Lab: Multimedia Information Encoding

Universidad Carlos III de Madrid, Multimedia, Year 2020-2021

Introduction and goals

The analysis and processing of multimedia data is a field that offers a wide variety of applications. One of the most straightforward applications of multimedia processing is coding, which coarsely consists on transforming a multimedia signal so it can be used and saved in digital devices.

In this lab, the student will develop a deeper knowledge on those coding concepts that were explained in the lecture classes. The design of coding strategies strongly depends on the source. Thus, the lab is split into 4 parts, where each part presents one of the multimedia information sources that were studied in class: speech, audio, image and video.

For some of these stages, coding will be required, while for others you will just have to run the provided code and reflect on the displayed results.

The main goals of the project can be summarized in two points:

- Review the main concepts of multimedia information coding that were studied in the lectures.
- Achieve a deeper knowledge on those concepts through the analysis of a set of case studies.

Project delivery ⚠

Important: for this project to be evaluated, it is necessary to upload a compressed file (zip format) to AulaGlobal containing the following:

- A document (**pdf**) following the format set by the file **lab_template** (available in AulaGlobal). The document (compressed in zip) will have a **maximum of 10 pages** and must contain the answers to the questions stated in the the following sections. **The use of figures and images to ilustrate and explain the arguments in your answers is strongly recommended**.
- The material provided with the lab, following the same hierarchy of directories and containing the scripts that the group needed to add or modify.

  Note: Only one upload is required for the projects evaluation, i.e., only one of the members of the group has to submit the zip file to AulaGlobal.

Speech Coding

The goal of this section is to improve the understanding of the main ideas about speech coding and vocoders that were studied in class. The material to do this part is in the directory `speech`. Change the Matlab's working directory to this location. Two scripts are provided (***spcoding1.m*** and ***spcoding2.m***), and a directory containing some sample signals (***speech_samples***).

Open the script ***spcoding1.m***. The script works with a speech file whose path is given by the variable ***afile*** (use the path `'./speech_samples/sentence.wav'` for now). First, the **audioinfo** function is used to obtain a struct that contains the information provided in the header of the audio file. Access its fields and explore the information they contain.

> **Q1**. Obtain the sampling frequency and the number of bits per sample of the file `sentence.wav`.

The ***audioread*** function loads the samples of the audio file in a vector ***x***. Finally, the ***stft*** function performs a sliding-window analysis of the speech signal. Run the script ***spcoding1.m*** with the file `sentence.wav`. The figure shows 3 plots. The first one shows the whole signal and the position of the analysis window. The second plot shows the current analysis window in the temporal domain. The third plot shows the analysis window in the frequency domain. You can press any key to shift the window forward through the speech file. To stop the script, click on the command window and press Ctrl+C.

> **Q2**. Why the frequency axis of the spectrum plots goes from 0 to 4000 Hz? Justify your answer.

> **Q3**. Look for a frame containing a clearly voiced sound. Explain the features that you have observed to reach this conclusion (in time and frequency).

> **Q4**. For the frame chosen in the previous question, compute approximately the fundamental frequency using the information of the plots: first using the temporal plot and then using the spectrum plot. Justify your answer.

> **Q5**. Look for a frame containing a clearly unvoiced sound. Explain the features that you have observed to reach this conclusion (in time and frequency).

Now run the script spcoding2.m. It is identical to spcoding1.m, unless the plot of the frequency domain also includes the spectral envelope. The goal is to study the spectral features of different speakers when they utter the same phoneme. As an example of perceptually different speakers, a female and a male speaker have been chosen (`w_oo.wav` and `m_oo.wav`).

> **Q6**. Discuss the spectral similarities and differences between the speakers in `w_oo.wav` and `m_oo.wav`. Focus on two features: fundamental frequency and position of the first two formants. Justify your answer.

Audio Coding

The goal of this section is to review the concepts of differential coding studied in class and apply them to mono-channel audio coding. The material to do this part is in the directory `audio`. Change the Matlab's working directory to this location. Two scripts are provided (***amp_cod.m*** and ***diff_cod.m***), and a directory (***audio_samples***) containing two audio signals. These signals were previously digitalized using the same sampling frequency and number of bits per sample.

It is required to compress the file `bassoon.wav` as much as possible, while minimizing the quality loss. In order to do that, to simple encoders have been implemented:

- **Option 1**: it consists on quantizing the amplitude of the signal directly, but using a lower number of bits per sample.
- **Option 2**: it consists on quantizing the prediction error (differential coding), with a lower number of bits per sample.

The first step is to perform an initial analysis of the files to encode. The script ***hist_comp.m*** computes the distribution of values of the signal for both alternatives. Run the script for the file `bassoon.wav`. The figure shows the distribution of values using **option 1** on the left and the distribution using **option 2** on the right.

> **Q7**. Assuming that the same number of bits per sample is used for both alternatives, use the histograms to discuss what approach would have more losses.

Now, the bit savings of each proposal will be evaluated. Use the file `bassoon.wav`. Run the script ***amp_cod.m***, which performs a requantization of the amplitude values of the audio signal directly, without transforming the signal. As it quantizes the signal with less bits than the original signal, it gets a bit saving. Make sure you understand the code of the script. Modify the variable `nb` to experiment with different numbers of bits per sample.

> **Q8**. What is the lowest value of `nb` for which you start to perceive the quantization noise?

Run the script ***diff_cod.m***, which transforms the signal before performing quantization. In particular, the algorithm uses a predictor to estimate the value of the signal on each time instant. This estimation is subtracted from the original value of the signal. The resulting signal is the residual of the prediction, which is subsequently quantized. Again, the number of bits per sample is reduced to save bits. Make sure you understand the code of the script. Modify the variable `nb` to experiment with different numbers of bits per sample.

> **Q9**. What is the lowest value of `nb` for which you start to perceive the quantization noise? Compare it with the result of **Q8** and try to explain the results.

> **Q10**. Visualize the differential coding diagram of the slides. Match each of the blocks in the diagram with the line of code that performs its task en ***diff_cod.m***. Justify your answer.

> **Q11**. Which piece of information is used by the predictor to estimate the next value of the input signal? Justify your answer.

Modify the scripts ***amp_cod.m*** and ***diff_cod.m*** to compute the Mean Square Error (MSE) between the original and the reconstructed signal. Additionally, implement the computation of the compression ratio.

> **Q12**. Obtain the MSE and the compression ratio of each proposal for the corresponding value `nb` obtained in **P8** and **P9**. Relate the results with `nb`.

Now run hist_comp.m with the file `wng.wav`.

> **Q13**. Compare the histograms obtained in this case with those obtained for `bassoon.wav`. What causes this difference? Justify your answer.

Image Coding

The goal of this section is to improve the understanding of the student about the bases of image coding using transforms. To do that, the analysis of a set of case studies is proposed, using the Discrete Cosine Transform (DCT) explained in class. The material to do this part is located in the directory `image`. Change the Matlab's working directory to this location. Two scripts are provided (***imcoding1.m*** and ***imcoding2.m***), and a directory (`data`).

Load the file `filters.mat` and visualize the 4 provided filters (use the ***imshow*** function). The goal of these filters is to select a specific region of the spectrum of spatial frequencies of the image. When the transform of the image is multiplied by one of the filters, the frequencies that are multiplied by 1 are preserved and those multiplied by 0 are deleted. Thus, the information contained in certain set of frequencies can be analysed from the preserved image content.

Run the script ***imcoding1.m***, which applies each of the mentioned filters to the original image and shows the result in the spatial and frequency domain. *Figure 1* shows the original image and its DCT. The other figures show the filtered image in both domains too.

> **Q14**. Observe the result of applying the first filter. Compare the original image with the filtered one. What image content has been preserved? What image content is lost? Relate the filtered image with its spectrum. Justify your answer.

> **Q15**. Repeat **Q14** for the rest of the filters.

The JPEG format does not compute the DCT over the whole image. Instead, the image is split into 8x8 blocks and the DCT is computed on each block. This method offers greater bit savings. Run the script ***imcoding2.m*** and visualize the result of computing the DCT, now block-wise.

> **Q16**. What image locations contain blocks where only low frequencies have significant values? Discuss your reasons.

Q17. What image locations contain blocks where middle and high frequencies have significant values? Discuss your reasons.

Video Coding

The goal of this section is to reinforce the understanding of the student in motion estimation for video coding. To do that, this prediction method is going to be evaluated in a set of video sequences. The material to do this part is located in the directory `video`. Change the Matlab's working directory to this location.

The material of this part shows the following directories:

- `videos`: it contains the video sequences that will be analysed in the questions.
- `toolbox`: it contains the required code to perform motion estimation.
- `outputs`: after running the code, it will contain the frames of the analysed video sequences with the computed motion vectors.

Additionally, the root directory provides the script ***vcoding1.m***, which calls a function to compute the motion estimation of a sequence that is given as a parameter. You can use `help` to obtain a description of the function and its parameters.

IMPORTANT: before running ***vcoding1.m***, you have to compile the code of the motion estimation toolbox. To do that, run the script ***compile.m***.

Q18. Run the script for the sequence `hall_monitor` and visualize the result. Describe the motion that appears on the scene. Can it be completely modelled by the motion vectors? Justify your answer.

Q19. Run the script for the sequence `flower` and visualize the result. In this sequence, the camera performs a *pan* that causes the whole scene to shift. Is this completely registered by the motion vectors? Why? Justify your answer.

Q20. Run the script for the sequence `tennis` and visualize the result. Focus on the moment of the *zoom out*. What pattern is shown by the motion vectors when that event happens? Is there any area in the scene where the motion vectors don't show that behavior? Why? Justify your answer.

Q21. Run the script for the sequence `coastguard` and visualize the result. Some water areas show some motion and some others don't. Why? Justify your answer.