universidade de aveiro deti departamento de electrónica, telecomunicações e informá

# Complementos Sobre Linguagens de Programação (2023/24)

## Project: Image and Video compression

- This project 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 and to collaborate during the development

### Deliverable 1

1. Implement a class to manipulate digital images. Test this class implementing a program to copy an image to another file, pixel by pixel. Consider image files in PPM format (https://netpbm.sourceforge.net/doc/ppm.html).
2. Using the OpenCV library, implement a video player that could be able to display video in RGB or YUV formats. Include an option in your player to perform the conversion between RGB and YUV color spaces. Take into consideration the planar mode in the three possible subsampling modes of YUV: 4:4:4, 4:2:2: and 4:2:0.
3. Include an option in your player to perform the next low level image processing operations. You should implement the algorithms by yourself. Take into consideration that you do not have to save the processed videos, but only display the processed images.
    a. Include a watermark into an image (it can be a string identifying the group or a chosen picture).

b. Transform an image into other color spaces, namely between RGB and YUV
c. Calculate and display the color histograms of an image.
d. Convert a color image/frame to grayscale
e. Apply histogram equalization.
f. Apply Gaussian and blur filters to an image, exploring different filter kernels.
g. Explore the use of segmentation algorithms based on threshold (as bonus, explore more advanced algorithms - e.g. watershed, region growing, etc).

**Deliverable 2**

4. Implement a class `BitStream`, to read/write bits from/to a file, as part of Golomb to read/write data in the encoded file. Recall that 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.
5. Implement a simple program to test the BitStream class. Unitary tests should be considered as well.

**Deliverable 3**

6. 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.

7. Implement a simple program to test the Golomb class. Unitary tests should be considered as well.

## Project Deliverable Deadline

- The deadline for Deliverables 1, 2, and 3 is 22nd October 2023, 23:59.

## Submission Instructions

- Please submit your deliverables by pushing your code to the Git repository. A git pull will be done at the deadline and the evaluation will be based on this submission.

## Documentation

- Please ensure that your deliverables are well-documented using Doxygen.

## Demonstration and Questions

- Please be prepared to give a demonstration of your project and answer questions in the next class.

### Deliverable 4

Using the Golomb coding algorithm, you have to implement a video codec for video sequences previously saved in files. The codec should rely on block based motion compensation and predictive coding.

8. Develop a lossless intra-frame 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 seven 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.).

**Deliverable 5**

9. Develop a lossless hybrid encoder (intra + inter coding), complying to the following requirements:
    a. The block size and the search area for inter-frame coding should be an input parameter of the encoder;
    b. The periodicity of the key (intra) frames should be an input parameter of the encoder. For encoding these frames, use the method developed in the first stage;
    c. As a bonus, you can develop an algorithm to estimate, in real-time, if the current frame should be encoded in intra or inter mode;
    d. All the information required by the decoder should be included in the bit-stream (video format, frame size, block size, search area, code parameters, etc.);
    e. Entropy coding should be performed using Golomb codes.

**Project Deliverable Deadline**

- The deadline for Deliverables 4 and 5 is 26th November 2023, 23:59.

**Submission Instructions**

- Please submit your deliverables by pushing your code to the Git repository. A git pull will be done at the deadline and the evaluation will be based on this submission.

**Documentation**

- Please ensure that your deliverables are well-documented using Doxygen.

## Demonstration and Questions

- Please be prepared to give a demonstration of your project and answer questions in the next class.

### Deliverable 6

10. Based on the lossless video codec developed in previous stages, 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 other lossy versions 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.

### Extra Mile

11. Explore the use of the `perf`[1] tool to profile your code and identify the most time-consuming operations, allowing you to focus optimization efforts on the critical sections of the program.

### Final presentation / Report (12th January)

12. 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).
    The final assessment will be based on the best results of compression ratio, processing time and error introduced (for the lossy version) taking as

reference the images and videos available on the Padlet.

---

[1] https://perf.wiki.kernel.org/index.php/Tutorial