Information and Coding (2021/22)

Project #02

Entropy coding of audio and images

- All the programs should be implemented in C++.
- Document all the code using the Doxygen tool.
- Create a Github project to manage the several versions of your software.
- Each program should be demonstrated during the classes

Part A

1. Implement a class BitStream to read/write bits from/to a file. This class should have, at least, methods to write one bit, read one bit, write n bits and read n bits. The resulting file should be binary (not text) and take into consideration that the minimum amount of data that you can access in a file is one byte (8 bits). You can implement other methods that you think might be necessary (for example, methods to read and write strings, in binary). This class should be optimized, due to its extensive usage during compression / decompression.

2. Implement a simple program to test the Bitstream class.

3. Implement an entropy encoder using Golomb codes. Start by developing a class `Golomb`, where you should implement, at least, one method to encode numbers (signed integers) and another one to decode them. It should be possible to specify the parameter `m` of the Golomb code.

4. Implement a simple program to test the Golomb class.

Part B

Using the Golomb coding algorithm, you have to implement an audio codec. The codec should rely on predictive coding.

1. Develop a lossless predictive audio codec, exploring temporal and channel redundancy, followed by Golomb encoding. You should try, at least, a simple prediction based on the causal neighbors of the sample being encoded. Remember that the ultimate goal is to attain the largest compression ratio as possible, although compression/decompression time should be reasonable.

2. Include an option into your encoder to calculate the histograms of the residuals obtained after prediction, as well as the entropy value. Compare these values with the entropy of the original data.

3. Include an option for lossy coding in the developed codec, based on residual quantization. You can take into consideration some software implemented during the Project #01. The ultimate goal is to attain the largest compression ratio as possible with the minimum error introduced.

Part C

Using the Golomb coding algorithm, you have to implement an image codec. The codec should rely on predictive coding. Consider a pre-processing stage in order to transform the image into the YUV 4:2:0 format before the encoding process. The codec is divided into two stages. You should consider each one as a different version of the codec.

1. Develop a lossless encoder that complies to the following requirements:
    a. The frames should be encoded using spatial predictive coding based on the non-linear predictor of JPEG-LS or the 7 JPEG linear predictors;
    b. Entropy coding should be performed using Golomb codes;
    c. All the information required by the decoder should be included in the bit-stream (video format, frame size, encoder parameters, etc.).

2. Based on the lossless codec developed in the previous stage, in this stage you should extend it in order to allow lossy coding. The encoder should receive three additional input parameters, indicating the quantization steps used for quantizing the prediction residuals of the three color components. The quantized values will be entropy coded using Golomb codes.

   As a bonus, you can implement another lossy version of the codec, based on transform coding of the prediction residuals, using the DCT as in the JPEG standard, and quantization of the coefficients. The quantized values have to be entropy encoded using Golomb codes or another coding method.

Part D

Elaborate a report, where you describe all the steps and decisions taken in all the items of the work. If appropriate, include measures of processing time, compression ratios and SNR (for the lossy case).

You should include the Doxygen documentation (if available) and a link to your repository.

Take as reference the images and audio files available on e-learning.