NTNU
Norwegian University of Science And Technology
Faculty of Information Technology, Mathematics and Electrical Engineering
Department of Electronics and Telecommunications

TTT4135 MULTIMEDIA SIGNAL PROCESSING
Assignment 2

Note

The assignment consists of two parts. First part deals with theoretical problems while the second one deals with practical issues related to H.264 encoding. Both parts should be done in groups of two students.
Please write the solutions in English or Norwegian, and submit it electronically on It's Learning in "Mandatory Assignements -¿ Assignment1". DEADLINE FOR HANDING IN THE SOLUTIONS IS March 14, 2021. Solutions submitted after the deadline WILL NOT be accepted.

Preliminaries

The following materials should be studied before starting this assignment. The related papers can be found in the folder 'References' on its learning.

- Chapters 11.1, 11.2 and 11.3 of the textbook (Digital compression for multimedia) for details of MPEG video coding standard
- Video Compression From Concepts to the H.264/AVC Standard
- Overview of the H.264/AVC Video Coding Standard
- New features and applications of the H-264 video coding standard

Part I

1 Temporal prediction

The inherent temporal correlation that exists in video data can be utilized for compression purposes.

a) Explain the principles of *motion compensation* and how this generally is done.

b) What do we mean by I, P and B-frames, and what types of motion compensation is done in each of these? What does this imply for the compression gains and visual quality for each of these frame types?

c) Why is there a difference between the *transmit order* and the *display order* of frames?

d) Give a short summary (no details) of how H.264 separates itself from other standards with respect to how motion compensation is done.

2 Decomposition/Transform

As is usual in still image coders, video coders do a frequency decomposition prior to quantization and entropy coding. It is the residual image (what remains after prediction) that is transformed[1].

a) Which transform has usually been used in classic video coders (such as MPEG-1)? What is used in H.264?

b) Block-transform-based video/still image coders usually exhibit so-called blocking artifacts. How is this effect minimized in H.264?

Part II

3 H.264

a) In this part of the exercise, one is to use a command prompt (DOS) H.264 coder to exemplify the current topic. Be warned of the fact that the encoder is computationally rather complex. This implies that each compression job will consume some time. The following files can be found in the folder 'Program' on its learning, along with all the other needed material.

- `lencod.exe`
  This is the encoder. It is run without trailing parameters from the command prompt.
- `encoder.cfg`
  Configuration file for the encoder. This must be opened in a text editor such as notepad or textpad when changes are to be made. All parameter settings that are done in this exercise are set inside this file.

[1] If no good prediction can be found, most video coders (including H.264) will use intra coding for the macro block in question.

- `ldecod.exe`
  The decoder. This is actually not used in this exercise, as the *encoder outputs a decoded sequence* in addition to the compressed file. It can be run, but needs the name of the decoder configuration file as an argument - like this: `ldecod decoder.cfg`
- `decoder.cfg`
  The mentioned configuration file for the decoder. No need to change settings here.
- `MOBILE_CIF_ORIGINAL.yuv`
  Raw video to be encoded.
- `vidview`
  Application used to play raw video (YUV-format). This is not a command prompt program. It runs as an ordinary Windows application. It can open both the original and the reconstructed sequence `MOBILE_CIF_RECONSTRUCTED.yuv` simultaneously, since this allows for synchronous comparison. This can be done using the "Dual open" function. Remember to check that the Sequence $->$ Default size option is set to CIF (352x288). Click Sequence $->$ Frame rate (or f/s) to adjust the frame speed when viewing for comparision.

b) Encode the raw video, and adjust the distance[2] between I-frames ("IntraPeriod"). Try f.ex. 5, 15 and 30. Use a relatively low bitrate (f.ex. 160kbps). What can be observed visually? Does the quality progression throughout frames correspond to the progression of SNR-values as reported by the encoder/decoder?

Note: The rate control module in the program does not work well for all bitrates. Therefore, cross check that the rate that you set in the config file is approximately equal to the actually produced bitrate. This can be seen from the end of the encoder output. When the produced bitrate is different from the intended bitrate, it is useful to adjust the quantization parameter for the first frame in the first GOP ("InitialQP"), which is found at the end of the configuration file. This is because the quantization

[2]This 'distance' does not count B-frames. With every second frame being a B-frame, the I-frame period will actually be twice this parameter value.

of this specific frame is not handled by the rate control module, but needs to be done manually.

Note: Code **many** GOPs (> 5) when doing this. This may take some time, but it is necessary to get meaningful results. The number of frames to encode is controlled by the parameter "LastFrameNumber", which is found as the first parameter in the "Additional stuff" category in the config file.

State the average SNR-values and the average bitrate[3] that was obtained for the different cases.

c) Fix the distance between I-frames to 30 (that is, set IntraPeriod = 15), but adjust the bitrate (can be set in the paragraph "Rate control" in `encoder.cfg`). At which bitrate (approximately) is it not possible to notice any difference between the original and the reconstructed video? It is recommended that both sequences are studied synchronously in the YUVviewer to determine this.

d) Turn off the rate control (set RateControlEnable = 0) and vary the encoding procedure through adjusting the *quantization parameters*. Limit this test to concern itself with adjustment of the following parameters:

- QPFirstFrame
- QPRemainingFrame
- QPBPicture

These parameteres can have values between zero and 51. Their significance is that a lower number means finer quantization and therefore higher rate and better quality. Explain the effect that is observed through adjustment of these (visually, SNR and bitrate). Note: The normally used five levels of visual effect of distortions are 'Imperceptible','Perceptible but not annoying', 'Slight annoying','Annoying' and 'Very annoying'.

[3]Can be read from the end of the command prompt output of the the encoder.