Image Processing and Signal - UsingMatlab

April 25

2025

By: Aya Hussain Merai
Class: B

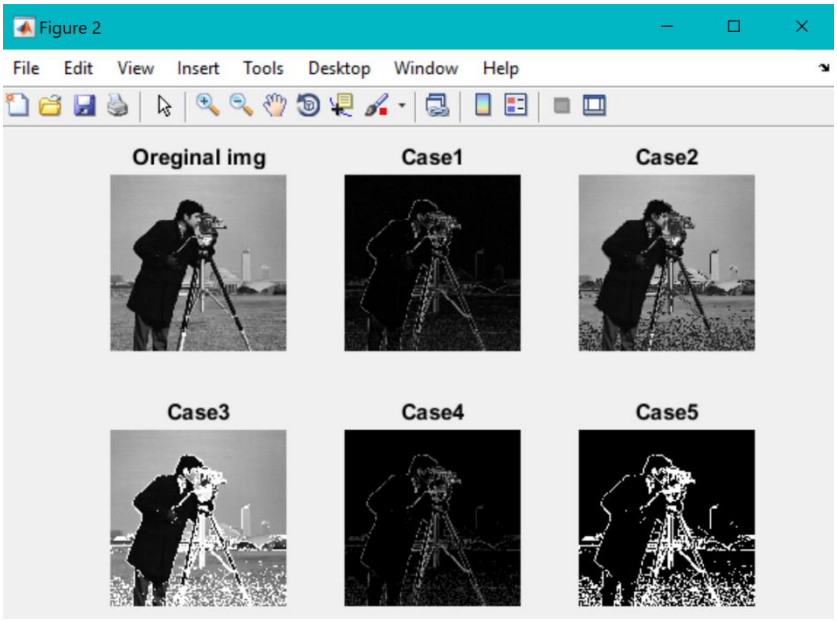
All Programs
Course2

```
clc; clear;
% Image SHARPNING
% Applying differentiation method to SHARP the following points:
f = [1 2 3 5;
     2 1 1 0;
     1 3 2 4;
     4 0 3 1];
T = 3; %  For CASE2
LG = 255; %  For CASE3
LB = 0; %  For CASE4
[M, N] = size(f);
fe = f;
fe(M+1, :) = fe(M, :);
fe(:, M+1) = fe(:, M);
G = zeros(M, N);
G1 = G;
G2 = G;
G3 = G;
G4 = G;
G5 = G;
for x=1:M
   for y=1:N
    G(x, y) = sqrt((fe(x,y)-fe(x+1,y))^2+(fe(x,y)-fe(x,y+1))^2);
    G1(x, y) = G(x, y);
    if G(x, y) >= T
       G2(x, y) = G(x, y);
       G3(x, y) = LG;
       G4(x, y) = G(x, y);
       G5(x, y) = LG;
    else
       G2(x, y) = fe(x, y);
       G3(x, y) = fe(x, y);
       G4(x, y) = LB;
       G5(x, y) = LB;
    end
  end
end
g1 = round(G1);
g2 = round(G2);
g3 = round(G3);
g4 = round(G4);
g5 = round(G5);
disp('f:');
disp(f);
disp('fe:');
disp(fe);
disp('------------------------------');
disp('g1:'); disp(g1);
disp('g2:'); disp(g2);
disp('g3:'); disp(g3);
disp('g4:'); disp(g4);
disp('g5:'); disp(g5);
```

Command Window

```
f:
     1     2     3     5
     2     1     1     0
     1     3     2     4
     4     0     3     1

fe:
     1     2     3     5     5
     2     1     1     0     0
     1     3     2     4     4
     4     0     3     1     1
     4     0     3     1     1
------------------------------
g1:
     1     1     3     5
     1     2     1     4
     4     3     2     3
     4     3     2     0

g2:
     1     2     3     5
     2     1     1     4
     4     3     2     3
     4     3     3     1

g3:
     1     2     3   255
     2     1     1   255
   255   255     2   255
   255   255     3     1

g4:
     0     0     0     5
     0     0     0     4
     4     3     0     3
     4     3     0     0

g5:
     0     0     0   255
     0     0     0   255
   255   255     0   255
   255   255     0     0

fx >>
```

```
clc; clear;
% Image SHARPNING
f = double(imread('cameraman.tif'));
T = 25; %  For CASE2
LG = 255; %  For CASE3
LB = 0; %  For CASE4
[M, N] = size(f);
fe = f;
fe(M+1, :) = fe(M, :);
fe(:, M+1) = fe(:, M);
G = zeros(M, N);
G1 = G;
G2 = G;
G3 = G;
G4 = G;
G5 = G;
for x=1:M
  for y=1:N
    G(x, y) = sqrt((fe(x,y)-fe(x+1,y))^2+(fe(x,y)-fe(x,y+1))^2);
    G1(x, y) = G(x, y);
    if G(x,y) >= T
      G2(x,y) = G(x,y);
      G3(x,y) = LG;
      G4(x,y) = G(x,y);
      G5(x,y) = LG;
    else
      G2(x,y) = fe(x,y);
      G3(x,y) = fe(x,y);
      G4(x,y) = LB;
      G5(x,y) = LB;
    end
  end
end
g1 = round(G1);
g2 = round(G2);
g3 = round(G3);
g4 = round(G4);
g5 = round(G5);
figure;
subplot(2,3,1),imshow(uint8(f));title('Oreginal img');
subplot(2,3,2),imshow(uint8(g1));title('Case1');

subplot(2,3,3),imshow(uint8(g2));title('Case2');
subplot(2,3,4),imshow(uint8(g3));title('Case3');

subplot(2,3,5),imshow(uint8(g4));title('Case4');
subplot(2,3,6),imshow(uint8(g5));title('Case5');
```

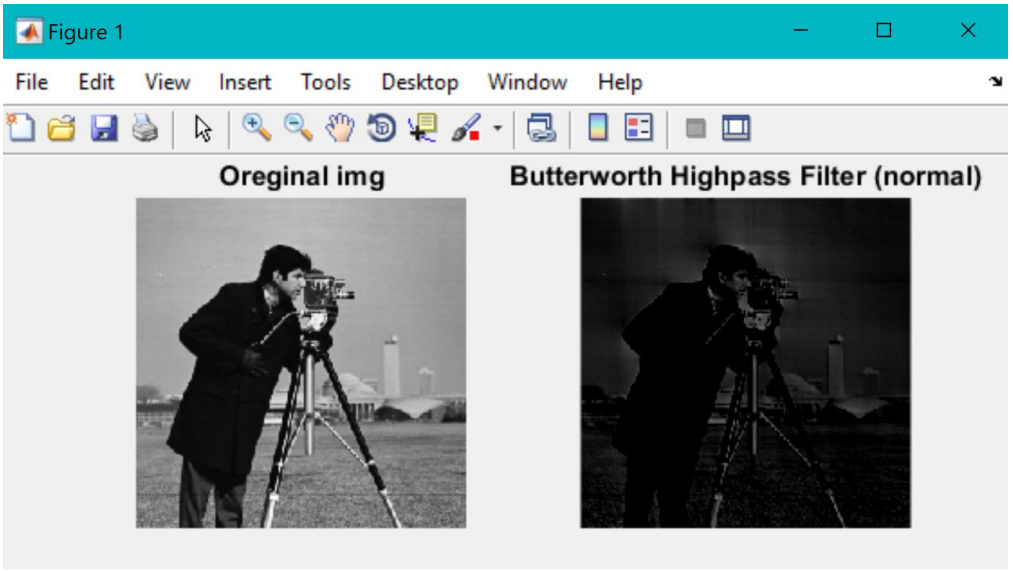


```
clc;
clear;
img = double(imread('cameraman.tif'));
F = fft2(img);
D0 = 128;
[r, c] = size(img);
H = D;

for u = 0:r-1
   for v = 0:c-1
      D = sqrt( (u)^2 + (v)^2 );
      if D <= D0
      H(u+1, v+1) = 0;
      else
        H(u+1, v+1) = 1;
      end
   end
end
G = F.*H;
img2 = round(ifft2(G));
figure;
subplot(1,2,1),imshow(uint8(img));title('Oreginal img');
subplot(1,2,2),imshow(uint8(img2));title('Ideal Highpass Filter');

% ---- Ideal MATRIX ---- %
clear;
D0 = 5;
size = 7;
H = zeros(size);
D = H;
for u = 0:size-1
   for v = 0:size-1
      D = sqrt( (u)^2 + (v)^2 );

      if D <= D0
        H(u+1, v+1) = 0;
      else
        H(u+1, v+1) = 1;
      end
   end
end
disp('Ideal Highpass Filter:');
disp(H);
```

Command Window

```
Ideal Highpass Filter:
     0     0     0     0     0     0     1
     0     0     0     0     0     1     1
     0     0     0     0     0     1     1
     0     0     0     0     0     1     1
     0     0     0     0     1     1     1
     0     1     1     1     1     1     1
     1     1     1     1     1     1     1
```

```
clc;
clear;
img = double(imread('cameraman.tif'));
F = fft2(img);
[r, c] = size(img);
D0 = 64;
n = 1;
H = D;
for u = 0:r-1
   for v = 0:c-1
      D = sqrt( (u)^2 + (v)^2 );
      H(u+1, v+1) = ( 1 / ( 1 + ( D0 / D )^(2*n) ) );
   end
end
G = F.*H;
img2 = round(ifft2(G));
figure;
subplot(1,2,1),imshow(uint8(img));title('Oreginal img');
subplot(1,2,2),imshow(uint8(img2));title('Butterworth Highpass Filter(normal)');
```




```
% ---- MATRIX ---- %

clear;
D0 = 5;
size = [9, 9];
r = size(1);
c = r;
n = 1;
X = 1;
% X = 0.414;
H = D;

for u = 0:r-1
   for v = 0:c-1
      D =
sqrt( (u)^2 + (v)^2 );
      H(u+1, v+1) = ( 1 / ( 1 + X * ( D0 / D )^(2*n) ) );
   end
end
H = round(H, 4);
disp('Butterworth Highpass Filter (normal):');
disp(H);
```

```
Command Window
Butterworth Highpass Filter (normal):
         0    0.0385    0.1379    0.2647    0.3902    0.5000    0.5902    0.6622    0.7191
    0.0385    0.0741    0.1667    0.2857    0.4048    0.5098    0.5968    0.6667    0.7222
    0.1379    0.1667    0.2424    0.3421    0.4444    0.5370    0.6154    0.6795    0.7312
    0.2647    0.2857    0.3421    0.4186    0.5000    0.5763    0.6429    0.6988    0.7449
    0.3902    0.4048    0.4444    0.5000    0.5614    0.6212    0.6753    0.7222    0.7619
    0.5000    0.5098    0.5370    0.5763    0.6212    0.6667    0.7093    0.7475    0.7807
    0.5902    0.5968    0.6154    0.6429    0.6753    0.7093    0.7423    0.7727    0.8000
    0.6622    0.6667    0.6795    0.6988    0.7222    0.7475    0.7727    0.7967    0.8188
    0.7191    0.7222    0.7312    0.7449    0.7619    0.7807    0.8000    0.8188    0.8366
fx >>
```

```
D0 = 3;
size = [5,
5];
r = size(1);
c = r;
n = 2;
H = D;
X = 0.414;
```

Command Window

```
Butterworth Highpass Filter (modified):
         0    0.0290    0.3230    0.7072    0.8842
    0.0290    0.1066    0.4271    0.7489    0.8960
    0.3230    0.4271    0.6562    0.8344    0.9226
    0.7072    0.7489    0.8344    0.9062    0.9491
    0.8842    0.8960    0.9226    0.9491    0.9683
```

```
for u = 0:r-1
   for v = 0:c-1
      D = sqrt( (u)^2 + (v)^2 );
      H(u+1, v+1) = ( 1 / ( 1 + X * ( D0 / D )^(2*n) ) );
   end
end

H = round(H, 4);
disp('Butterworth Highpass Filter (modified):');
disp(H);
```

```
clc;
clear;
img = double(imread('cameraman.tif'));
[r, c] = size(img);
F = fft2(img);
D0 = 230;
n = 1;
H = zeros(r, c);
for u = 0:r-1
   for v = 0:c-1
      D = sqrt( (u)^2 + (v)^2 );
      H(u+1, v+1) = exp(-( D0/D )^n);
   end
end
G = F.*H;
img2 = round(ifft2(G));
figure;
subplot(1,2,1),imshow(uint8(img));title('Oreginal img');
subplot(1,2,2),imshow(uint8(img2));title('Exponitial Highpass Filter(normal)');
% ---------------- %
% ---FOR MATRIX--- %
% ---------------- %
clear;
size = [5, 5];
D0 = 3;
n = 1;
H = zeros(r, c);
for u = 0: size-1
   for v = 0: size-1
      D = sqrt( (u)^2 + (v)^2 );
      H(u+1, v+1) = exp(-( D0/D )^(n));
   end
end
disp('Exp Highpass Filter (normal) :'); disp(H);
```

```
Command Window
Exp Highpass Filter (normal) :
         0    0.0498    0.2231    0.3679    0.4724
    0.0498    0.1199    0.2614    0.3873    0.4831
    0.2231    0.2614    0.3462    0.4352    0.5113
    0.3679    0.3873    0.4352    0.4931    0.5488
    0.4724    0.4831    0.5113    0.5488    0.5884
fx >>
```


```
clc;
clear;
img = double(imread('cameraman.tif'));
[r, c] = size(img);
F = fft2(img);
D0 = 230;
n = 2;
X = 0.347;
H = zeros(r, c);
for u = 0:r-1
   for v = 0:c-1
      D = sqrt( (u)^2 +
(v)^2 );
      H(u+1, v+1) = exp(-X*( D0/D )^n);
   end
end
G = F.*H;
img2 = round(ifft2(G));
figure;
subplot(1,2,1),imshow(uint8(img));title('Oreginal img');
subplot(1,2,2),imshow(uint8(img2));title('Exponitial Highpass Filter(modified)');
```


```
% ---------------- %
% ---FOR MATRIX--- %
% ---------------- %
clear;
size = 5;
r = size(1);
c = size(2);
D0 = 3;
n = 2;
X = 0.347; % X = 1;
H = zeros(r, c);
for u = 0:r-1
   for v = 0:c-1
      D = sqrt( (u)^2 + (v)^2 );
      H(u+1, v+1) = exp(-X*( D0/D )^(n));
   end
end
disp('Exp Highpass Filter (modified) :');
disp(H);
```

```
Command Window
Exp Highpass Filter (modified) :
         0    0.0440    0.4581    0.7068    0.8227
    0.0440    0.2098    0.5355    0.7318    0.8322
    0.4581    0.5355    0.6768    0.7864    0.8554
    0.7068    0.7318    0.7864    0.8407    0.8826
    0.8227    0.8322    0.8554    0.8826    0.9070
```

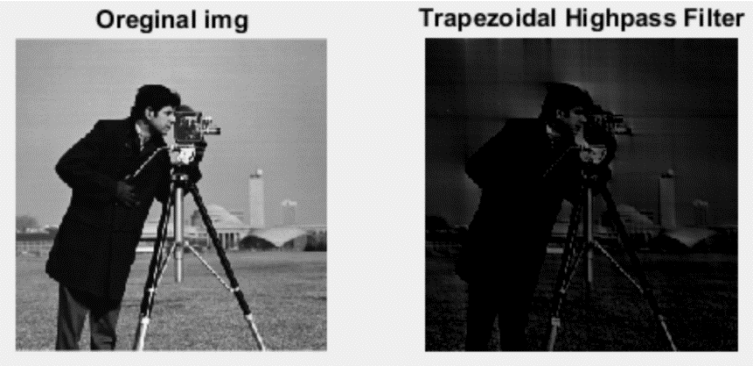
```
clc;
clear;
D0 = 7;   D1 = 5;   n = 2;   size = 9;
H = zeros(size);
for u = 0:size-1
   for v = 0:size-1
      D = sqrt( (u)^2 + (v)^2 );
```

```
      if D < D1
        H(u+1, v+1) = 0;
      elseif D1 <= D && D <= D0
        H(u+1, v+1) = 1/(D0 - D1)*(D(u+1, v+1)-D1);
      elseif D > D0
        H(u+1, v+1) = 1;
      end
   end
end
H = round(H, 4);
disp('Trapezoidal Highpass Filter:');
disp(num2str(H));
```

Command Window

```
Trapezoidal Highpass Filter:
    0         0         0         0         0         0    0.5000    1.0000    1.0000
    0         0         0         0         0    0.0495    0.5414    1.0000    1.0000
    0         0         0         0         0    0.1926    0.6623    1.0000    1.0000
    0         0         0         0         0    0.4155    0.8541    1.0000    1.0000
    0         0         0         0    0.3284    0.7016    1.0000    1.0000    1.0000
    0    0.0495    0.1926    0.4155    0.7016    1.0000    1.0000    1.0000    1.0000
0.5000    0.5414    0.6623    0.8541    1.0000    1.0000    1.0000    1.0000    1.0000
    1         1         1         1         1         1         1         1         1
    1         1         1         1         1         1         1         1         1
```

```
% ----FOR IMG---- %
I =
double(imread('cameraman.tif'));
[r, c] = size(I);  F = fft2(I);
n = 2;
D0 = 155;
D1 =55;
H = zeros(r, c);
for u = 0:r-1
 for v = 0:c-1
    D = sqrt( (u)^2 + (v)^2 );
        if D < D1
            H(u+1, v+1) = 0;
        elseif D1 <= D  && D <= D0
            H(u+1, v+1) = 1/(D0 - D1)*(D(u+1, v+1)-D1);
        elseif D > D0
            H(u+1, v+1) = 1;
        end
    end
end
H = round(H, 4);      G = F.*H;     I2 = round(ifft2(G));
figure;
subplot(1,2,1),imshow(uint8(I));title('Oreginal img');
subplot(1,2,2),imshow(uint8(I2));title('Trapezoidal Highpass Filter');
```



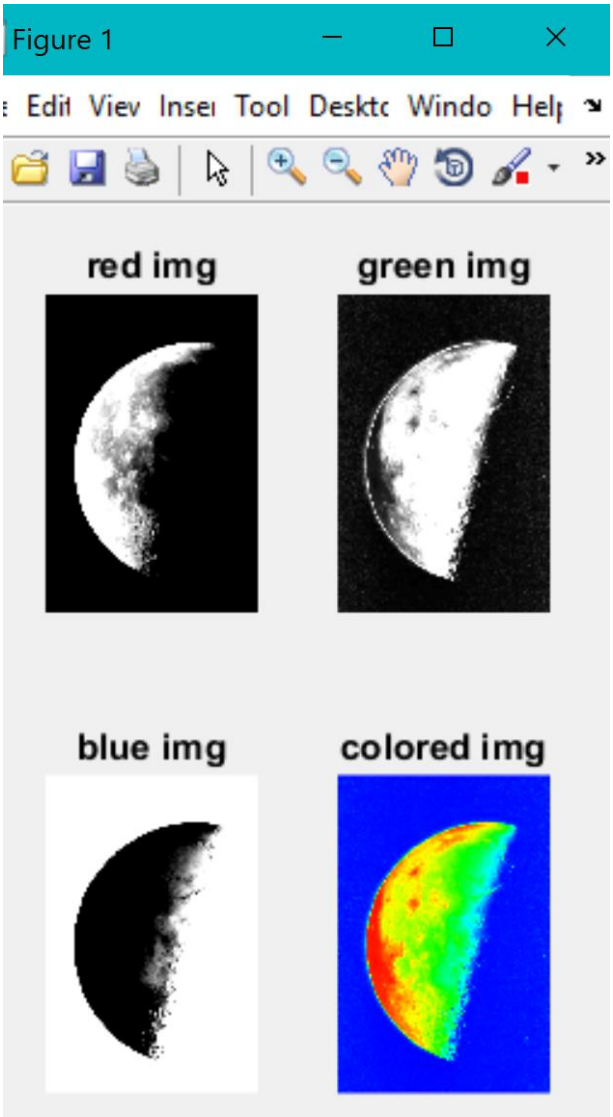
```
clc;
clear;

% Prog1: Write matlab program to design homomorphic filter cut off
frequency  (D0) = 5        size =10*10

D0 = 5;
size = 10;
H_low = zeros(size);
H_high = zeros(size);
D = H_low;

for u = 0:size-1
    for v = 0:size-1
        D = sqrt( (u)^2 + (v)^2 );

        if D <= D0
            H_low(u+1, v+1) = 1.1;
            H_high(u+1, v+1) = 0.9;
        else
            H_low(u+1, v+1) = 0.9;
            H_high(u+1, v+1) = 1.1;
        end
    end
end
disp('Ideal Lowpass Filter:');
disp(num2str(H_low));
disp('------------------------------------------------------------');
disp('Ideal Highpass Filter:');
disp(num2str(H_high));
```

Command Window

```
Ideal Lowpass Filter:
1.1        1.1        1.1        1.1        1.1        1.1        0.9        0.9        0.9        0.9
1.1        1.1        1.1        1.1        1.1        0.9        0.9        0.9        0.9        0.9
1.1        1.1        1.1        1.1        1.1        0.9        0.9        0.9        0.9        0.9
1.1        1.1        1.1        1.1        1.1        0.9        0.9        0.9        0.9        0.9
1.1        1.1        1.1        1.1        0.9        0.9        0.9        0.9        0.9        0.9
1.1        0.9        0.9        0.9        0.9        0.9        0.9        0.9        0.9        0.9
0.9        0.9        0.9        0.9        0.9        0.9        0.9        0.9        0.9        0.9
0.9        0.9        0.9        0.9        0.9        0.9        0.9        0.9        0.9        0.9
0.9        0.9        0.9        0.9        0.9        0.9        0.9        0.9        0.9        0.9
0.9        0.9        0.9        0.9        0.9        0.9        0.9        0.9        0.9        0.9
------------------------------------------------------------
Ideal Highpass Filter:
0.9        0.9        0.9        0.9        0.9        0.9        1.1        1.1        1.1        1.1
0.9        0.9        0.9        0.9        0.9        1.1        1.1        1.1        1.1        1.1
0.9        0.9        0.9        0.9        0.9        1.1        1.1        1.1        1.1        1.1
0.9        0.9        0.9        0.9        0.9        1.1        1.1        1.1        1.1        1.1
0.9        0.9        0.9        0.9        1.1        1.1        1.1        1.1        1.1        1.1
0.9        1.1        1.1        1.1        1.1        1.1        1.1        1.1        1.1        1.1
1.1        1.1        1.1        1.1        1.1        1.1        1.1        1.1        1.1        1.1
1.1        1.1        1.1        1.1        1.1        1.1        1.1        1.1        1.1        1.1
1.1        1.1        1.1        1.1        1.1        1.1        1.1        1.1        1.1        1.1
1.1        1.1        1.1        1.1        1.1        1.1        1.1        1.1        1.1        1.1
fx >>
```

```
clc;
clear;
% Prog2: Write Matlab program to enhancement the image by using
Homomorphic filter
img = double(imread('cameraman.tif'));
[r, c] = size(img);
F = fft2(log(double(img)));
D0_H = 50;
D0_L = 150;
H_low = zeros(r, c);
H_high = zeros(r, c);
for u = 0:r-1
    for v = 0:c-1
        D = sqrt( (u)^2 + (v)^2 );
        if D <= D0_H  % FOR HIGHPASS
            H_high(u+1, v+1) = 0.9;
        else
            H_high(u+1, v+1) = 1.1;
        end
        if D  <= D0_L % FOR LOWPASS
            H_low(u+1, v+1) = 1.1;
        else
            H_low(u+1, v+1) = 0.9;
        end
    end
end
G_high = F.*H_high;
g_high = round(exp(ifft2(G_high)));
img = double(imread('cameraman.tif'));
[r, c] = size(img);
F = fft2(log(double(img)));
G_low = F.*H_low;
g_low = round(exp(ifft2(G_low)));
figure;
subplot(1,3,1),imshow(uint8(img));title('Oreginal img');
subplot(1,3,2),imshow(uint8(g_high));title('Ideal Highpass Filter');
subplot(1,3,3),imshow(uint8(g_low));title('Ideal Lowpass Filter');
```

```
clc;
clear;
% Write a program to Color a graylevel image of size 256*256
img = imread('moon.tif');
L = 255;
[row, col] = size(img);
red   = zeros(row, col);
green = zeros(row, col);
blue  = zeros(row, col);
for i=1:row
  for j=1:col
      gray = double(img(i,j));
%   RED
    if 0 <= gray && gray <= L/2
        red(i,j) = 0;

    elseif L/2 <= gray && gray <= 3*L/4
        red(i,j) = 4*gray-2*L;

    elseif 3*L/4 <= gray && gray <= L
        red(i,j) = L;
        green(i,j) = 4*L - 4*gray;
    end
%   BLUE
    if 0 <= gray && gray <= L/4
        blue(i,j) = L;
        green(i,j) = 4*gray;

    elseif L/4 <= gray && gray <= L/2
        blue(i,j) = 2*L - 4*gray;

    elseif L/2 <= gray && gray <= L
        blue(i,j) = 0;
    end
%   GREEN
    if L/4 <= gray && gray <= 3*L/4
        green(i,j) = L;
    end
  end
end
red = red/L;
green = green/L;
blue = blue/L;

colored3 = zeros(row, col, 3);
colored3(:, :, 1) = red;
colored3(:, :, 2) = green;
colored3(:, :, 3) = blue;

figure;
subplot(2,2,1);imshow(red);title('red img');
subplot(2,2,2);imshow(green);title('green img');
subplot(2,2,3);imshow(blue);title('blue img');
subplot(2,2,4);imshow(colored3, []);title('colored img');
```



```
clc;
clear;
%  Use Huffman encoding algorithm to encode (compress) an image with
8gray levels, when the probabilities of graylevels are as follows:
Graylevel = 1:8;
probability = [0.3, 0.23, 0.15, 0.08, 0.06, 0.06, 0.06, 0.06];

prob_len = length(probability);
prob = zeros(prob_len);
x = zeros(prob_len);

prob(1,:) = sort(probability, 'descend');
x(1,:) = Graylevel;

merge_table = [];

for i=2:prob_len
  prob(i,:) = prob(i-1,:);
  x(i,:) = x(i-1,:);
  not_zero_idx = find(prob(i,:) > 0);
  if length(not_zero_idx) < 2
    break;
  end
  last_two = not_zero_idx(end-1:end);
  prob_sum = sum(prob(i, last_two));
  x_new = max(x(i,:)) + 1;
  merge_table=[merge_table; x(i,last_two(1)),x(i,last_two(2)), x_new];
  prob(i, last_two(1)) = 0;
  prob(i, last_two(2)) = prob_sum;
  x(i, last_two(1)) = 0;
  x(i, last_two(2)) = x_new;
  [prob(i,:), order] = sort(prob(i,:), 'descend');
  x(i,:) = x(i, order);
end
codes = cell(1, prob_len);
for symbol = Graylevel
  code = '';
  current = symbol;
  while true
    idx = find(merge_table(:,1)==current | merge_table(:,2)==current,
1, 'first');
    if isempty(idx)
      break;
    end
    if merge_table(idx,1) == current
      code = ['0', code];
      current = merge_table(idx,3);
    else
      code = ['1', code];
      current = merge_table(idx,3);
    end
  end
  codes{symbol} = code;
end
```

```
disp('x:');
disp(x);

disp('Prop:');
disp(prob);

disp('y:');
disp(codes);
```

```
Command Window
x:
     1     2     3     4     5     6     7     8
     1     2     3     9     4     5     6     0
     1     2     3     9    10     4     0     0
     1     2    11     3     9     0     0     0
     1    12     2    11     0     0     0     0
    13     1    12     0     0     0     0     0
    14    13     0     0     0     0     0     0
    15     0     0     0     0     0     0     0

Prop:
    0.3000    0.2300    0.1500    0.0800    0.0600    0.0600    0.0600    0.0600
    0.3000    0.2300    0.1500    0.1200    0.0800    0.0600    0.0600         0
    0.3000    0.2300    0.1500    0.1200    0.1200    0.0800         0         0
    0.3000    0.2300    0.2000    0.1500    0.1200         0         0         0
    0.3000    0.2700    0.2300    0.2000         0         0         0         0
    0.4300    0.3000    0.2700         0         0         0         0         0
    0.5700    0.4300         0         0         0         0         0         0
    1.0000         0         0         0         0         0         0         0

y:
    '00'    '10'    '010'    '111'    '1100'    '1101'    '0110'    '0111'
```

Lab 6

```
clc;
clear;
% Arithmetic Coding
message = input('message: ', 's');
msg_len = length(message);
symbols = unique(message);
symbol_counts = zeros(1, length(symbols));
for i = 1:msg_len
    idx = find(symbols == message(i));
    symbol_counts(idx) = symbol_counts(idx) + 1;
end
prob = symbol_counts / msg_len;

range = zeros(1, length(symbols));
for i = 2:length(symbols)
    range(i) = range(i-1) + prob(i-1);
end
disp('Symbol | Probability | Range (Low)');
for i = 1:length(symbols)
    fprintf('   %c    |     %.1f       |    %.1f\n', symbols(i), prob(i),
range(i));
end
wordcode = 0;
for i = 1:msg_len
    idx = find(symbols == message(i));
    mul = 1;
    for k = 1:i-1
      prev_idx = find(symbols == message(k));
     mul = mul * prob(prev_idx);
   end
   wordcode = wordcode + range(idx)* mul;
  disp([num2str(i), ': ',
num2str(wordcode)])
end
decoded = repmat(' ', 1, msg_len);
code_decode = wordcode;

fprintf('\nDecoding steps:\n');
for i = 1:msg_len
  for j = 1:length(symbols)
    low = range(j);
    high = low + prob(j);
```

```
Command Window
message: CLASS
Symbol | Probability | Range (Low)
   A   |     0.2     |    0.0
   C   |     0.2     |    0.2
   L   |     0.2     |    0.4
   S   |     0.4     |    0.6
1: 0.2
2: 0.28
3: 0.28
4: 0.2848
5: 0.28672

Decoding steps:
1:   0.28672 = C
2:   0.43360 = L
3:   0.16800 = A
4:   0.84000 = S
5:   0.60000 = S
Decoded message: CLASS
```

```
    if code_decode >= low && code_decode < high
      decoded(i) = symbols(j);
      fprintf('%d:  %.5f = %c\n', i, code_decode, symbols(j));
      code_decode = (code_decode - low) / prob(j);
      break;
    end
  end
end
disp(['Decoded message: ', decoded]);
```

Lab 7

```
clc;
clear;
%% LZW Compression
inputMessage = ' BET BE BEE BED BEG';
dictionarySymbols = {};
dictionaryCodes = [];
nextAvailableCode = 256;
for i = 1:length(inputMessage)
    currentChar = inputMessage(i);
    symbolExists = false;
    for j = 1:length(dictionarySymbols)
        if strcmp(dictionarySymbols{j}, currentChar)
            symbolExists = true;
            break;
        end
    end
    if ~symbolExists
        dictionarySymbols{end+1} = currentChar;
        dictionaryCodes(end+1) = nextAvailableCode;
        nextAvailableCode = nextAvailableCode + 1;
    end
end
currentSequence = '';
encodedOutput = [];
stepCounter = 1;
fprintf('S |Char | Code | Output | New Code/Dictionary\n');
fprintf('-----------------------------------------\n');
for i = 1:length(inputMessage)
    nextChar = inputMessage(i);
    combinedSequence = [currentSequence, nextChar];
    sequenceFound = false;
    for j = 1:length(dictionarySymbols)
        if strcmp(dictionarySymbols{j}, combinedSequence)
            sequenceFound = true;
            break;
        end
    end
    if sequenceFound
        currentSequence = combinedSequence;
    else
        outputCode = -1;
        for j = 1:length(dictionarySymbols)
            if strcmp(dictionarySymbols{j}, currentSequence)
                outputCode = dictionaryCodes(j);
```

```
                break;
            end
        end
        if isempty(currentSequence)
            fprintf('%-2d| "%s"  |  -   |        | "%s" -> %d\n', ...
                stepCounter, nextChar, combinedSequence,
nextAvailableCode);
else
            fprintf('%-2d| "%s"  | %3d | "%s"   | "%s" -> %d\n', ...
                stepCounter, nextChar, outputCode, currentSequence,
combinedSequence, nextAvailableCode);
        end
        encodedOutput(end+1) = outputCode;
        dictionarySymbols{end+1} = combinedSequence;
        dictionaryCodes(end+1) = nextAvailableCode;
        nextAvailableCode = nextAvailableCode + 1;

        currentSequence = nextChar;
        stepCounter = stepCounter + 1;
    end
end
if ~isempty(currentSequence)
    outputCode = -1;
    for j = 1:length(dictionarySymbols)
        if strcmp(dictionarySymbols{j}, currentSequence)
            outputCode = dictionaryCodes(j);
            break;
        end
    end
    fprintf('%-2d| End   | %3d | "%s"\n', stepCounter, outputCode,
currentSequence);
    encodedOutput(end+1) = outputCode;
end
fprintf('\nLZW Encoded Output:\n');
disp(encodedOutput);
```

Command Window

```
S |Char | Code | Output | New Code/Dictionary
-----------------------------------------------
1 | "B"  | 256 | " "   | " B" -> 262
2 | "E"  | 257 | "B"   | "BE" -> 263
3 | "T"  | 258 | "E"   | "ET" -> 264
4 | " "  | 259 | "T"   | "T " -> 265
5 | "E"  | 262 | " B"   | " BE" -> 266
6 | " "  | 258 | "E"   | "E " -> 267
7 | "E"  | 266 | " BE"   | " BEE" -> 268
8 | "B"  | 267 | "E "   | "E B" -> 269
9 | "D"  | 263 | "BE"   | "BED" -> 270
10| " "  | 260 | "D"   | "D " -> 271
11| "G"  | 266 | " BE"   | " BEG" -> 272
12| End   | 261 | "G"

LZW Encoded Output:
   256   257   258   259   262   258   266   267   263   260   266   261
```

```
clc;
clear;
% Write matlab program to apply the DCT algorithm to the following
points:
f = [2 3 4 4];
F = [];
C = 1/sqrt(2);
N = length(f);
for u=0:N-1
    i = 0;
    for x=0:N-1
       i = i + f(x+1) * cos(pi * (2 * x + 1) * u / ( 2 * N ));
    end
    F(u+1) = sqrt(2/N) * C * i;
    C = 1;
end
disp(['DCT(f):  ', num2str(F)]);

% ---------------------------------
inverse=[];

for x = 0:N-1
    I = 0;
    for u=0:N-1
        if u == 0
            C = 1 / sqrt(2);
        else
            C = 1;
        end
        i = i + C * F(u+1) * cos(pi * (2 * x + 1) * u / ( 2 * N ));
    end
    inverse(x+1) = sqrt(2/N) * C * i;
end
disp(['Inverse:  ', num2str(inverse)]);
```

Command Window

```
DCT(f):  6.5     -1.5772        -0.5     0.11209
Inverse:  2           3           4           4
```