LNMIIT
The LNM Institute of Information Technology
DEPARTMENT OF ELECTRONICS & COMMUNICATION ENGINEERING

Experiment No. # 6
Transform Based Lossy Compression

---

1) **Objectives:**
   a) Discrete Cosine Transform and it's energy compaction property.
   b) Simulink based image compression.
2) **Software used:**
   a) MATLAB.

A. *Pre-Lab*

a) Read about Discrete Cosine tranform and image compression.
b) practice simulink.

I. Discrete Cosine Transform (DCT)

A. *Theory*

1) J. Proakis and D. Manolakis, Digital signal processing: principles, algorithms, and applications

B. *Procedure*

1) Discrete Cosine Transform: Write a MATLAB code myCompression.m that takes in the image, find the DCT, store the significant transform domain coefficients and discards the nonsignificant coefficients, thus compress the original image in a compact form. Reconstruction of the image is done by following all the steps of compression in reverse manner.
2) Let Im_1 be the gray-scale image of size $N \times N$. The corresponding $2D$-Discrete Cosine Transfrom ImF_1 is given by

$$ImF_1(k;l) = \frac{1}{\sqrt{2N}}\beta(k)\beta(l)\sum_{i=0}^{N-1}\sum_{j=0}^{N-1} Im_1(i,j)\cos\left(\frac{\pi(2i+1)k}{2N}\right)\cos\left(\frac{\pi(2j+1)l}{2N}\right)$$

where

$$\begin{aligned}\beta(u) &= \frac{1}{\sqrt{2}}, && u = 0\\ &= 1, && u > 0\end{aligned}$$

   a) Use the above expression and write down the expression for 2D-DCT with N = 8.
   b) Divide the image for ex. 'cameraman' into non-overlapping blocks of size 8×8. For each of them compute the 2D-DCT.
   c) Now starting from the bottom-left corner of each block that is obtained from 2D-DCT, knock-off the coeffcients gradually one by one, or in other words neglect the higher frequency terms, and for each removal of coefficient reconstruct an approximation to the original 8×8 image

block via 2D-IDCT. Put together all the blocks and the approximation to the original image. (apply zig-zag pattern to scan the image)

d) Perform the above procedure for four cases.
Case 1: Keep top 48 coefficients out of 64.
Case 2: Keep top 32 coefficients out of 64.
Case 3: Keep top 16 coefficients out of 64.
Case 4: Keep top 8 coefficients out of 64.

e) Calculate the mean square error $\epsilon = |Im_1 - \hat{I}m_1|^2$, where $\hat{I}m_1$ is a reconstructed approximation.

f) Plot the graph of Mean Squared Error ϵ versus Compression Ratio ρ. Where

$$\rho = \frac{N^2 - \text{Total No. of knocked off coefficents}}{N^2}$$

3) **Observation:**
   a) Use inbuilt DCT and IDCT function to compress and reconstruct any input image. Also by knocking off half the pixels of compresed image show the image artifact effects(at edges/borders in the image).
   b) Write down the myCompression.m function using the expression and cross verify with the inbuilt DCT function.
   c) Divide the whole image in 8×8 blocks use above function to compress the whole image with top 32 coefficents and find the mse. (Is artifact still visible ?)
   d) Repeat the above steps for given four different cases and plot the graph between mse and compression ratio.
   e) Repeat the experiment in simulink.
4) **Conclusion:** Conclude the experiment.

II. Compression in Simulink

1) Open simulink and create a model file with .slx extension.

2) Transfer the image data from command window to the above generated model file.

3) Write down or utilize myCompression.m. in model file to compress and reconstruct the image.

4) Write a generalized model which takes dividing block size(for ex. 4,8,16), Total no. of knock off coefficents(for ex. Top 16, 32, 48) and perform compression and reconstruction.

Well Done
