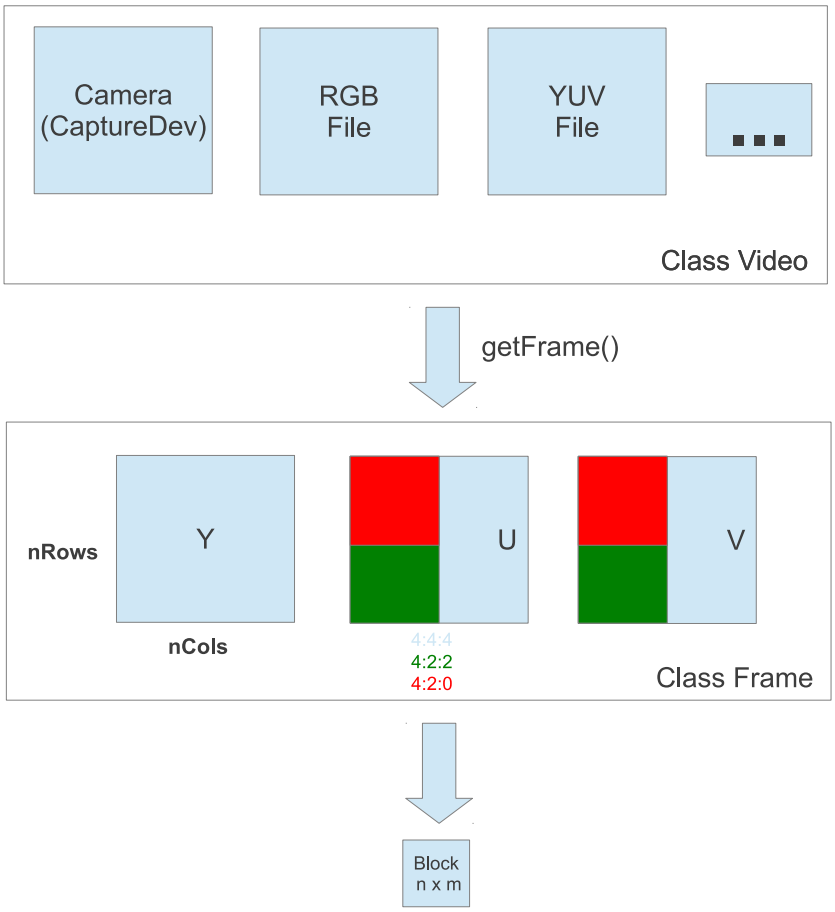
UNIVERSIDADE DE AVEIRO
DEPARTAMENTO DE ELECTRÓNICA, TELECOMUNICAÇÕES E INFORMÁTICA

Audio and Video Coding (2013/2014)

Lab work n$^{\circ}$ 1 — Due: 24 Out. 2013

1. Develop a set of `C++ Classes` to manipulate sequences in the following color spaces: RGB, YUV/YC_bC_r 4:2:0, YUV/YC_bC_r 4:2:2 and YUV/YC_bC_r 4:4:4.

   At least you should implement three classes. One denoted `Video` that should provide several methods, namely load files, save files, access to cameras, convert formats, among others. The second class, denoted `Frame`, should use an efficient data structure to store the video data. Internally, the video should be stored in YUV planar mode, being possible to manipulate the three subsampling modes: YUV 4:2:0, YUV 4:2:2 and YUV 4:4:4. It should contain several methods, namely, display frames, get and put pixel values, get and put blocks, and some others according to your needs. The third one, `Block`, should be used to manipulate blocks of image data.



2. Adapt the program (`yuvShow`) to use the developed module in order to be able of displaying videos in the three YUV/YC_bC_r formats and in RGB format, both from files or from a digital camera.

3. Implement a program to copy a video, block by block, trying it with different block sizes. The main goal is to test the `Block` class developed before.

4. Develop a `videoCmp` program, that compares two sequences in terms of PSNR and root mean squared error. The peak signal to noise ratio is given by:

$$\text{PSNR} = 10 \log \frac{A^2}{e^2}$$

where A is the maximum value of the signal, e^2 is the mean squared error between the reconstructed image, $\tilde{f}$, and the original image, f,

$$e^2 = \frac{1}{N_R N_C} \sum_{i=1}^{N_R} \sum_{j=1}^{N_C} [f(i,j) - \tilde{f}(i,j)]^2$$

5. Write a program, named `videoResize`, that can perform spatial resolution reduction, as well as spatial resolution expansion. For example, a reduction of 1:2 transforms a video with frames of $N \times M$ pixels into another with $N/2 \times M/2$) and an expansion of 1:2 transforms a $N \times M$ video into one of resolution $2N \times 2M$ pixels). Use the program `videoComp` to determine how much distortion was introduced by the two operations. Do not forget to compare the several solutions implemented and to comment the results.

   Note: There are several possibilities for performing these operations. Implement at least two for each case.

6. Implement a program to perform some simple transformations in the video, denoted `videoEffects`. It should allow the following operations:

   - conversion of color videos to black and white;
   - invert the colors;
   - increase or decrease the luminance of a video by a factor;
   - doing temporal sub-sampling.

7. Write a program, named `videoConvert`, that can perform the conversion between videos in RGB, YUV/YC_bC_r 4:2:0 to YUV/YC_bC_r 4:2:2 and YUV/YC_bC_r 4:4:4 and vice-versa. The goal is to generate new files in the desired format, independently of the input. Adapt the class `Video` to have new methods to save videos in different formats.

8. Elaborate a small report, where you describe all the steps and decisions taken in all the items of the work. Include also all the measures that you have obtained, both in terms of processing time and video comparisons.