PROJECT 3: Video Coding using MPEG-2

Issued: 03/01/2015 Due: 03/29/2015 at 11:59 PM, PST

General Instructions:

1. Assignment descriptions and all files mentioned in this homework are available under "Homework" on DEN Website: http://den.usc.edu.
2. Please follow the submission guidelines for homework, found in the "Course Documents" folder on DEN website.

The homework project includes 3 problems. You will be using the FFMPEG software package to achieve MPEG-2 video encoding and decoding tasks. This software package is widely used in academia and industry. You will benefit tremendously by understanding it thoroughly with reading and practicing. The TA will give you a sequence of lectures on FFMPEG to guide you along this line.

You need to download the following files:

1) FFMPEG reference software:
https://www.ffmpeg.org/download.html
2) H.264/AVC source code package:
http://www.videolan.org/developers/x264.html
3) Testing sequence (YUV 420 format):
http://trace.eas.asu.edu/yuv/akiyo/akiyo_cif.7z
http://trace.eas.asu.edu/yuv/foreman/foreman_cif.7z
4) YUV Player (you can choose any one or search for other similar software):
http://www.yuvplayer.com/
http://dsplab.diei.unipg.it/~baruffa/dvbt/binaries/player/

Notes:

- FFMPEG is written on the Linux platform in C language. Please download the source codes from the website first. For Linux/Mac OS users, makefiles are provided and it is easy to compile and build. For Windows users, the installation is more complex and it is better to use the same development environment, such as MinGW.
- Before the installation of the FFMPEG software, please download the H.264/AVC source code package and build it into your system first. Then, add it into the configuration setup to compile FFMPEG and H.264/AVC together. This part is needed in Homework#4.
- Read the **FFMPEG installation guidelines.pdf** and the material in its official website (https://www.ffmpeg.org/platform.html).

Problem 1: Written questions (18%)

Answer the following questions:

A. Review two lossy compression codecs – MPEG1 and MPEG2, and describe the major differences between them. (6%)
*Hint: Take the **MPEG2 encoder** as an example, read and understand the FFMPEG source code package, including the data structure and the implementation flowchart.*

B. The size of the input stream for encoding can be very large. It requires a lot of memory to store the file. However, FFMPEG uses another strategy to manage the memory. Please explain how FFMPEG deal with this issue. (3%)
Hint: Check the main function in ffmpeg.c. Find out the part (i.e. the function name) to which the strategy is applied.

C. Start from the main function in ffmpeg.c, draw the function flow of converting the input video stream using the MPEG2 codec. More specifically, find out the key functions called during the process step by step. Explain the function input/output parameter. You can skip some functions like initialization and memory release. (6%)

D. FFMPEG actually runs very fast (compared with other software) to convert the video. We can see the source code is optimized for both memory management and fast calculation. List 2~3 points on performance optimization adopted by the FFMPEG source code. (3%)

Problem 2: Encoder Configuration and Motion Estimation (52%)

After compiling the FFMPEG software, you can use the command line to call different codecs for compression. The standard statement is:

ffmpeg *[global_options] {[input_file_options] -i input_file} ... {[output_file_options] output_file} ...*

For example, to set the video bitrate of the output file to 3000 kbit/s:

ffmpeg *-i input.avi -b:v 3000k -minrate 3000k output.avi*

A. Please check the command line parameters in the FFMPEG official website. Set the following parameters: (6%)
- Set compression codecs to MPEG2
- Set motion estimation method to EPZS (Enhanced Predictive Zonal Search)
- Set thread detection to 2
- Set frame rate to 25
- Set GOP size to 12
- Set the output bitrate to 250kbits/s

Encode Foreman.cif and Akiyo.cif using FFMPEG. Report the resulting PSNR, bit rates and total encoding time. Decode your MPEG2 bit-stream. Play the original and resulting videos, and make some brief comments.

B. The following experiments should be conducted based on the previous setting.
   (1) No B-frames are used in the default setting of FFMPEG, which means the coding structure is IPPP…. You are asked to insert one, two and three B-frames between adjacent I/P or P/P-frames, respectively. Compare your new results with that in Part A with discussion.
   (2) The GOP size will affect the video quality. Please try three GOP sizes (8, 15 and 30) and compare their results. How to decide a proper GOP size for a specific video input?
   (3) Repeat the above sub-problem but with three different bitrates (300kbps, 200kbps, 100kbps) while keeping the same GOP size (say, 15).

Encode two video sequences and report the resulting PSNR, bit rates and total encoding time with discussion on the performance differences. Does your subjective perception match with the PSNR comparison? Explain why or why not. (18%)

C. After understanding the encoding process in the MPEG-2 codec, you should be able to find where *motion estimation* (ME) takes place. Do the followings:
   (1) Reset the parameters the same as in Problem 2 Part A. Add your codes to extract the motion vector information for each 16x16 marcoblock and output the data to a file. (15%)
   (2) Collect the motion vectors and produce 3D histogram plots (horizontal offset, vertical offset, and number of vectors) for motion vectors for the two bit streams with three different bitrates (300kbps, 200kbps, 100kbps). Discuss the results. (6%)
   (3) The implemented motion estimation method is EPZS [1]. Please describe the idea of this search method (any combination of texts/figures with maximum up to 1 page). (7%)

Problem 3: Rate control (30%)

Rate control is an important topic in video compression. This process is controlled by quantization parameter (QP). In MPEG2, QP is ranged from 2~31. The smaller the QP, the higher the video quality. When encoding a video clip, it sometimes demands a constant bitrate output transfer, which is called the constant-bit-rate (CBR) video. However, since bits allocated to different frame types (I, P, B) are different, the buffer of the encoder is fluctuating. In this problem, you are asked to design a simple rate control method based on the buffer information.

A. Set the parameters of the encoder as follows:
   - Frame rate: 25
   - GOP size: 10
   - Output bitrate: 250kbits/s
   - Buffer size: 128k.

   Find the buffer value during the encoding process. Plot the buffer fullness figure and the PSNR value along the time (frame by frame). (10%)

B. In the previous testing, we can see QP is changing along the time by fixing the bitrate. It is controlled by the target bit allocation. In this part, we design a stable rate control method based on the buffer value. (20%)

The buffer size is fixed at 128k in Part A. We want the buffer to maintain a certain level in the encoding process, say T = 64k bits. Since the buffer output speed is constant, the fluctuation arises due to the variation of the input speed – encoded bits. For example, if the buffer is larger than T, we need decrease the input bit speed, which means to increase the QP value. Please study paper [2] about buffer control. In this paper, it dynamically adjusts the quantization parameter for each frame by considering coded frame quality, allocated transmission bandwidth, buffer occupancy and scene changes. In the proposed strategy, the QP is divided into the basic quantization scale (BQS), which is used to maintain the constant level of visual quality, and the adjustable quantization scale (AQS), which is used to fit the transmission rate and avoid overflow and underflow conditions. Then, do the following three tasks.

1) Please summarize the rate control scheme (with words and equations) in [2] based on your understanding (maximum - up to 1 page).
2) Plot the buffer fullness figure after buffer control (frame by frame). Compare the result to the case without control. Discuss the performance. Also, plot the PSNR value of each frame for both cases.
3) Discuss the robustness of the above rate control strategy by varying the output bitrates (e.g. from a large value to a small value, or vice versa). When is underflow or overflow likely to occur?

References:

[1] Tourapis, Alexis M. "Enhanced predictive zonal search for single and multiple frame motion estimation." In Electronic Imaging 2002, pp. 1069-1079. International Society for Optics and Photonics, 2002.

[2] Chang, Chih-Feng, and Jia-Shung Wang. "A stable buffer control strategy for MPEG coding." Circuits and Systems for Video Technology, IEEE Transactions on 7.6 (1997): 920-924.