j+1)=Y(i+1,j+1)+(3/16)\*e;

end

end

end

end

end

rr=255-b1;

gg=255-b2;

bb=255-b3;

RGB123=zeros(479,640, 3);

RGB123(:,:,1)=rr;

RGB123(:,:,2)=gg;

RGB123(:,:,3)=bb;

figure(3)

imshow(uint8(RGB123))

title('Color Halftone By Separable Error Diffusion')

1. MBVQ

% Sourabh Tirodkar

% 3589406164

% tirodkar@usc.edu

% Submission Date- 16th Feb,20

% PROBLEM 2C

% COLOR HALFTONING

% ROSE IMAGE

% MBVQ METHOD

I = imread('C:\Users\user\Desktop\USC Classes\Spring 2020\EE569- Digital Image Processing\Homeworks\HW #2\EE569\_Spring\_2020\_HW2\_Materials\Problem2\Rose.jpg');

figure(1)

imshow(uint8(I));

I = double(I);

R=I(:,:,1);

G=I(:,:,2);

B=I(:,:,3);

v=zeros(480,640,3);

[row, col]=size(R);

fr=double(R);

fg=double(G);

fb=double(B);

for i=1:row

if(mod(i,2)~=0)

if i~=row

for j=1:col

[mbvq,v]=MBVQ(double(R(i,j)),double(G(i,j)),double(B(i,j)),fr(i,j),fg(i,j),fb(i,j));

br(i,j)=v(1,1,1);

bg(i,j)=v(1,1,2);

bb(i,j)=v(1,1,3);

er=fr(i,j)-br(i,j);

eg=fg(i,j)-bg(i,j);

eb=fb(i,j)-bb(i,j);

if j==1

fr(i,j+1)=fr(i,j+1)+(7/16)\*er;

fr(i+1,j+1)=fr(i+1,j+1)+(1/16)\*er;

fr(i+1,j)=fr(i+1,j)+(5/16)\*er;

fg(i,j+1)=fg(i,j+1)+(7/16)\*eg;

fg(i+1,j+1)=fg(i+1,j+1)+(1/16)\*eg;

fg(i+1,j)=fg(i+1,j)+(5/16)\*eg;

fb(i,j+1)=fb(i,j+1)+(7/16)\*eb;

fb(i+1,j+1)=fb(i+1,j+1)+(1/16)\*eb;

fb(i+1,j)=fb(i+1,j)+(5/16)\*eb;

elseif j==col

fr(i+1,j-1)=fr(i+1,j-1)+(3/16)\*er;

fr(i+1,j)=fr(i+1,j)+(5/16)\*er;

fg(i+1,j-1)=fg(i+1,j-1)+(3/16)\*eg;

fg(i+1,j)=fg(i+1,j)+(5/16)\*eg;

fb(i+1,j-1)=fb(i+1,j-1)+(3/16)\*eb;

fb(i+1,j)=fb(i+1,j)+(5/16)\*eb;

else

fr(i,j+1)=fr(i,j+1)+(7/16)\*er;

fr(i+1,j-1)=fr(i+1,j-1)+(3/16)\*er;

fr(i+1,j)=fr(i+1,j)+(5/16)\*er;

fr(i+1,j+1)=fr(i+1,j+1)+(1/16)\*er;

fg(i,j+1)=fg(i,j+1)+(7/16)\*eg;

fg(i+1,j-1)=fg(i+1,j-1)+(3/16)\*eg;

fg(i+1,j)=fg(i+1,j)+(5/16)\*eg;

fg(i+1,j+1)=fg(i+1,j+1)+(1/16)\*eg;

fb(i,j+1)=fb(i,j+1)+(7/16)\*eb;

fb(i+1,j-1)=fb(i+1,j-1)+(3/16)\*eb;

fb(i+1,j)=fb(i+1,j)+(5/16)\*eb;

fb(i+1,j+1)=fb(i+1,j+1)+(1/16)\*eb;

end

end

else

for j=1:col

[mbvq,v]=MBVQ(double(R(i,j)),double(G(i,j)),double(B(i,j)),fr(i,j),fg(i,j),fb(i,j));

br(i,j)=v(1,1,1);

bg(i,j)=v(1,1,2);

bb(i,j)=v(1,1,3);

er=fr(i,j)-br(i,j);

eg=fg(i,j)-bg(i,j);

eb=fb(i,j)-bb(i,j);

if j~=col

fr(i,j+1)=fr(i,j+1)+(7/16)\*er;

fg(i,j+1)=fg(i,j+1)+(7/16)\*eg;

fb(i,j+1)=fb(i,j+1)+(7/16)\*eb;

end

end

end

else

if i~=row

for j=col:-1:1

[mbvq,v]=MBVQ(double(R(i,j)),double(G(i,j)),double(B(i,j)),fr(i,j),fg(i,j),fb(i,j));

br(i,j)=v(1,1,1);

bg(i,j)=v(1,1,2);

bb(i,j)=v(1,1,3);

er=fr(i,j)-br(i,j);

eg=fg(i,j)-bg(i,j);

eb=fb(i,j)-bb(i,j);

if j==col

fr(i,j-1)=fr(i,j-1)+(7/16)\*er;

fr(i+1,j-1)=fr(i+1,j-1)+(1/16)\*er;

fr(i+1,j)=fr(i+1,j)+(5/16)\*er;

fg(i,j-1)=fg(i,j-1)+(7/16)\*eg;

fg(i+1,j-1)=fg(i+1,j-1)+(1/16)\*eg;

fg(i+1,j)=fg(i+1,j)+(5/16)\*eg;

fb(i,j-1)=fb(i,j-1)+(7/16)\*eb;

fb(i+1,j-1)=fb(i+1,j-1)+(1/16)\*eb;

fb(i+1,j)=fb(i+1,j)+(5/16)\*eb;

elseif j==1

fr(i+1,j+1)=fr(i+1,j+1)+(3/16)\*er;

fr(i+1,j)=fr(i+1,j)+(5/16)\*er;

fg(i+1,j+1)=fg(i+1,j+1)+(3/16)\*eg;

fg(i+1,j)=fg(i+1,j)+(5/16)\*eg;

fb(i+1,j+1)=fb(i+1,j+1)+(3/16)\*eb;

fb(i+1,j)=fb(i+1,j)+(5/16)\*eb;

else

fr(i,j-1)=fr(i,j-1)+(7/16)\*er;

fr(i+1,j-1)=fr(i+1,j-1)+(1/16)\*er;

fr(i+1,j)=fr(i+1,j)+(5/16)\*er;

fr(i+1,j+1)=fr(i+1,j+1)+(3/16)\*er;

fg(i,j-1)=fg(i,j-1)+(7/16)\*eg;

fg(i+1,j-1)=fg(i+1,j-1)+(1/16)\*eg;

fg(i+1,j)=fg(i+1,j)+(5/16)\*eg;

fg(i+1,j+1)=fg(i+1,j+1)+(3/16)\*eg;

fb(i,j-1)=fb(i,j-1)+(7/16)\*eb;

fb(i+1,j-1)=fb(i+1,j-1)+(1/16)\*eb;

fb(i+1,j)=fb(i+1,j)+(5/16)\*eb;

fb(i+1,j+1)=fb(i+1,j+1)+(3/16)\*eb;

end

end

else

for j=col:-1:1

[mbvq,v]=MBVQ(double(R(i,j)),double(G(i,j)),double(B(i,j)),fr(i,j),fg(i,j),fb(i,j));

br(i,j)=v(1,1,1);

bg(i,j)=v(1,1,2);

bb(i,j)=v(1,1,3);

er=fr(i,j)-br(i,j);

eg=fg(i,j)-bg(i,j);

eb=fb(i,j)-bb(i,j);

if j~=1

fr(i,j-1)=fr(i,j-1)+(7/16)\*er;

fg(i,j-1)=fg(i,j-1)+(7/16)\*eg;

fb(i,j-1)=fb(i,j-1)+(7/16)\*eb;

end

end

end

end

end

mbvq\_rgb(:,:,1)=br;

figure(2);

imshow(uint8(mbvq\_rgb(:,:,1)));

title('MBVQ - Red');

mbvq\_rgb(:,:,2)=bg;

figure(3);

imshow(uint8(mbvq\_rgb(:,:,2)));

title('MBVQ - Green');

mbvq\_rgb(:,:,3)=bb;

figure(4);

imshow(uint8(mbvq\_rgb(:,:,3)));

title('MBVQ - Blue');

figure(5);

imshow(uint8(mbvq\_rgb));

title('MBVQ algorithm');