Lab 6 – JPEG Compression

Goal: Introduction of principles of the JPEG baseline coding system.

Introduction

The JPEG standard provides a powerful compression tool used worldwide for different applications. This standard has been adopted as the leading lossy compression standard for natural images due to its excellent compression capabilities and its configurability. In this lab we will present the basic concepts used for JPEG coding and experiment with different coding parameters.

JPEG baseline coding algorithm (simplified version) [1]

The JPEG baseline coding algorithm consists of the following steps:

1. The image is divided into 8×8 non-overlapping blocks.
2. Each block is level-shifted by subtracting 128 from it.
3. Each (level-shifted) block is transformed with Discrete Cosine Transform (DCT).
4. Each block (of DCT coefficients) is quantized using a quantization table. The quantization table is modified by the "**quality**" factor that controls the quality of compression.
5. Each block (of quantized DCT coefficients) is reordered in accordance with a zigzag pattern.
6. In each block (of quantized DCT coefficients in zigzag order) all trailing zeros (starting immediately after the last non-zero coefficient and up to the end of the block) are discarded and a special End-Of-Block (EOB) symbol is inserted instead in order to represent them.

Remark: In the real JPEG baseline coding, after removal of the trailing zeros, each block is coded with Huffman coding, but this issue is beyond our scope. Furthermore, in each block all zero-runs are coded and not only the final zero-run. Each DC coefficient (the first coefficient in the block) is encoded as a difference from the DC coefficient in a previous block [5, 6].

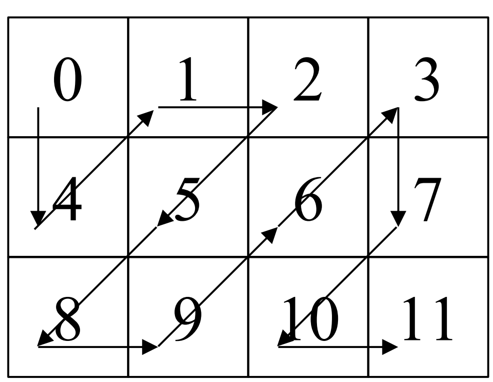
Preliminary report

Answer the following questions:

1. Why was DCT chosen as transform domain for JPEG? What are the advantages of DCT over other transforms, e.g. DFT? *Hint*: how does DFT and DCT treat a periodic extension of a signal and what are consequences of these extension methods on image border processing?

2. (*Programming*) Implement in python the Discrete Cosine Transform of a 1-D real signal of length N (N is even) via Discrete Fourier Transform. You have to write a function ***def dct_new(x_sig)*** that returns *X_dct*, where the input parameter is *x_sig* – a 1-D real signal of length N (N is even) that has to be transformed, and the output parameter is *X_dct* – the DCT of x_sig.
   The use of a ***scipy.fftpack.dct*** command (or other direct functions) is prohibited. You are allowed to use an ***scipy.fftpack.fft*** command.
   <u>*Hint*</u>: see in [6] the derivation of a formula that allows finding an N-point DCT using N-point DFT.

3. (*Programming*) Implement zigzag ordering of a pattern (matrix) of arbitrary size MxN. You have a MxN pattern and you put sequential numbers (indices) in its entries filling row by row. Now you have to extract the indices from the pattern in zigzag order. The order of zigzag must be according to the following figure – starting at the upper left corner and moving down (this ordering is different from the ordering shown in Figure 8.12 of [1]).



In this Figure the parameters $\{M = 3,\ N = 4\}$ were used.

You have to write a function ***def zigzag(M, N)*** that returns *zig* parameter, where the input parameters are M – number of rows in a pattern, N – number of columns in a pattern, and the output parameter is *zig* - the row vector (of the size $1 \times (MN)$) that contains the indices of a pattern in zigzag order.

For example, for the Figure above the function call

>> zig = zigzag(3, 4) ;

has to yield the following vector: Zig = [0 4 1 2 5 8 9 6 3 7 10 11].

Test your program for small values of M and N. Make sure that your program works correctly for the cases $\{M = 4,\ N = 4\}$, $\{M = 8,\ N = 8\}$ and $\{M = 16,\ N = 16\}$.

Description of the experiment

Open the Jupyter notebook supplied for Lab 6 and follow the instructions, use image of your choice.

Final report

Submit the results of testing and demonstrations from the previous section of the experiment, with the image of your choice. Explanation your results in each step and submit all the graphs and images.

Answer all the questions appeared in the final report in your Jupyter notebook - in the end of each section as <u>Raw NBConvert</u> cell, save the files as <u>PDF</u> format.

Use: File -> Download as -> PDF. Make sure to submit all your plots and results.

References

[1] R. C. Gonzalez and R. E. Woods, *Digital Image Processing*. Prentice-Hall, Inc., 2002, Second Edition (Library Dewey number 621.368 GON).

[2] R. C. Gonzalez and R. E. Woods, *Digital Image Processing*. Addison-Wesley Publishing Company, Inc., 1992 (Library Dewey number 621.368 GON).

[3] A. K. Jain, *Fundamentals of Digital Image Processing*. Prentice-Hall, Inc., 1989 (Library Dewey number 621.368 JAI).

[4] G. K. Wallace, "The JPEG still picture compression standard," *IEEE Trans. Consumer Electron.*, vol. 38, no. 1, pp. xviii-xxxiv, February 1992.

[5] G. K. Wallace, "The JPEG still picture compression standard," *Commun. ACM*, vol. 34, no. 4, pp. 30-44, April 1991.

[6] http://fourier.eng.hmc.edu/e161/lectures/dct/node2.html