UNIVERSIDADE DE AVEIRO
DEPARTAMENTO DE ELECTRÓNICA, TELECOMUNICAÇÕES E INFORMÁTICA

Information and Coding (2025/26)

Lab work nº 2 — Due: 16 Nov 2025

Intro

- In this work, besides dealing with audio files, you will also need to manipulate image files.
- This will be done using the OpenCV library (`https://opencv.org/`). Therefore, start by installing this library and take a look at `https://docs.opencv.org/4.x/de/d7a/tutorial_table_of_content_core.html`. **Note:** before trying to download and build the OpenCV library from scratch, find out if there are prebuilt packages for your system. As usual, you will need both the runtime and development libraries.

Part I

1. Using the OpenCV library, implement a program to extract a color channel from an image, reading and writing pixel by pixel, creating a single channel image with the result. The file names and channel number should be passed as command line arguments to the program. You may find some examples of images in moodle.
2. Without using possible existing functions of OpenCV, implement a program that:
   (a) Creates the negative version of an image
   (b) Creates a mirrored version of an image: (a) horizontally; (b) vertically
   (c) Rotates an image by a multiple of 90º
   (d) Increases (more light) / decreases (less light) the intensity values of an image

Part II

3. Implement a C++ class for Golomb coding. This class should have (at least) functions for encoding an integer (generating the corresponding sequence of bits) and for decoding an appropriate sequence of bits into an integer. The class will have a parameter, m, that will allow fitting the codes to the appropriate probability distribution. It should also be able to handle negative numbers, using two different approaches (chosen by the user): (1) sign and magnitude; (2) positive/negative value interleaving.

Part III

4. Implement a lossless audio codec, based on Golomb coding of the prediction residuals. The codec should be able to handle both mono and stereo audio files. Consider both temporal and, in the case of stereo audio, inter-channel prediction. Remember that the efficiency of Golomb coding depends on a parameter, m, which can be fixed and indicated directly to the encoder or, preferably, be adaptive and optimally determined during encoding/decoding.

Part IV

5. Implement a lossless image codec for grayscale images, based on Golomb coding of the prediction residuals. This codec should provide the best possible compression, using appropriate predictors and values of m for the Golomb codes.

Part V

6. Elaborate a concise report, where you describe all the relevant steps and decisions taken in all the items of the work. When appropriate, include also measures of processing time and compression ratios. For this, use several audio and image examples. This report should not be a description of the code implemented. Instead, it should illustrate what can be obtained using the software developed and how it can be obtained. Also, it is important to compare your results with those obtained with existing lossless audio and image codecs.