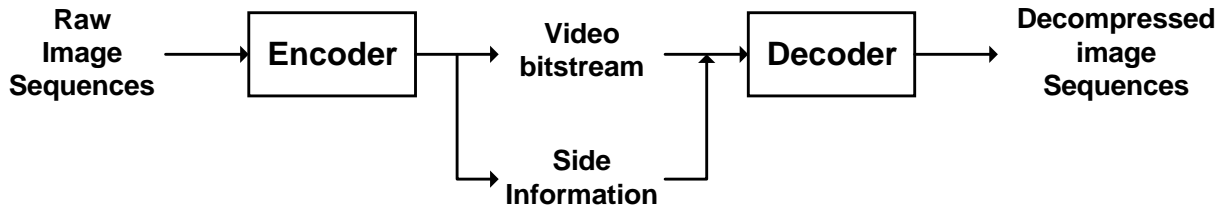


ENEE631 Fall 2001 Course Project

This course project will be done by a team of 2-3 students. The members in the team will receive the same grade, unless unbalanced contribution is noted.

Video Codec Element



- Over the past few weeks, we have completed in C/C++ three building blocks, namely, fast image transform, image codec, and motion estimation/compensation. You are asked to design a video codec based on hybrid coding techniques for grayscale image sequences¹ in the course project.
- Above figure shows the block diagram. The encoder compresses the raw image sequences and output a video sequence of your own format, which doesn't have to be MPEG compliant. Note that you also need to store the necessary side information that a decoder will have to know. For simplicity, the side information can be stored in a separate file and can be losslessly compressed using existing tools like zip/gzip/LZW. The decoder will decode your own format video and decompress it into a sequence of images.
- The frame size is integer multiples of 16x16. Your video codec should use transform coding technique (such as those in your image codec building block) to reduce the spatial redundancy and use motion estimation/compensation to reduce the temporal redundancy. We will provide you a set of default Huffman codebooks to facilitate your integration.² You can also use the two MPEG-1 base quantization tables (one for I-block and one for motion compensation residue block) that were discussed in the lecture as a reference. Your video codec design should give good tradeoff and performance in terms of bit rate, video quality (objective and subjective), and computation time/complexity.

Elective Element

In addition to the functionality of video coding and decoding, you are asked to propose and implement an add-on module. List in the Appendix are some possibilities. This elective element is not limited to them. However, it should not be just a straightforward extension to your video codec element (e.g., it should not just test different fast block-matching algorithms). The elective element is aiming at giving you some experience of literature study, designs, implementation, testing and evaluations.

¹ The image sequence can be obtained from <http://www.ece.umd.edu/class/enee631/am/proj/graden.zip> There are totally 30 raw images. The dimension of each image is 240x352.

² Huffman table and quantization table can be obtained from <http://www.ece.umd.edu/class/enee631/am/proj/proj.htm>

What to Hand-in?

- **Proposal of 1-2 pages ~ electronic submission due 11/13/2001 1:59pm EST.**

Your proposal should (1) describe your video codec structure and systematic plan for design and testing, (2) summarize your literature study and explain your elective element, (3) briefly describe your plan/schedule for doing this project.

- **Demos and oral presentations ~ final week of classes. To be scheduled.**

Each team will be asked to give an oral presentation of your project and answer questions from the audience. The presentation is about 10-15 minutes, covering both the video codec element and the elective element. In the final week of classes, you will also be asked to demonstrate your codec and the elective element, though it may not be the final “product” and you can further improve it.

- **Final written report ~ due in the exam week. To be announced.**

Give brief summarization of background and motivation, describe and justify your designs, evaluate your system with experimental results, and analyze the results you have come up with. The report should be written in the style that is similar to a paper for a technical conference. Please also include a brief description of the contributions made by each team member.

In the mean time, please create a directory of “enee631” with a subdirectory of “project” under your public directory and make them readable (i.e., they should be accessible via <http://your-web-address/enee631/project/>). Put the related video/image results and all of your source codes in this directory by the due time. Include a plain text file with filename “README.txt” to briefly explain what each image/source files included in the directory is. If the files are too large, you can use “gzip” or other lossless techniques to compress the file.

Grading

Project	48% of course grades
[Stage-1] 3 building blocks	28%
[Stage-2]	20%
Proposal	2%
Presentation and peer review	6%
Implementation and results	6%
Final report and overall	6%

Please note that there will be a factor multiplied to your score in Stage-2, depending on the completeness of your project. The factor is added up according to the following table:

Video codec element	0.6
---------------------	-----

Elective element	0.4
Further improvement	up to 0.25

The “further improvement” may include the optimization of various modules in your video codec, additional functionality of the codec (such as effectively handling color video sequence, etc.), and rich elective element.

Regarding Implementation

It is a good idea to plan your implementation in a systematic way. Test each module thoroughly before integrating them into the system. You may also start with a simplified system and gradually improve module by module.

You should write all core modules yourselves. The modules that give further improvement, such as arithmetic coding, can be “borrowed” and modified from the source codes that you may find in the Internet and/or other references. In that case, please indicate the source and acknowledge the authors of those source codes. Please do this assignment using C/C++. See instructor if you prefer other language.

[Appendix] Suggested ideas for the elective element

1. Mesh or object based motion compensation and coding

Ref: class notes, Wang’s book, and Bovik’s Handbook.

2. Motion JPEG via wavelet-based (JPEG 2000) compression

Ref: <http://www.jpeg.org/> and class notes

3. Compressed-domain DC image extraction and shot segmentation

Ref: Yeo-Liu’s paper and class notes

4. Compressed-domain camera motion estimation with application to video content annotation

Ref: <http://www.ee.princeton.edu/~ramadge/recres.html> (IEEE Trans. CSVT’00 paper)

5. Compressed-domain video search by examples

Ref: <http://www.ee.princeton.edu/~ramadge/recres.html> (ICIP’99 paper)

6. Compressed-domain visible watermarking (logo stamping)

Ref: J. Meng, S-F. Chang: "Embedding visible video watermarks in the compressed domain", International Conf. on Image Processing (ICIP), 1998.