Name: Amitesh Kumar Sah

NYU ID : N19714360

Option 1: Write a program to perform vector quantization on a gray scale image using 4x4 pixels as a vector. You should design your codebook using all the blocks in the image as training data, using the generalized Lloyd algorithm. Then quantize the image using your codebook. You can choose the codebook size, say, L=128 or 256. If your program can work with any specified codebook size L, then you can observe the quality of quantized images with different L.

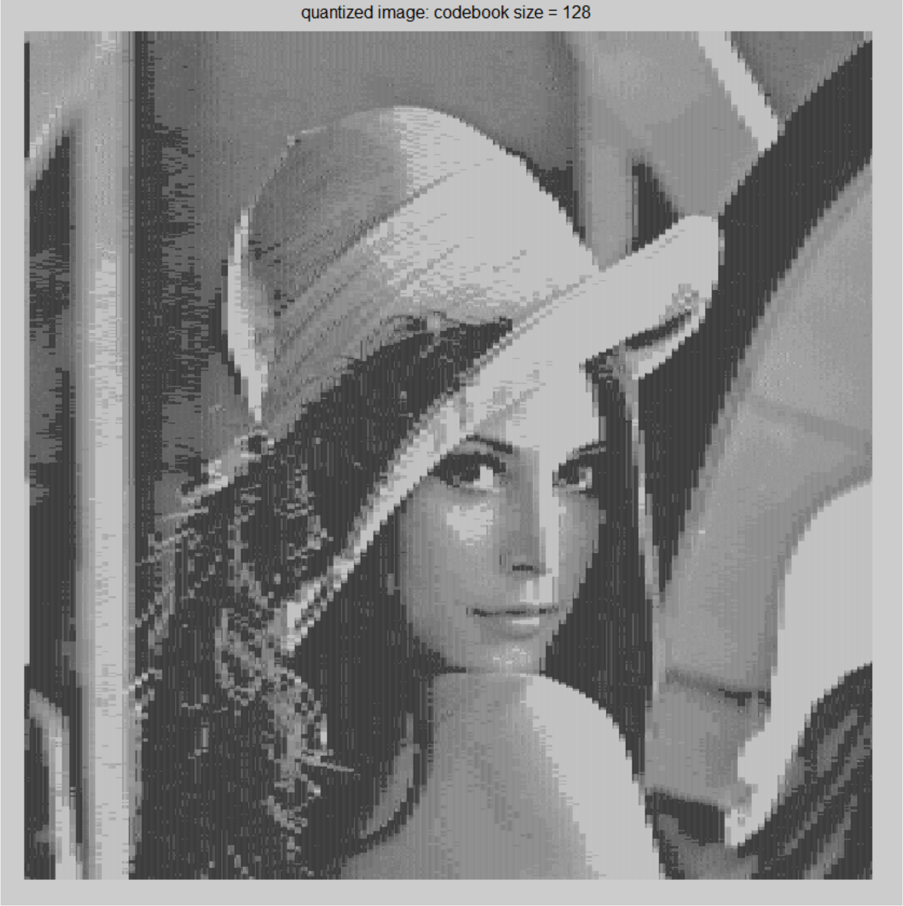
Matlab program:

```
function [ qimg ] = VQ_LBG( img,size_w,size_b )
img = double(img)/255;
[m,n] = size(img);
data1 = zeros(m*n,1);
for i = 1:m
    for j = 1:n
        data1((i - 1)*n + j) = img(i,j);
    end
end
M1 = floor(m*n/size_w);
r = mod(m*n,size_w);
if r > 0
    M1 = M1 + 1;
end
data2 = zeros(M1,size_w);
l = 1;
A = zeros(size_w,1);
r = 1;
for i = 1:m*n
    A(r) = data1(i);
    if r == size_w
        data2(l,:) = A;
        l = l + 1;
        r = 1;
    else
        r = r + 1;
    end
end
code_book = zeros(size_b,size_w);
l = 1;
A = zeros(size_w,1);
r = 1;
for i = 1:size_b*size_w
    A(r) = data1(i);
    if r == size_w
        code_book(l,:) = A;
        l = l + 1;
        r = 1;
    else
        r = r + 1;
    end
end
MIU = zeros(M1,size_b);
t = 1;
 while t == 1
     for i = 1:M1
         B = zeros(size_w,1);
         B = data2(i,:);
         A = zeros(size_w,1);
         A = code_book(1,:);
         tep = 0.0;
         for l = 1:size_w
             tep = tep + (A(l) - B(l))^2;
         end
         r = 1;
         for j = 2:size_b
             A = code_book(j,:);
             temp = sum((A - B).^2);
             if temp < tep
                 r = j;
                 tep = temp;
             end
         end
         MIU(i,r) = 1.0;
     end
     t = 0;
     code_book1 = zeros(size_b,size_w);
     for j = 1:size_b
         for l = 1:size_w
             tep = 0.0;
             for i = 1:M1
                 code_book1(j,l) = code_book1(j,l) + MIU(i,j)*data2(i,l);
                 tep = tep + MIU(i,j);
             end
             if tep > 0
                 code_book1(j,l) = code_book1(j,l)/tep;
             else
                 code_book1(j,l) = 0.0;
             end
         end
     end
     tep = 0.0;
     for j = 1:size_b
         for l = 1:size_w
             tep = tep + (code_book1(j,l) - code_book(j,l))^2;
         end
     end
     if tep/size_b < 0.000001
```

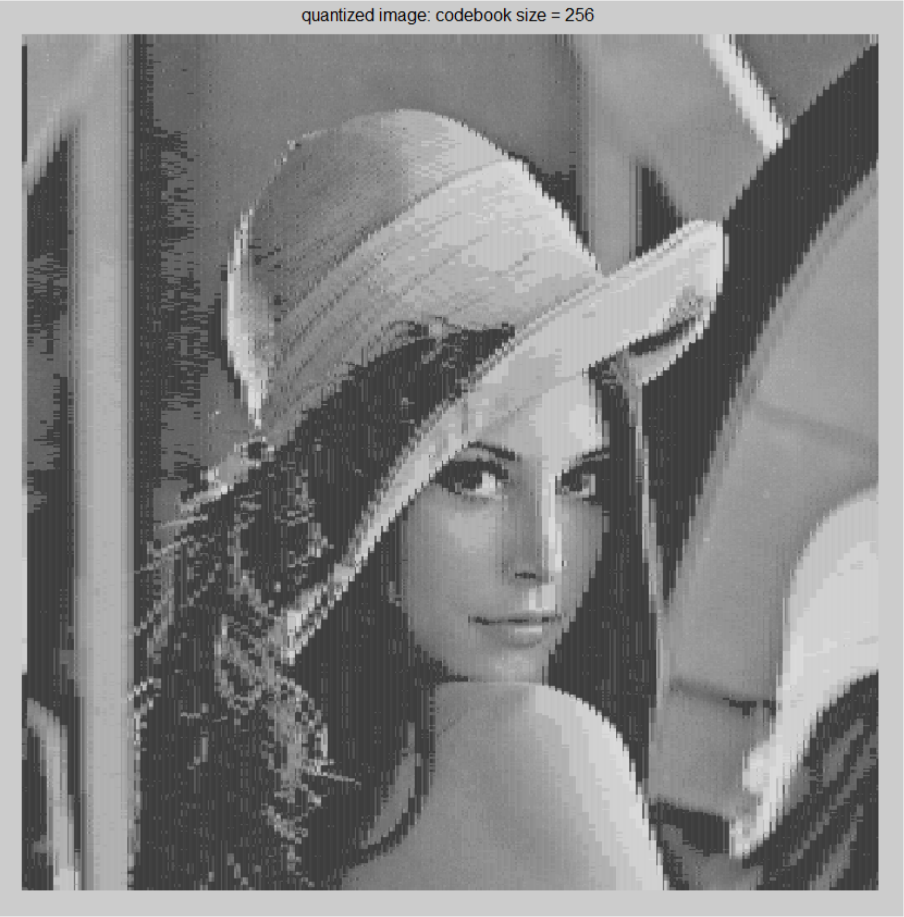
```
         t = 0;
     end
     code_book = code_book1;
 end

 data3 = zeros(M1,size_w);
 for i = 1:M1
     for j = 1:size_b
         if MIU(i,j) == 1
             t = j;
         end
     end
     data3(i,:) = code_book(t,:);
 end

 data5 = zeros(m,n);
 for i = 1:m
     for j = 1:n
         tep = (i - 1)*n + j;
         i1 = floor(tep/size_w);
         if i1 == 0
             i1 = 1;
         end
         j1 = mod(tep,size_w);
         if j1 == 0
             j1 = size_w;
         end
         data5(i,j) = floor(data3(i1,j1)*255);
     end
 end

 qimg=uint8(data5);

end
```

```
>> img=imread('lena512gray.bmp');
>> qimg1=VQ_LBG(img,4,128);
>> figure
>> imshow(qimg1)
>> title('quantized image: codebook size = 128')
>> qimg2=VQ_LBG(img,4,256);
>> figure
>> imshow(qimg2)
>> title('quantized image: codebook size = 256')
```

We can observe that the quality of the quantized image with L=256 is better than the quality of the quantized image with L=128.



quantized image: codebook size = 128



quantized image: codebook size = 256