Lab 4 - Image Processing EEE412

Image compression

Report is due **14 days** from the date of running this lab

Objectives:

1- To familiarize with the concepts of image compression.
2- To develop some basic image compression tools.

Download:

Download "Lab 4 material.rar", and load them into your Matlab workspace the following Matlab functions: `fsize, entropy_enc, entropy_dec` and `Arith07.`

Matlab functions:

The following are some built-in *Matlab* functions that might be used in this session:

```
DCT2, IDCT2, ZIP
```

Hint: read the help about each of the previous functions and any other function you might use. Some Matlab functions have a section describing the *Algorithm(s)* they use; it is worth reading this section.

Flexibility in a design[1]

"The flexibility in a design is important, in fact it provides the design with room to grow, to cater for requirement changes during development and for additional requirements that the users will want in the future. Putting the right amount of flexibility into a design is an important part of the act of designing: it requires human creativity.

During the design process, the designer has to make choices about the construction he/she is creating and about the building blocks he will be using. The choice of these building blocks determine part of the flexibility that will be exhibited by the final product. As an example, if I order a wall, the supplier can deliver a concrete wall or a brick wall. Both walls fit the

[1] The "Flexibility in a design" section is from "On the Flexibility of Programming Languages" by Pieter J. Schoenmakers

requirements, yet if I change my mind later on, e.g., because I want windows in my wall, adding windows to the brick wall will be easier than to the concrete one. The bricks offer more flexibility than the concrete. As another example, if a problem needs to be solved by writing a Matlab program, some code needs to be written as the solution. The code is designed by the programmer, and the flexibility of the solution is determined by its design. If the code is packed into a single function, the code can only be applied to solve that particular problem. If the solution is split in several sub-solutions to sub-problems, and the code split in various functions, each of which solves a sub-problem, those functions may later be applied to solve the same sub-problem in other bigger problems. But differently, while the design of the solution is not different, making the sub-solutions accessible makes them usable for solving other problems. Multiple functions exhibit more flexibility than a single function."

Tasks:

1. Huffman compression and decompression (**30'**)

（1） Write a function of Huffman compression and a function of huffman decompression.
function [`Huffman_dictionary, CR, imc`] = Huffman_compress (im),
where ***im*** is an input image, ***imc*** is a output image with Huffman compression, ***CR*** is the compression ratio, and ***`Huffman_dictionary`*** `is the dictionary by the huffman compression. The input image` ***`im`*** `is encoded by 8-bit per pixel.`

（2） Write a function of Huffman decompression
function [`imo`] = Huffman_ decompression (`Huffman_dictionary, imc`),
where the parameters are the same meaning as the task (1).

（3） Use the matrix $\mathrm{Qmat}_{\mathrm{NxN}}$ (given in equation (1), N=16) as the input im in the task (1), run your above programs. Please print all of the outputs and compare imo to im.

2. Image compression (**30'**)

Write a Matlab function which compresses an image `im`. The function should have the following declaration:

```
function [rate] = compress_im(im, Qmat, QP, N, file_name);
```

This function should do the following:

a) Evaluate the two-dimensional DCT of all the N X N non-overlapping blocks of the image `im`.

b) To convert the floating point numbers of the 2D-DCT into integer numbers, quantize each N X N block using the following formula

$$round\left(\frac{b_{ij}}{Sq_{ij}}\right)$$

Where b_{ij} is the i[th] row and j[th] column of the N X N 2D-DCT transformed block, whereas, q_{ij} is the element of the quantization matrix $\mathrm{Qmat}_{\mathrm{NxN}}$ (given in equation (1)). And S is a scalar value given by equation (2).

c) Entropy/symbol encode all the integer indexes of the 2D-DCT coefficients of the image; use the provided arithmetic-encoder[2] for this.

d) Save the entropy encoded vector as `file_name`.

e) Using the function `fsize` evaluate the bit rate (in terms of bit per pixel [bpp]) of the compressed image and return it as `rate`.

f) What are the main differences between this compression engine and Jpeg ?

3. Image decompression (**15**')

Write a Matlab function which decompresses the file `file_name` and return the corresponding image `imo`. The function should have the following declaration:

```
function [imo] = decompress_im(Qmat, QP, N, file_name);
```

This function should invert all steps in task 1 to produce the decompressed image `imo`.

4. Rate-Distortion (RD) Performance evaluation (**25**')

Using the developed coded on image `Lenna512.bmp`, and `N=16`, fill the following table:

QP	**1**	**15**	**29**	**43**	**…**	**…**	**99**
Rate [bpp]							
PSNR [dB]							

Plot the PSNR versus bit rate curve for the image `Lenna512.bmp`; it is worth noticing that this curve is wrongly and commonly called RD curve in many resources.

Compare the Rate-PSNR performance of your codec with that of Jpeg. For the latter codec you could use the Matlab command `imwrite` with proper parameters.

Compare the performance of these codecs objectively and subjectively.

[2] http://en.wikipedia.org/wiki/Arithmetic_coding

Evaluate the Rate-PSNR curve when `N=8` and compare the performance with the one obtained with `N=16`, which one is better, and why ?

Propose some approaches to enhance the RD efficiency of your codec!

$$\mathrm{Qmat}_{\mathrm{NxN}} = \begin{bmatrix} 1 & 2 & 2 & 4 & 4 & 4 & 4 & 8 & \cdots \\ 2 & 2 & 2 & 4 & 4 & 4 & 4 & 8 & \\ 2 & 2 & 2 & 4 & 4 & 4 & 4 & 8 & \\ 4 & 4 & 4 & 4 & 4 & 4 & 4 & 8 & \\ 4 & 4 & 4 & 4 & 4 & 4 & 4 & 8 & \\ 4 & 4 & 4 & 4 & 4 & 4 & 4 & 8 & \\ 4 & 4 & 4 & 4 & 4 & 4 & 4 & 8 & \\ 8 & 8 & 8 & 8 & 8 & 8 & 8 & 8 & \\ \vdots & & & & & & & & \ddots \end{bmatrix} \quad (1)$$

$$S = \frac{100 - QP}{50}, QP > 50$$
$$S = \frac{50}{QP}, QP \le 50 \quad (2)$$

Lab Report

Write a **short** report which should contain a **concise description** of your results and observations. **Include** listings of the **Matlab scripts** that you have written. **Describe each of the images** that you were asked to display.

Submit the report electronically and a hardcopy version into the collecting box beside the office EB310 (Hand written reports are not accepted).

This page last modified on 2017-11-08 12:05 AM