PROJECT 4: Video Coding using H.264

Issued: 04/05/2015 Due: 04/26/2015 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, which can be found in the "Course Documents" folder on DEN website.

The homework project includes 3 problems. You will use the FFMPEG software package for H.264 video coding and related tasks. This software package is widely used in academia and industry. You will benefit tremendously by understanding it thoroughly with reading and practicing.

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):
Foreman: http://trace.eas.asu.edu/yuv/foreman/foreman_cif.7z
Elephants Dream: https://archive.org/download/ElephantsDream/ED-360-png.tar
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/
5) H264 player (VLC): http://www.videolan.org/vlc/

Notes:

- For H.264 video coding, FFMPEG serves as a shell platform, dealing with demuxing and muxing multimedia data (video & audio). It calls the x264 open source library to finish the encoding task. In this homework, you are not required to make any modification in FFMPEG. Instead, you should focus on the x264 library. You need understand and modify some parts in it.
- It is optional for you to use FFMPEG as the I/O routine. The parameter setting reference for FFMPEG H.264 encoding is given in the following website:
https://sites.google.com/site/linuxencoding/x264-ffmpeg-mapping
You can also directly compile and run x264 to get the h264 compressed file. For options in the x264 library, please check this link:
http://en.wikibooks.org/wiki/MeGUI/x264_Settings

Problem 1: Written questions (10%)

Compare H.264 with MPEG2, and describe the new techniques adopted in H.264 that improve the rate-distortion performance. (Limit your write-up to 3 pages.)

H.264/AVC Software Package

The ITU-T Video Coding Experts Group and the ISO/IEC Moving Pictures Experts Group formed the Joint Video Team (JVT) and completed the standardization of H.264/MPEG-4 AVC in 2003. Like previous standards, H.264 specifies only the decoder and allows coding gain and speed improvement in encoder design. The encoder developed by the JVT, known as the Joint Model (JM), offers a reference software tool. Its usage is somehow limited due to its slow speed. x264 is another H.264 source code package open to the public. It is used in many popular applications like ffdshow, ffmpeg and MEncoder. In recent study, x264 offer better quality than several commercial H.264 encoders. The high performance of x264 is attributed to its rate control, motion estimation, macroblock mode decision, quantization and frame type decision algorithms [3]. The following two problems aim to help you understand two modules implemented in x264.

Problem 2: Motion estimation in x264 (50%)

Motion estimation (ME) is the most complex and time consuming part of the H.264 encoder as it uses multiple prediction modes and reference frames.

1. There are four different integer-pixel motion estimation methods provided by x264: diamond (DIA), hexagon (HEX), uneven multihexagon (UMH) [1] and successive elimination exhaustive search (ESA) [2]. Please describe DIA, HEX, UMH and ESA (with words and equations) based on your understanding. (20%)
2. x264 implements a modified initialization scheme for all motion search methods so as to improve PSNR. Also, an early termination and range adaptive (ETRA) algorithm for UMH improves the speed. For more details, please read and understand paper [3]. Explain the motion search strategy implemented in x264. (10%)
3. Point out where extra MV's initialization and the ETRA usage in the x264 source code. Compare the performance of motion search methods in x264 with and without optimizations. Report the performance of encoding the **Foreman** sequence using the following motion search schemes: (20%)

DIA_x264	DIA_no_emv
HEX_x264	HEX_no_emv
UMH_x264	UMH_no_ETRA
UMH_no_emv	UMH_no_both
ESA_x264	ESA_no_emv

Table 1: Search methods

In Table 1, "_no_emv" means only four neighbor motion vectors used (without the use of listed extra MV candidates for initialization [3]). The "ETRA" algorithm is only applies to the UMH searching scheme in UMH_x264..

Report the PSNR of each ME approach in Table 1. Also, measure the time of motion estimation of each case for the **Foreman** sequence. Compare and discuss your results.

Note: you can set the search method by command setting. However, if you want to disable extra MV’s initialization and ETRA, you need to modify the code on your own. Also, the time to report is the motion search time, not the total encoding time.

Problem 3: Rate control in x264 (40%)

Rate control allows selection of encoding parameters to maximize quality under the bit-rate constraint and the decoder buffer constraint. Rate control in H.264 can be performed at three different granularities – 1) the GOP (group of pictures) level, 2) the picture level, and 3) the macroblocks level. At each level, the rate control algorithm selects the quantization parameter (QP) values that determine the quantization of transformed coefficients. As the QP increases, the quantization step size increases and the bitrate decreases. There are five rate control modes in x264: one two-pass mode and four one-pass modes. The QP value can vary from one macroblock to another in the constant bitrate mode while the QP value can only vary from one frame to another in other modes [3]

1. Describe the following five different modes in x264 [3]. (20%) (1 page for each method)

Video buffer verifier compliant constant bitrate (CBR)
Average Bitrate (ABR)
Constant rate-factor mode (CRF)
Constant quantizer mode (CQP)
Two pass (2pass)

2. Test each rate control schemes with the **Elephants Dream** sequence at a target bitrate of 800 kb/s. Please report the final bitrate of each encoded video and the PSNR. Discuss the performance of each scheme. (20%)

Note: sequence is 640x480 pixels resolution, 25 frames per second and total 15691 frames. For CQP mode, please try different QPs (0~51) and compare their results.

References:

[1] X. Yi, J. Zhang, and N. Ling, “Improved and simplified fast motion estimation for JM,” JVT-P021, July 2005.

[2] X. Gao, C. J. Duanmu, and C. R. Zou,“A multilevel successive elimination algorithm for block matching motion estimation,” IEEE Trans. on Image Proc., vol. 9, no. 3, pp. 501–504, 2000.

[3] Merritt L, Vanam R. Improved rate control and motion estimation for H. 264 encoder[C]//Image Processing, 2007. ICIP 2007. IEEE International Conference on. IEEE, 2007, 5: V-309-V-312.

[4] Jing, X., and L-P. Chau. "Fast approach for H. 264 inter mode decision." Electronics letters 40.17 (2004): 1050-1052.

[5] Kim, Byung-Gyu. "Fast selective intra-mode search algorithm based on adaptive thresholding scheme for H. 264/AVC encoding." Circuits and Systems for Video Technology, IEEE Transactions on 18.1 (2008): 127-133.