IC Project 3

December 2023

Contents

1 Introduction

This project was around the compression of images and videos using Golomb's algorithm and predictive coding. The final goal was to achieve a software that would efficiently compress and decompress files without losses.

2 Part 1

2.1 Compressor 1 frame

The first program implemented in file **ex_1.cpp** provided the functionality to compress an image. It was achieved using the encoding as bits of the residuals of the pixels in the image. For each pixel, the predicted pixel was calculated as (a+b)/2, with *a* being the pixel to the left and b the pixel directly on top.
After calculating the predicted pixels, the residual for each pixel was calculated as $residual = actual - predicted$. Then, the residual matrix formed was compressed using Golomb's encoding.
In order to build and test it in path **IC-42078/src**:

```
$ mkdir build
$ cd build
$ cmake ..
$ make
$ ./ex_1 <path to input image>
```

Then, in the project's root folder, the *ex_1_data.comp* file will be the compressed file and *ex_1_data_decompressed.comp* will be the original file generated from the compressed file.

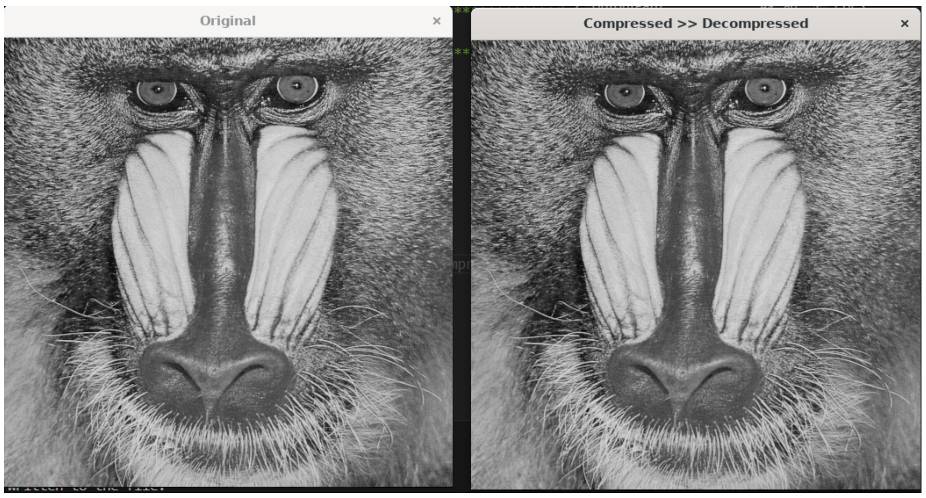

Figure 1: Original image at the right and single channel (Channel 0) image at the left

The image above, as an example, was compressed from 769kb to 702kb, a reduction of almost 9% in the size.

2.2 Compressor video

The video compressor used as skeleton the 1 frame compressor, extending the logic for each frame in the video. Altough the compression could be completed, it could not reduce the file size. The decompression could not be achieved because of video codec format problems.
In order to test the code for video compression (although it fails):

```
$ mkdir build
$ cd build
$ cmake ..
$ make
$ ./ex_2 <path to video>
```