FEUP

Applications and New Services, MM0050
Multimedia Systems and Applications - M.EIC026
2023-2024

Lab Assignment 1

Assignment Name: Sampling and Quantisation of media signals

Due Date: February 22th, 2024. Submit in Moodle your report file in PDF and all the Matlab or notebooks/Python scripts with the code you have developed as a single compressed archive.

Purpose

This work intends to apply and consolidate knowledge about the representation of media signals in the digital domain; to get acquainted with the effects of sampling and quantisation on the quality of the digital signal.

Why is it important that multimedia students gain the listed skills and knowledge? Because it allows students to gain a better understanding on the principles and effects of sampling and quantisation of media signals. The intention is to learn the effects that those processes have on the quality of the signal by varying the sampling rate and the number of quantisation levels used to represent te acquired samples. This is important because it conditions the quality of the digital representation of the real world.

Skills

In this assignment, you'll be learning how to:

1) develop simple practical algorithms using Matlab or Python to manipulate the process of sampling and quantisation;
2) apply filters prior to sampling;
3) distinguish between the type of distortions introduced by each of the two process involved in the digitisation.

Knowledge

In this assignment, you'll be learning about:

1) the processes applied to media signals to convert them from analog to digital;
2) the effects of sub-sampling and re-quantising in audio signals;
3) the role of filters applied to audio signals prior to and after sampling.

Task overview

To conduct this work, audio signals (speech or music) may be acquired with the mp3 format and stored with the Wave encapsulation format or to use already available signals with the Wave format (found in the Moodle of the course, in the folder "audio files"). To those signals it will be applied sampling and quantisation operations with different values and using or not filters. This will be achieved by using scripts Matlab available on Moodle in the folder "scripts" or by running and modifying Python notebooks. By comparing the quality of the sound generated by the different algorithms and with the different parameters' values the students should acquired a better understanding of the role played by them on the final quality (sampling, quantisation, pre and post-filtering).

It is advised to use headphones to better evaluate the quality and not interfere with the experiences of the other colleagues.

Implementation work to be carried out

1. Optional/Introductory part that may be developed outside the classroom

Select a music file mp3, open the program VLC media player[1] and select the option "File>Convert/Stream…"; in the section "Choose Profile" select "Audio- mp3" on the left and then click on the right "Customize"; to select the "WAV" encapsulation method; press the option "Audio Codec", select "WAV" for the codec and choose a sampling rate ("Samplerate") of 11025 Hz; press "Apply" and when going back to the previous window, select an mp3 file in the option "Open media" (you may also "drag-and-drop" the mp3 file you want to work with). Finally select "Save as File" indicating the desired filename and the location.

Repeat these procedures but now selecting a sampling rate of 44100Hz (audio CD quality).

Playback each of the stored files and compare their quality. Take note of the major differences that you have noticed. Include that information in your report.

2. Variation of sampling frequency with or without filters

In this part you are going to work with Matlab using the scripts "amostragemInterp_semFiltro.m" e "amostragemInterp_comFiltro.m" available on the course Moodle. These programs are modified versions of programs obtained from the Web site of the course Multimedia Communication Systems I of the Polytechnic University in Brooklyn.

Before starting your work, analyse the code of the available programs to better understand the operations performed. To help you in this task, flowcharts of the programs are provided at the end of this report. Remember that, every time that you do not understand the operation of a given Matlab function or the type of parameters it expects to receive or the output it produces, you may write in the command window of Matlab "help name_of_the_function".

As input files you should use uncompressed audio files with the WAVE format ("xxx.wav"). You will find some examples on Moodle in the folder "audio files". You may also convert your own mp3 file adopting the procedure explain in the previous point selecting the largest possible sampling rate (44100Hz).

Start Matlab and change to your own working directory. Copy all files (scripts and audio files) to that directory.

i. run the script "amostragemInterp_semFiltro.m" using a .wav file. For example, if you want to use the file "inputSound.wav", want to save the processed sound in the file "outputSound.wav" and perform a 4 to 1 sub-sampling, you should execute the following command on Matlab:

   >>amostragemInterp_semFiltro('inputSound.wav','outputSound.wav',4)

ii. Compare the original sound and the processed sounds after sub-sampling and interpolation in terms of perceptual quality, waveform and spectrum. Evaluate the obtained mean square error (note: MSE=Max/10^(PSNR/20)). Run the script again, now performing a subsampling with a factor 2. Compare the results obtained in the two experiences.

iii. Repeat now the same experiences but with the script "amostragemInterp_comFiltro.m". Compare the results from the perceptual and objective viewpoints. Which of the programs leads to a better results for the same value of k? Analyse the frequency response of the used filters. Which should be ideally the cut-off frequency of these filters? Verify if they present sufficient attenuation in the desired cut bands.

[1] http://www.videolan.org/vlc/

In the report you should provide answers to the questions raised above, present the relevant waveforms of the signals and comments of your perceptual and objective analysis, comparing the results obtained in the different experiences. Filtering experiences

3. Quantisation experiences

In these experiments you will use the script "quant_uniform.m", which allows to quantise an audio signal using a variable number of quantisation levels or bits per sample, defined by the user as an input parameter.

Run the program "quant_uniform.m" using as an input signal an uncompressed sound with the wave format just like in the previous experiments but making sure that that signal has been digitised using 16 bits per sample (PCM format: 16 bits per sample and 44,1kHz for te sampling frequency). If you use file called "inputSound.wav", want to save the processed sound in the file "outputSound.wav" and perform a quantisation with 256 levels, you should execute the following command in Matlab:

```
>>quant_uniform('inputSound.wav','outputSound.wav',256)
```

Run the script twice using 256 and 16 quantisation levels. Compare the results among them and also individually against the original sound in terms of perceptual and objective quality. Compare the quantisation errors obtained.

4. Dithering experiences (bonus)

Develop a script that introduces dithering in the audio wave file when re-quantising to a lower number of levels. Check the Matlab functions available.

Criteria for Success

It is important that you get acquainted with the the two operations that need to be performed for digitising media signals, notably audio. Upon conducting this assignment you should be able to describe the effects that sampling has on the quality of the signal, discuss the ability of the human ear to perceive differences in sampling rates, the type of distortions that may arise due to the sampling process and how to minimise them. Likewise, you should have a clear understanding of the quantisation operation, the impact on the quality of the signal, discuss the severity of distortions introduced and perceived by the human ear according to the type of content and quantisation levels and explain where the distortions come from. You should also understand at a high level the dithering technique and discuss its benefits. Finally, you should understand the practical aspects of performing filtering.

Report should be delivered up to 22 February in Moodle.

Note: The template used to describe this assignment is adapted from the University of Las Vegas, Nevada, and the [Transparency in Learning and Teaching (TILT) Project.]

Annex 1 - Flowcharts of the Matlab scripts

amostragemInter_semFiltro

- ler/importar para o Matlab o ficheiro original de som
- tocar o som importado

- executar uma sub-amostragem (*down sampling)* do ficheiro original de som
- gravar num novo ficheiro o resultado sub-amostrado
- tocar o resultado sub-amostrado

- executar uma interpolação do ficheiro original de som
- gravar num novo ficheiro o resultado interpolado
- tocar o resultado sobre-amostrado

- apresentar no écran o espectro dos três sinais (original, sub-amostrado e interpolado)

- calcular o MSE entre o som interpolado e o som original

quant_uniform

- ler/importar para o Matlab o ficheiro original de som
- tocar o som importado

- obter o passo de quantização a aplicar

- aplicar quantização uniforme utilizando o passo de quantização especificado

- guardar em disco o som re-quantizado e tocá-lo

- apresentar diagrams com as formas dos sons original e re-quantizado