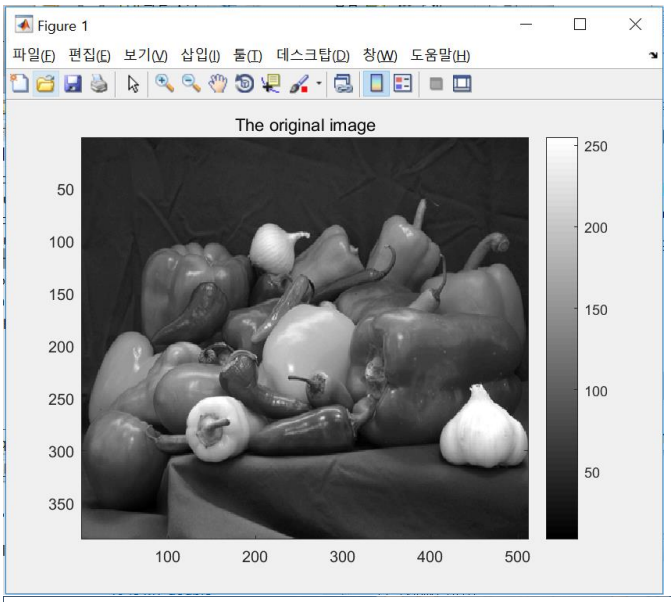
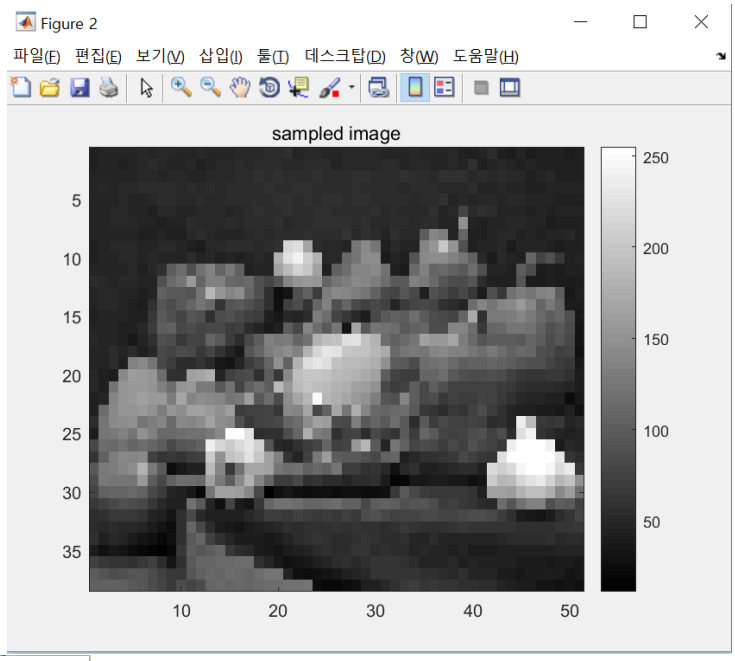
ECE172A
Jiseok Jeong
A14281757

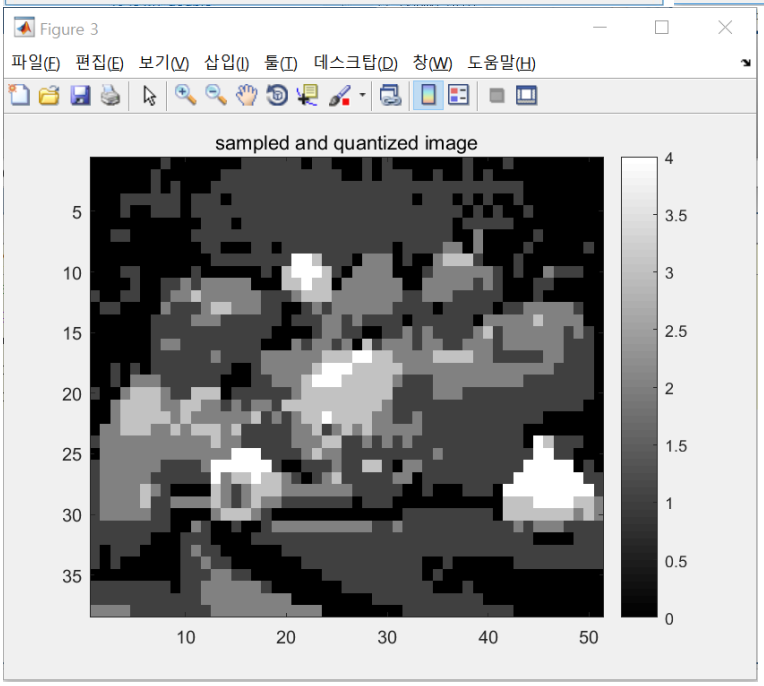
Homwork 3

Academic Integrity Policy: Integrity of scholarship is essential for an academic community. The University expects that both faculty and students will honor this principle and in so doing protect the validity of University intellectual work. For students, this means that all academic work will be done by the individual to whom it is assigned, without unauthorized aid of any kind. By including this in my report, I agree to abide by the Academic Integrity Policy mentioned
above.

Problem 1. Sampling & Quantization (10 points)







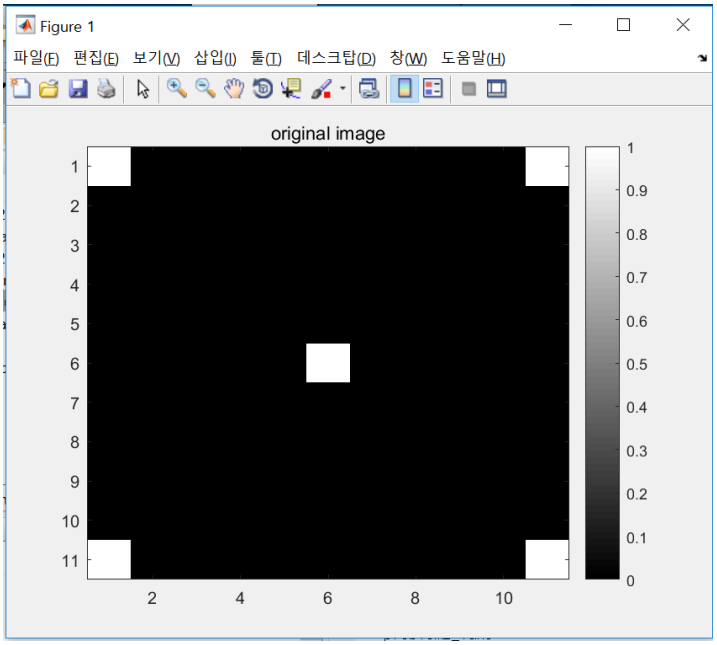
•How would the above function be of use in image compression?

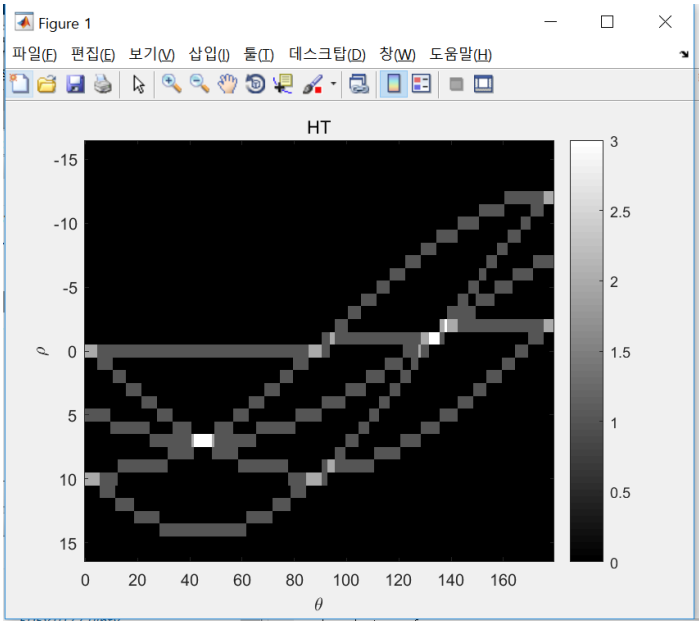
Ans: By sampling, we can reduce the file size of images that help us process these images even faster. On top of that, we can represent the gray levels in an image using quantization.

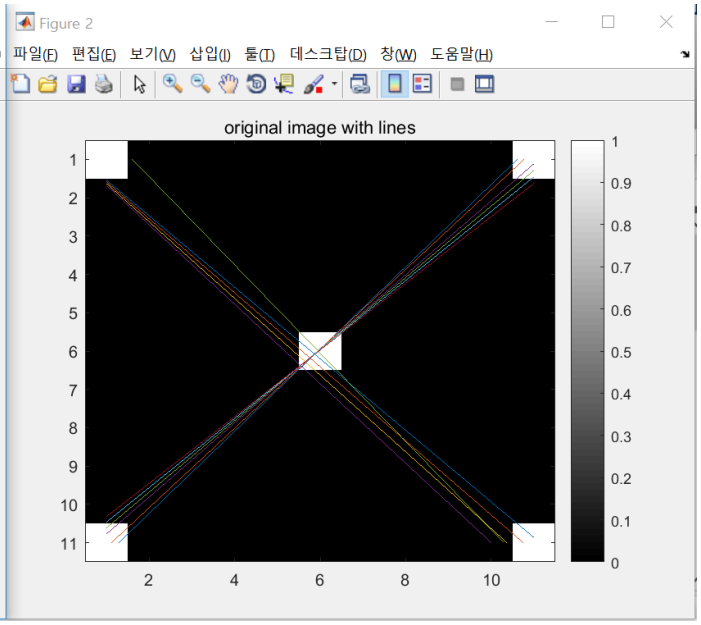
ECE172A
Jiseok Jeong
A14281757

Problem 2. Hough Transform (20 points)

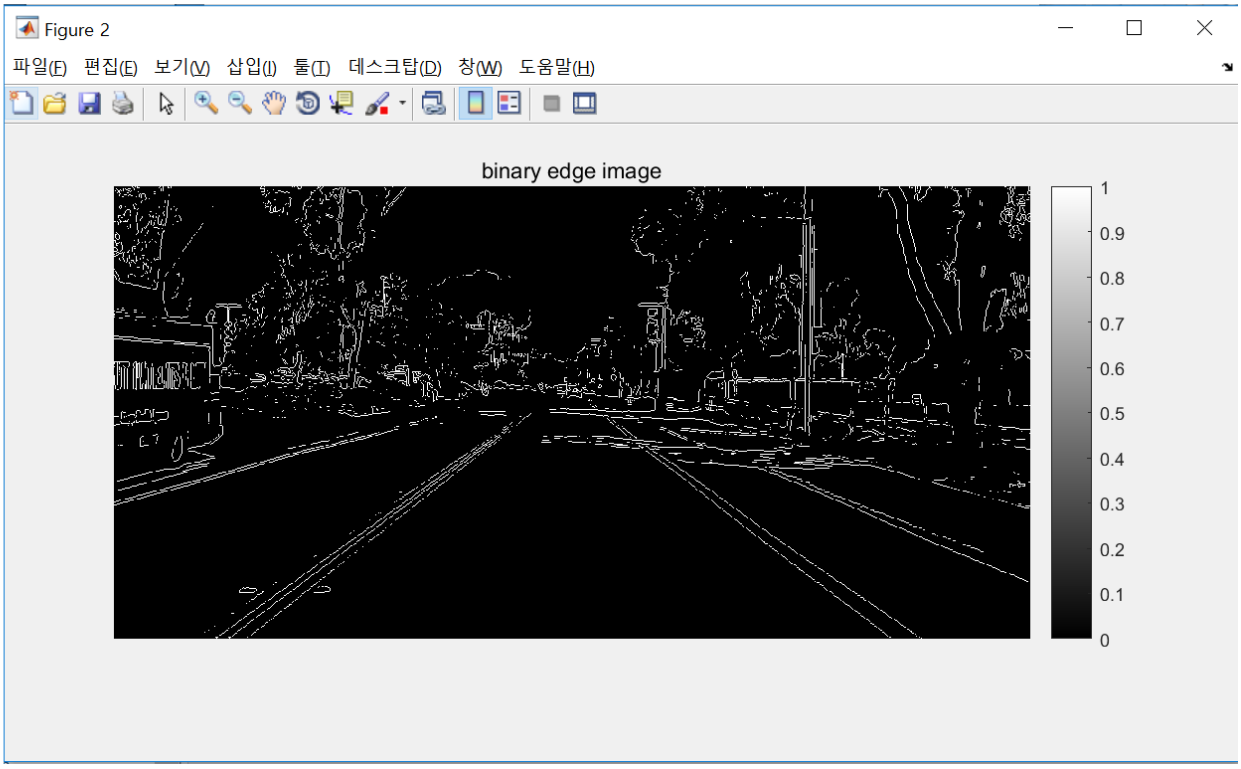
(ii)



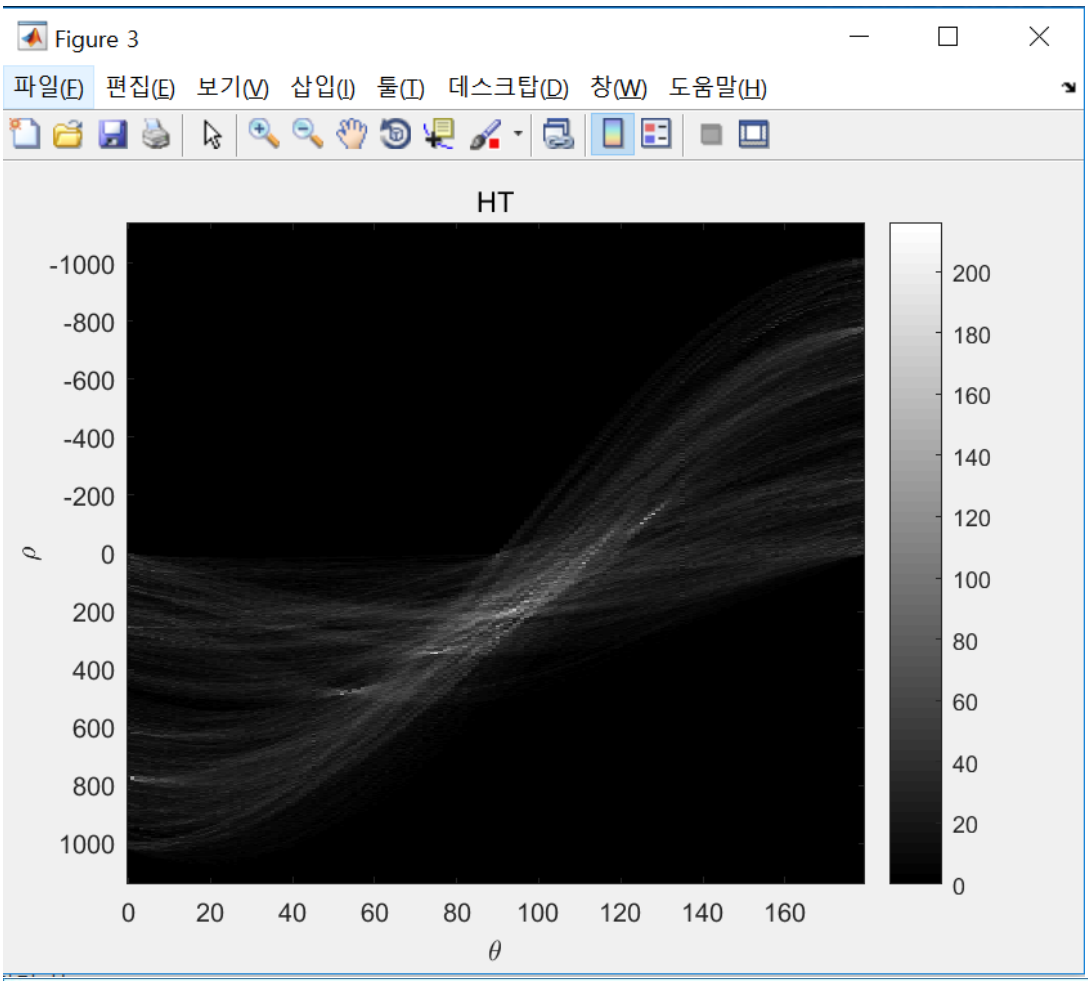


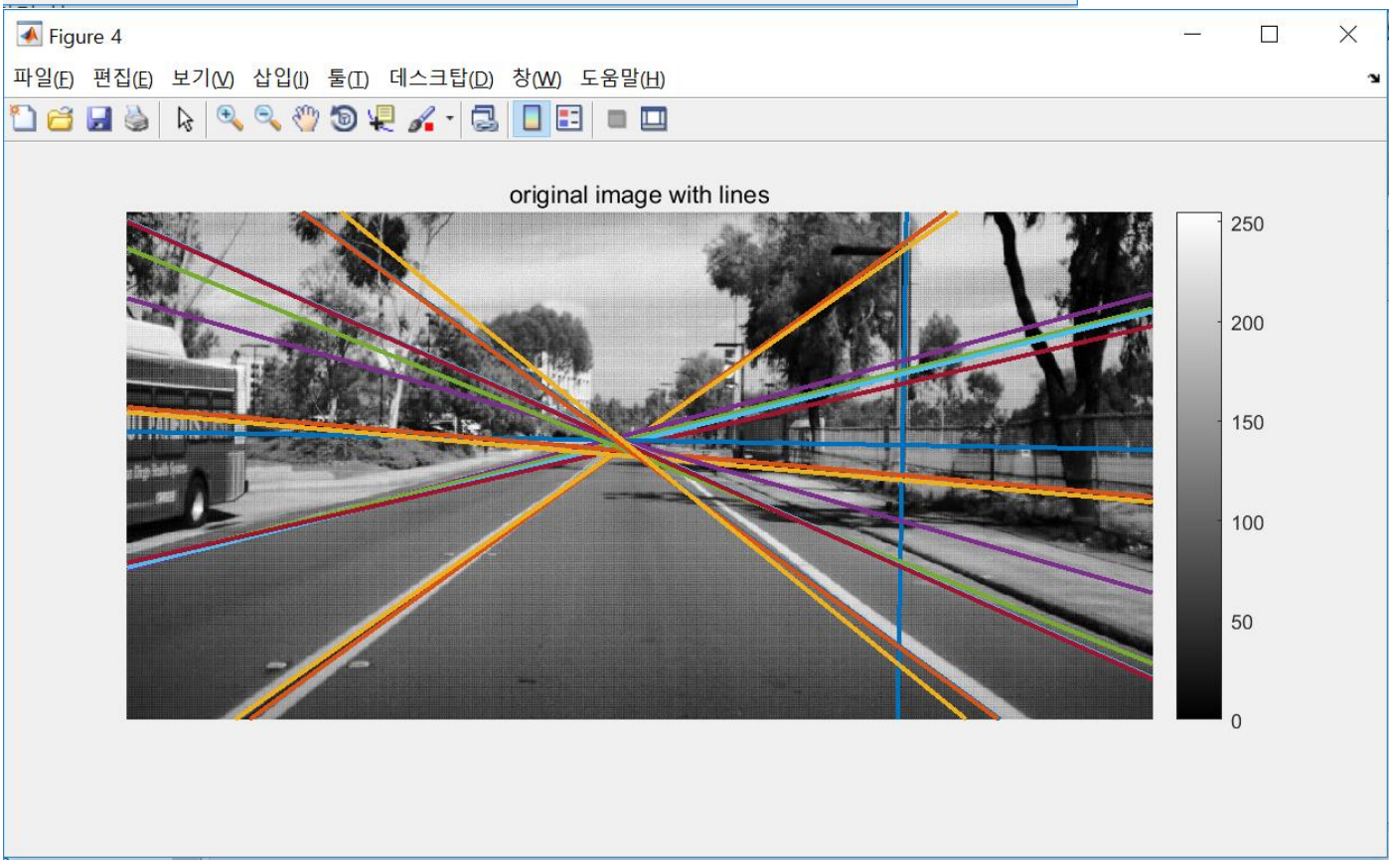


ECE172A
Jiseok Jeong
A14281757
(iii)



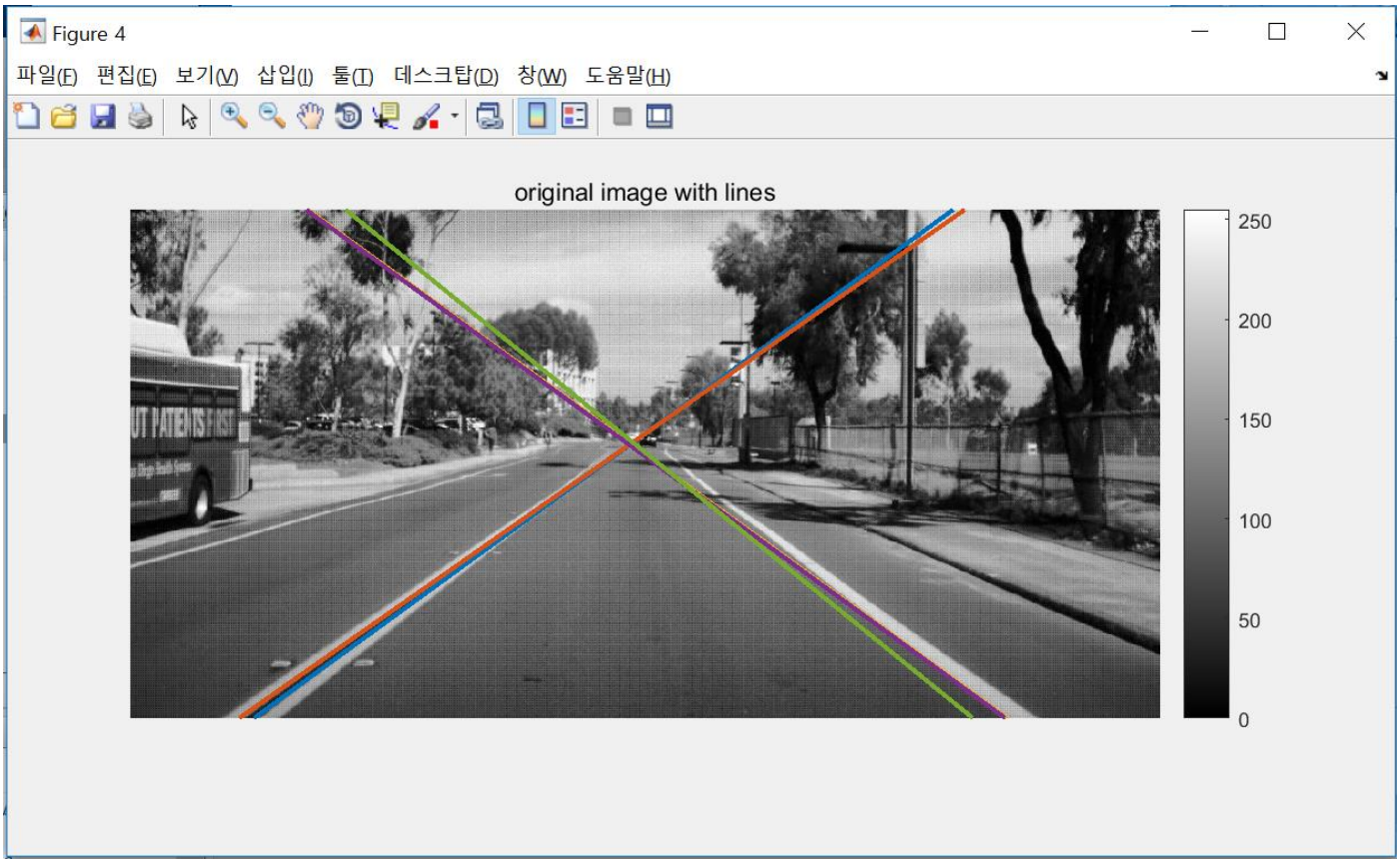
ECE172A
Jiseok Jeong
A14281757





ECE172A
Jiseok Jeong
A14281757

(iv)



As we can guess from the HT of this image,

the range of theta is '*45<=theta<=65, 120<=theta<=140*'

ECE172A
Jiseok Jeong
A14281757

APPENDIX

Problem 1

```
function output_img = sample_quant(input_img)
%% sampling
size_input = size(input_img);
i=1;
j=1;
while(10*i < size_input(1))
      while(10*j < size_input(2))
            sampled(i,j) = input_img(i*10,j*10);
            j=j+1;
      end
      i=i+1;
      j=1;
end
figure(2), imagesc(sampled); colormap(gray); colorbar; title('sampled image'); % show the sampled image

%% quantization
size_sampled = size(sampled);
for i=1 : size_sampled(1)
      for j=1 : size_sampled(2)
            if (sampled(i,j)>=0) && (sampled(i,j)<=50)
                  final(i,j) = 0;
            elseif (sampled(i,j)>=51) && (sampled(i,j)<=101)
                  final(i,j) = 1;
            elseif (sampled(i,j)>=102) && (sampled(i,j)<=152)
                  final(i,j) = 2;
            elseif (sampled(i,j)>=153) && (sampled(i,j)<=203)
                  final(i,j) = 3;
            else
                  final(i,j) = 4;
            end
      end
end
output_img = final;
end
```

%% run a file for probelm 1

```
input=imread('peppers.png');
figure(1), imagesc(input); colorbar; colormap(gray); title('The original image');
output = sample_quant(input);
figure(3), imagesc(output); colormap(gray); colorbar; title('sampled and quantized image');
```

Problem 2

ECE172A
Jiseok Jeong
A14281757

2(i),2(ii)

```
image = zeros(11,11);
image(1,1)=1;
image(1,11)=1;
image(11,1)=1;
image(11,11)=1;
image(6,6)=1;
figure(1); imagesc(image); colormap(gray); colorbar; title('original image');
size_img = size(image);
H = size_img(1);
W = size_img(2);
rho = -round(sqrt(H*H+W*W)):1:round(sqrt(H*H+W*W));
theta = 0:1:179;

A = zeros(length(rho),length(theta));
[y,x] = find(image==1);

for ii=1:length(x)
      rho1 = x(ii)*cos(theta*pi/180)+y(ii)*sin(theta*pi/180);
      rho1 = round(rho1+sqrt(H*H+W*W));
      for iii=1 : length(theta)
      A(rho1(iii),iii)=A(rho1(iii),iii)+1;
      end
end
figure(2),imagesc(theta,rho,A); title('HT'); xlabel('\theta'); ylabel('\rho'); colorbar; colormap gray;

[rho_line,theta_line] = find(A>=3);

figure(3); imagesc(image); colormap(gray); colorbar; title('original image with lines');

hold on;

for iv=1:length(rho_line)
x_i = [1 W];
y_i                                                =                                                (rho_line(iv)-sqrt(H*H+W*W)-
x_i*cos((theta_line(iv)+theta(1))*pi/180))/sin((theta_line(iv)+theta(1))*pi/180);

y_ii = [1 H];
x_ii                                               =                                                (rho_line(iv)-sqrt(H*H+W*W)-
y_ii*sin((theta_line(iv)+theta(1))*pi/180))/cos((theta_line(iv)+theta(1))*pi/180);

x_final=[];
y_final=[];

for k=1 : 2
      if y_i(k)>=1 && y_i(k)<=H
            x_final = [x_final,x_i(k)];
            y_final = [y_final,y_i(k)];
      elseif x_ii(k)>=1 && x_ii(k)<=W
            x_final = [x_final,x_ii(k)];
            y_final = [y_final,y_ii(k)];
```

```
    end

end
plot(x_final,y_final);

end
hold off;
```

2(iii), 2(iv)

```
 close all;

I = imread('lane.png');
I=I(:,:,1);
figure(1); imshow(I); colorbar; title('original image')
image = edge(I,'sobel');
figure(2); imshow(image); colorbar; title('binary edge image')
size_img = size(image);
H = size_img(1);
W = size_img(2);
rho = -round(sqrt(H*H+W*W)):1:round(sqrt(H*H+W*W));
theta = 0:1:179;


A = zeros(length(rho),length(theta));
[y,x] = find(image==1);

for ii=1:length(x)
    rho1 = x(ii)*cos(theta*pi/180)+y(ii)*sin(theta*pi/180);
    rho1 = round(rho1+sqrt(H*H+W*W));
    for iii=1 : length(theta)
    A(rho1(iii),iii)=A(rho1(iii),iii)+1;
    end
end
figure(3),imagesc(theta,rho,A); title('HT'); xlabel('\theta'); ylabel('\rho'); colorbar; colormap gray;

[rho_line,theta_line] = find(A>=0.75*max(A(:)));

figure(4); imshow(I); colorbar; title('original image with lines')
hold on;

for iv=1:length(rho_line)
x_i = [1 W];
y_i = (rho_line(iv)-sqrt(H*H+W*W)-x_i*cos((theta_line(iv)+theta(1))*pi/180))/sin((theta_line(iv)+theta(1))*pi/180);

y_ii = [1 H];
x_ii = (rho_line(iv)-sqrt(H*H+W*W)-y_ii*sin((theta_line(iv)+theta(1))*pi/180))/cos((theta_line(iv)+theta(1))*pi/180);

x_final=[];
y_final=[];
```

ECE172A
Jiseok Jeong
A14281757

```
%% find lines that across the image
for k=1 : 2
    if y_i(k)>=1 && y_i(k)<=H
        x_final = [x_final,x_i(k)];
        y_final = [y_final,y_i(k)];
    elseif x_ii(k)>=1 && x_ii(k)<=W
        x_final = [x_final,x_ii(k)];
        y_final = [y_final,y_ii(k)];
    end

end
%% thresholding angle
if theta_line(iv)+theta(1)>=45 && theta_line(iv)+theta(1)<=65 || theta_line(iv)+theta(1)>=120 && theta_line(iv)+theta(1)<=140
plot(x_final,y_final,'linewidth',2);
end

end
hold off;
```