

JPEG and JPEG2000 Image Compression

ENEE631 Digital Image Processing

Project 1

Due: 07 April 2011

JPEG (Joint Photography Experts Group) and JPEG 2000 are two popular image compression standards. The JPEG standard exploits ideas from DCT transform, differential coding, and entropy coding. Users can select a quality factor to obtain the desired rate-distortion trade-off. For example, either to generate an image with higher quality but a larger file size or to obtain one with a lower quality but a smaller file size. Though widely used in practice, JPEG has been later succeeded by JPEG2000. JPEG 2000 extensively exploits wavelet-based image coding and the embedded zero-tree wavelet coding techniques. It can compress an image with a low bit rate while preserving fine image details. It supports both lossy and lossless coding and provides progressive transmission. The goal of this project is to explore both the JPEG and JPEG2000 image compression standards.

Specifically, you are asked to

- 1.** Implement a JPEG image encoder and decoder supporting the sequential mode in the JPEG standard. (DCT transform, quantization, differential encoder, zigzag scan, and Huffman coding.)
- 2.** Implement a JPEG2000-like image encoder and decoder. (Wavelet transform, EZW coding, and an entropy coding coder.)
- 3.** Compose a technical report. In the report, you should compare the JPEG and JPEG2000 standards. You should report the performance of your implementations both quantitatively and qualitatively. You should discuss the artifacts in low bit rates. Plot the SNR-BPP curves. Report your discovery. Discuss the pros and cons of various codec designs, for example, in terms of rate distortion, computational complexity, and rate scalability.

For additional credit:

You are encouraged to make other explorations, for example, try out advanced entropy coding and see the impact on the compression ratio by different entropy coding techniques; apply advanced quantization techniques but keep in mind the decoder should have enough information for doing inverse quantization. You can also compare your implementation with the commercial software.

Note:

1. You are free to choose your preferred programming languages. But if you plan to use something other than the MATLAB or the c/c++ languages, please contact your TA. Make sure that your code can be compiled and executed in the TA's computer.
2. You have to provide a readme file explaining your file structure and how to compile them.
3. You have to provide a demo program. The demo program should do the following things. (a) Read an input image (b) Encode the image to a binary file (c) Read the binary file (d) Decode the binary file to an image (e) display both the input and output images.
4. A modular implementation will help you in reusing and organizing your codes.
5. The compression rate should be based on the actual size of the output binary file.
6. The templates of the technical report will be provided in the course website.
Your report should be within 8 pages including all the figures and references. Please carefully plan your space.

Submission

1. Pack all of your codes in a single file, rename it as `enee631_project1_code_yourlastname.zip`, and email it to your TA before **EST 11:59pm 4/7/2011**. Late submission is not accepted.
2. Your report should be in the PDF format and rename as `enee631_project1_report_yourlastname.pdf`. Send an electronic version to your TA before **EST 11:59pm 4/7/2011**. Late submission is not accepted.
3. Example: If your last name is Liu, your code should be renamed as **`enee631_project1_code_liu.zip`** and your report should be renamed as **`enee631_project1_report_liu.pdf`**.