Homework #3

Due Date: 11:59pm Monday 7 April 2014

In this homework you will write a C++ program that can compress and decompress images using the Arithmetic Coding algorithm. Your program will read the image on `stdin` and outputs the resulting image on `stdout`, and get in the required operation on the command line.

Please present a report containing your answers as well as a zip file containing all your code.

1. [3 points] Implement the functions `binary_to_decimal()` and `decimal_to_binary()` to convert to/from decimal fractions and their binary representations. Test your functions using simple values to make sure it's working.

   You can use the `bitset` container from the STL if you want, but it's not necessary. For more information, check this page http://www.cplusplus.com/reference/bitset/bitset/.
2. [4 points] Implement the function `encode()` to encode an image using arithmetic coding and output the total number of bits required on the first line, followed by the tags for every block of symbols, one per line. Try the function on the small images `msg1.pgm`, `msg2.pgm` and `msg3.pgm` to make sure it's working correctly.
3. [4 points] Implement the function `decode()` to decode an image encoded using arithmetic coding and output the corresponding PPM image. Note that you will need to pass in the image metadata i.e. the number of rows, columns, and maximum value on the command line, since these are not included in the encoded message. Try the function on the output of encoding the small images `msg1.pgm`, `msg2.pgm` and `msg3.pgm` to make sure it's working correctly.
4. [3 points] Try your encoder on the four 256x256 images shown in class (sena, sensin, earth, omaha) and compute the compression ratio and number of bits required when using different block sizes (number of symbols per tag) e.g. values for m = 1, 2, 3, 5, 10. Do you get similar numbers to the numbers given in the book? Comment on your answer.

Command Line

You need to modify the main file `hw03.cpp` to include the required functionality. Your program should be named hw03, and should be called as follows:

- To convert from binary to decimal fractions:

  ```
  ./hw03 -bin_to_dec 0x80000000
  ```

  this will output the decimal representation of the binary fraction 0.11 which should be 0.5. Note that the MSB of the binary fraction is the first bit on the input i.e. bit 31, followed by bit 30, … and any bit not specified in the input is assumed to be 0.

- To convert from decimal to binary fractions:

  ```
  ./hw03 -dec_to_bin 0.5
  ```

  this will output the binary representation of the decimal fraction 0.5 which should be 0x80000000.

- To encode an input image and print out the total number of bits required on the first line followed by the binary representation of the tags, one per line:

  ```
  ./hw03 -encode m < input.ppm
  ```

  where *m* is the block size (number of symbols per block), the input image is called `input.ppm` and the output is written to `stdout`. For example, to encode `msg1.pgm` using 3 symbols per block, you could run:

  ```
  ./hw03 -encode 3 < msg1.pgm
  ```

- To decode an image encoded above:

  ```
  ./hw03 -decode m xres yres max < image.enc
  ```

  where *m* is the block size (number of symbols per block), the encoded image is called `image.enc` and the original image had xres columns, yres rows, and maximum intensity max. For example, to decode the encoded version of `sena.pgm`, using 3 symbols per block, you could run:

  ```
  ./hw03 -encode 3 < sena.pgm | ./hw03 -decode 3 256 256 255
  ```

Instructions

- All code should be implemented in C++ under Linux.
- Please submit your homework in one zip file named as follows: *CMPN206.HW##.FirstName.LastName.zip*, so for example if your name is Mohamed Aly and this is homework #1, then the file name should be *CMPN206.HW01.Mohamed.Aly.zip*.
- Please include all your code and sample output in the zip file, with a README file to explain what you did. Failure to follow these instructions will cause deductions from your grade.
- You are allowed to discuss the problems among yourselves. However, **copying** any part of the code will result a grade of **ZERO**. No exceptions.

Grading

- 14 points: requirements above
- 1 point: submission instructions