universidade de aveiro deti departamento de electrónica, telecomunicações e informática

Complementos Sobre Linguagens de Programação (2020/21)

Project: Image Processing 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

Deliverable 1 - 28 oct. 2020

1. Using the OpenCV library, implement a video player that could be able to display video in RGB.
2. Include an option in your player to perform the next low level image processing operations. 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 YUV and HSV and visualize individually each color channel as grayscale image.
    c. Calculate and display the color histograms of an image.
    d. Convert a color image to grayscale and apply histogram equalization.
    e. Apply gaussian and blur filters to an image, exploring different filter kernels.
    f. Explore the use of segmentation algorithms (using threshold or more advances algorithms - e.g. watershed, region growing, etc).
    g. Explore the use of the most common morphological operators in the segmented images.
    h. Find image gradients (Sobel, Scharr and Laplacian derivatives).
    i. Apply the Canny edge detection algorithm, exploring its parameters. Explore how to manipulate the corresponding contours.

3. 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.

Deliverable 2 - 4 nov. 2020

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 - 11 nov. 2020

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.

Deliverable 4 - 25 nov. 2020

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. The project is divided into three stages. The first one is a video player. For the other three, you should consider each one as a different version of the codec.

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 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.).

Deliverable 5 - 16 dez. 2020

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.

Deliverable 6 - 13 jan 2021

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 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.

Extra Mile - 13 jan 2021

11. Implement a video player using the Qt framework.

Final presentation / Report - 26 jan 2021

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 mark will be calculated based on the best results of compression ratio, processing time and error introduced (for the lossy version) taking as reference the following videos available on https://media.xiph.org/video/derf/:

    a. ducks_take_off
    b. in_to_tree
    c. old_town_cross
    d. park_joy