NTNU
Norwegian University of Science And Technology
Faculty of Information Technology,and Electrical Engineering
Department of Electronic Systems

TTT4135 MULTIMEDIA SIGNAL PROCESSING
Assignment 1

Note

The assignment consists of two parts. First part deals with theoretical problems while the second one deals with practical problems related to JPEG and JPEG 2000. Both parts should be done in groups made up of two or three students.
Please write the solutions in English or Norwegian, and submit it electronically on BlackBoard in "Arbeidskrav/Mandatory Assignment 1". Please provide matlab code (where applies) and place it in the corresponding section of the final document. DEADLINE FOR HANDING IN THE SOLUTIONS IS February 14, 2022 (23:59).
TIP: Use the publish tool in Matlab to generate a PDF of the code.

Evaluation criteria

The Assigmnet work counts 20% of the final grade. The grading used will be between A-F as per NTNU standard. The Assignment consist of two parts. The problems have varying weights.The different weights are stated at the start of each problem. The upper achievable score is 100.
TIP: Use the publish tool in Matlab to generate a PDF of the code.

Part I (55 points)

1 Still image compression

We will here consider a simple example of a still image compression that exemplifies the operation of the JPEG image compression standard. Table 1 shows an 8x8 pixels section of a monochrome (grayscale) image. You will need a computer with Matlab installed for this (or, alternatively, it can be done by hand - but that may take a while).

124	125	122	120	122	119	117	118
121	121	120	119	119	120	120	118
126	124	123	122	121	121	120	120
124	124	125	125	126	125	124	124
127	127	128	129	130	128	127	125
143	142	143	142	140	139	139	139
150	148	152	152	152	152	150	151
156	159	158	155	158	158	157	156

Table 1: 8x8 pixel values from a picture

a) Apply the 2-dimensional DCT (Discrete Cosine Transform) on the data block. Use the function *DCT2* in MATLAB. (10 points)

16	11	10	16	24	40	51	61
12	12	14	19	26	58	60	55
14	13	16	24	40	57	69	56
14	17	22	29	51	87	80	62
18	22	37	56	68	109	103	77
24	35	55	64	81	104	113	92
49	64	78	87	103	121	120	101
72	92	95	98	112	100	103	99

Table 2: Quantization table (q_{ij})

The quantization table is given in Table 2. The precision/resolution of the transform coefficients through quantization is done as follows:

$$k_{ij} = \left\lfloor \frac{l_{ij}}{q_{ij}} + 0.5 \right\rfloor$$

k_{ij} are here the transform coefficients after quantization, and $\lfloor x \rfloor$ is the largest integer smaller or equal to x.

b) Find the coefficients k_{ij}. What do we gain by encoding/quantizing the image in the transform domain rather than encoding/quantizing the original image pixels directly? (10 points)

In a real-world source coding scheme we would typically use entropy coding for the quantized transform coefficients, but we do not consider that in this

exercise. In the receiver we would either way have to do an inverse DCT in order to reconstruct an approximation of the encoded image.

c) Perform the reconstruction of the quantized transform coefficients by using the Matlab command *IDCT2* and find the PSNR (Peak Signal-to-Noise Ratio) for the 8x8 block (Remember to dequantize the quantized transform coefficients before applying IDCT2). (10 points)

2 Prediction

Assume that $x(n)$ is a weakly stationary process with autocorrelation given by $R_x(k)$, $R_x(0) = \sigma_x$, $R_x(1) = a$, $R_x(k) = 0$ for $|k| > 1$.

a) Find the optimal linear predictor for $x(n)$. Use a second order predictor. (5 points)

3 Compression basics

a) What are the fundamental steps for lossless and lossy multimedia compression, respectively? (5 points)

b) What are the principles behind these steps? (5 points)

c) What are the most used algorithms in each step? (5 points)

d) If it is lossy compression, how do we evaluate the compression quality? (5 points)

Part II (45 points)

4 JPEG and JPEG2000

a) Give a short overview of the most important differences between JPEG and JPEG 2000. Discuss this in two categories: (5 points)

- Differences in functionalities (*f.ex. scalability, error robustness/resilience)*
- Differences in coding techniques (*f.ex. transform, image partitioning, entropy coding*)

b) This part of the exercise uses supplied executables.These can be found in the folder "Program" on Blackboard, along with all the other needed material. These excuatables can be run from the command prompt on windows, and on the Terminal on mac. The command prompt can be accessed through "Start" and entering "cmd" in the "Search:" field. On Mac u can find the terminal by searching in the Launchpad. If you are unfamiliar with navigating on the computer/mac using the command line/Terminal - ask one of your fellow students.

- For windows users the executables are as follows:
  - *jpeg_encoder.exe* A simple JPEG encoder. Instructions are found by running the program without supplying any parameters/switches.
  - *jpeg_decoder.exe* The decoder corresponding the the above mentioned JPEG encoder.
  - *j2000_encoder.exe* A JPEG2000 encoder that implements most of the functionalities that are specified in the standard. Instructions are found by using the command line parameter '-u'. A collection of the most important parameters are found in the supplied pdf file [1].
  - *j2000_decoder.exe* The corresponding decoder. For usage instructions, see pdf file[2]. or use the '-u' parameter. Notice that you might encounter an error message that mentions the parameter '-Cno_speedup'. If this occurs, include that parameter together with the others you are using.
- For Mac user, use the following:
  - for JPEG: we use Independent JPEG Group's JPEG software, lib-jpeg Turbo. This provides two programs, cjpeg to compress an image file into JPEG format, and djpeg to decompress a JPEG file back into a conventional image format. More information of using the excutable can be found using the following link: `https://github.com/libjpeg-turbo/libjpeg-turbo/blob/main/usage.txt`

[1] JPEG2000ENCODER - excerpt of usage instructions.pdf
[2] JPEG2000DECODER - excerpt of usage instructions.pdf

  - OpenJPEG:This is an open source JPEG2000 codec written in C, and uses *opj_compress* for encoding, *opj_decompress* decoding. More information of using the excutable can be found using the following link: `https://github.com/uclouvain/openjpeg/wiki/DocJ2KCodec`
  - To run the jpeg/jpeg2000 on Mac u need to install homebrew which can done by writing the command on the following website in the terminal:`https://brew.sh/`. Then when hombrew is installed, u can easily download jpeg and jpeg2000 using homebrewe using the following commands: "brew install jpeg" for jpeg, and "brew install openjpeg" for jpeg2000. If everything is installed, then you can start running the encoders/decoders using the commands as described further down.
  - in case the installation fails, check documentation about the codecs for other installation methods.

- These tools are used to encode the supplied images. The results should be compared both perceptually and through rate/PSNR results. Example of basic usage of the provided executables for windows:
  - JPEG encoding the raw file "cafe.pgm" using quality level 40 (the encoded JPEG file is called "cafe_N40.jpg"): `jpeg_encoder -quality 40 cafe.pgm cafe_N40.jpg`
  - Decoding the above encoded JPEG file: `jpeg_decoder cafe_N40.jpg cafe_jpeg_decoded_N40.pgm`
  - Encoding "cafe.pgm" using JPEG2000 at 0.4 bits per pixel (encoded file is called "cafe_04bpp.jp2"): `j2000_encoder -i cafe.pgm -o cafe_04bpp.jp2 -rate 0.4`
  - Decoding the above JPEG2000 encoded file to output file called "cafe_jp2_04bpp.pgm": `j2000_decoder -i cafe_04bpp.jp2 -o cafe_jp2_04bpp.pgm -Cno_speedup`
- Example of basic usage of the executable on mac:
  - Encoding cafe.pgm using JPEG with quality level 40 (encoded file is called "cafe_40_jpeg.jfif") `cjpeg -outfile cafe_40_jpeg.jfif -quality 40  cafe.pgm`
  - Decoding cafe_40_jpeg.jfif (decoded file is called cafe_40_jpeg_d.pgm) `djpeg -outfile cafe_40_jpeg_d.pgm cafe\_40\_jpeg.jfif`

  - Encoding "cafe.pgm" using JPEG2000 with compression rate of 40 (encoded file is called "cafe_40.j2k"):
    `opj_compress -i cafe.pgm -r 40 -o cafe_40.j2k`
  - Decoding cafe40.j2k (decoded file is called "cafe_40_de.pgm")
    `opj_decompress -i cafe_40.j2k -o cafe_40_de.pgm`

c) Try to describe the differences between the coding error (distortion) that is introduces when coding at lower rates using the two coders. How can you explain these different errors? (20 points)

d) Calculate (in matlab/python)and Make a table where PSNR is listed for the two coders (for the same images) at different rates/compression(for example, quality level of JPEG is 5, 20, 40, 60 and 80). (20 points)

IMPORTANT NOTE: (only for windows users)

- When comparing the performance of the two coders you have to use the same number of bits per pixel (BPP).
  In the JPEG2000 coder software, this is easily done by directly setting the number of BPP from the command line. In the JPEG coder software BPP is not directly set, however a quality level parameter is used. To find out how many BPP the selected quality level represents you have to perform the compression and compute the BPP for the JPEG-picture as:

$$\text{BPP} = \frac{\text{file size for the compressed image (in number of bits)}}{\text{number of pixels in the picture (x-pixels * y-pixels)}}$$

  Therefore, you may have to try several values for the quality level in order to obtain approximately the same BPP as for the JPEG2000 compressed image.
  Relevant values for BPP might be in the range 0.3 - 3

- Notice that in order to view the .pgm files (originals and the resulting decoded images), the supplied freeware program Irfanview (*iview385.exe*) can be used.

- THE ENCODED JPEG2000 IMAGES CANNOT BE DECODED AND DISPLAYED PROPERLY BY THE IRFANVIEW PROGRAM. IT

IS NECESSARY TO DECODE TO THE PGM FORMAT MANUALLY BEFORE VIEWING THE RESULT IN THE IRFANVIEW PROGRAM.