

Assignment 3 – Playing with Open-Source H.265/HEVC Encoders

Highlights

- This is an *open-ended* project; which means you get to pick your own goals. Like open-book exams, open-ended projects are usually tougher. Please allocate ample amount of time and start early.
- This is a *group* project. You may team up with two other students (please fill in the form <http://goo.gl/HO17I5>). However, in the project report, please estimate the contributions of individuals both qualitatively (who did what) and quantitatively (percentage of contribution).
- *Deadline*: 5/31 (Wednesday), 23:59:59.

Introduction:

In this assignment, you are going to modify an H.265/HEVC encoder. High Efficiency Video Coding (HEVC), also known as H.265, is a new video compression standard. H.265/HEVC include many more coding tools, which are similar to those in H.264/MPEG-4 AVC, but more powerful. H.265/HEVC was developed with the goal of providing twice the compression efficiency of the previous standard, H.264/MPEG-4 AVC. One of the improvement is **motion prediction**. Blocks of pixels are encoded by making reference to another area in the same frame (intra-prediction), or in another frame (inter-prediction). Where H.264/AVC defines macroblocks up to 16×16 pixels, H.265/HEVC can describe a much larger range of macroblock sizes, up to 64 x 64 pixels. However, compression efficiency results vary depending on the type of content and the encoder settings (which is TRUE for ALL codecs). We want you to get your hands dirty and play with (i.e., modify and improve) open-source H.265/HEVC encoders.

Open-Source H.265/HEVC encoders:

In theory, you can choose any open-source encoder, but we recommend the following three. If you want to work on other codecs, please discuss with the instructor before making the decision.

1. x265 (<http://x265.org/>). x265 is an H.265 / HEVC video encoder library and application. It is designed to encode video or images into an H.265 / HEVC encoded bitstream.
2. Turing Codec (<http://turingcodec.org/>). Turing codec is an H.265/HEVC software video encoder and decoder for efficient video compression.
3. Kvazaar (<http://ultravideo.cs.tut.fi/#encoder>). Kvazaar is yet another open-source H.265/HEVC codec. It was awarded as the winner of the Open Source Software Competition in ACM Multimedia Conference 2016.

Goals:

You get to choose what you want to enhance or implement, but justify in your final report why your project is important.

1. Rate control. A rate control algorithm dynamically adjusts encoder parameters to achieve a target bitrate. It allocates a budget of bits to each group of pictures, individual picture and/or sub-picture in a video sequence. One thought would be recycling the ICME'17 Twitch grand challenge problem, but using an H.265/HEVC codec. ICME'17 has a detailed document describing the goal of “DASH-friendly rate control”, please see <http://www.icme2017.org/wp-content/uploads/2017/01/TwitchProposalforICME2017GrandChallenge.pdf>.
2. Context-aware video coding for games. Context-aware video coding for games refers to the idea that encoder can encode the video based on the game context. For example, a racing game that we know most of the pixel will move backward in the next frame. Then our encoder can save time from the *motion estimation*. The tricky part of this project is to design an efficient and flexible *API* for game developers to tap in your enhanced H.265/HEVC codec. You will need to carefully design the API and implement it. You probably want to use an open-source game as an example of how to “use” your API. Sample (and simple) open-source games are: Bzflag (<https://www.bzflag.org/>) and Alien Arena (<http://red.planetarena.org/>).

3. Any other enhancement you are interested in. Please discuss with the instructor first.

What to Turn In:

- Source code of your enhancement. Please provide the original zip (or tar or other compressed) file of the original H.265/HEVC codec. Please also provide the modified zip file of your own H.265/HEVC codec.
- Project report, which is a document including the following sections:
 1. Motivation, including Limitations of the Chosen Codec
 2. Code Structure of the Chosen Codec
 3. Design of Your Enhancement
 4. Implementation of Your Enhancement
 5. Instructions on Compiling and Running Your Codec
 6. Evaluations that Quantitatively Compare Original and Your Codecs and Show the Improvement
 7. Lessons Learned
- TA will provide an ftp site and instructions on how to upload your code and report (in PDF, please)
- This assignment will be marked by the instructor, not the TA.