Audio Coding - Practice Lessons

General information

Instructors:

Oleg Golokolenko oleg.golokolenko@tu-ilmenau.de phone: 69-1673

Website:

http://www.tu-ilmenau.de/mt/lehrveranstaltungen/lehre-fuer-master-mt/audio-coding/

Moodle: moodle2.tu-ilmenau.de

- Check for updates
- News (schedule changes, etc.)
- Homework tasks
- Lecture slides

Audio Coding - Practice Lessons

Additional date for Practice Lessons

- Seminar dates:
  - Tuesday, 15:00-16:30 Sr K 2003B (odd weeks)
  - Tuesday, 15:00-16:30 Sr HU 010 (even weeks)
- **Enrollment in Doodle via**

https://doodle.com/poll/zd4udqp8fet3gsxw

Audio Coding - Practice Lessons

Homework

- Counts 30% towards the final grade (midterm and final written exam 70%)
- Assignments will be published on the website, on Moodle and during the practice lessons
- You are requested to bring along the solution for the tasks and present it. There it will be reviewed and graded
- A group of max. 3 students may deliver one single solution but all of them must be able to explain it
- Homework **submissions by e-mail** will **not be accepted!**
- If there are questions regarding the task, please feel free to ask via moodle forum, e-mail, phone, etc.

Audio Coding - Practice Lessons

Additional hints

- The grades you will get for the homework will only be taken into account in the current and in the following semester. They will be valid until the end of the summer semester 2018
- Your final grade will consist out of the points collected from midterm and final exams along with the homework points
- ***Midterm Exam – first week of December***

Audio Coding - Quizzes

Moodle2 Quiz

- Published on moodle2.tu-ilmenau.de
- Log in with your regular TU account:
  - Fakultät EI → Institut für Medientechnik → FG Angewandte Mediensysteme → Audio Coding
- Enroll in the Audio Coding course
- Quizzes are published weekly after the lecture
- Solve the quiz in order to recapitulate the lecture and to get points
- Submit within one week after publication!
- The Moodle quizzes contribute 25% to the final seminar grade (Python homework solutions 75%)

$$\mathit{Final\_grade} = 0.3*(0.25*\mathit{Quizzes}+0.75*\mathit{HWs}) \\ +0.7(0.5*\mathit{Midterm\_Exam}+0.5*\mathit{Final\_Exam})$$

Seminar goals and rules

Goals:

- Connect theory knowledge with practice
- Get practical experience in audio codec design

Rules:

- **Stand-alone** coder and decoder programs
- Save encoded audio samples into **binary files**

  → to evaluate the program and compare with future improvements
- Play out decoded audio using **Python** (verification)
- **Use of functions**
- Delayed submission without justified reason = -20% from your grade
- Every next attempt = -10% from your grade

Homework Assignment 1

Basic Python Audio processing

- Take an audio signal *Track16.wav* (download from moodle)
- Open/read a *.wav audio into array; find out **sampling frequency**, **number of bits** and print them out to **console**
  - Hint – to read wav-file use *scipy.io.wavfile*
  - Be careful: the name of the audio file is a string, you must write „name.wav“
- Play back the audio at the correct sampling frequency
  - Hint – to play audio use *pyaudio* or *sound.py*
  - Be careful: *pyaudio stream.write()* needs string values!!
- **Extract 8 seconds** from the middle of the audio data.
  - Be careful: you need to translate first the seconds into samples!
- Normalize the fragment (**take into account data type and its range**)
- Plot and play both channels separately to check you have not chosen a silent fragment:
  - Hint - „matplotlib.pyplot“

Homework Assignment 1

Task 2: Fast Fourier Transformation (FFT)

- Use previous audio signal (only one channel)
- Process it block wise, by taking 4 consecutive blocks of 1024 samples
- Apply the FFT to it
- Plot the magnitude of it (The 4 different FFT blocks in different colors on top of each other)

Homework Assignment 1

Task 3: Quantization

- Write a python function „`encframewk.py`‟ that reads `"Track16.wav"` audio file with 16 bit per sample and stores the samples with 16 bit per sample in a binary file `"encoded.bin"`
  - **Hint:** for saving file you can use library pickle with command `pickle.dump(array, open(“file_name‟, “wb‟), 1)` or such
- Write a python function „`decframewk.py`‟ that reads `"encoded.bin"`, and stores clean data into 16 bit samples back into the `"decoded.wav"`
  - Hint: you can use for loading `"*.bin"` file `pickle.load(file)`
- Observ if the size of the `*.wav` and the `*.bin` files are different?

Homework Assignment 1

Task 4: Quantization

- Extend the framework into "encfamewk8bit.py" and "decframewk8bit.py"
- Store its encoded binary data into a binary file "encoded8bit.bin‟
- To increase compression, and hence to obtain a smaller binary file „encoded8bit.bin‟ (half the size), in the "encframewk8bit.py" quantize the audio samples from the sound file to 8 bit per sample
- Use uniform quantizer

Homework Assignment 1

Task 4: Quantization

- In the `"decframewk8bit.py"`, de-quantize the samples from the binary file and store it into the sound file `"decoded8bit.wav"`.
- Plot the original sound wave from `“Track16.wav”` using `"matplotlib"`, and also plot the sound wave from `"decoded8bit.wav"` on top of each other (only one channel).
- How do they compare?
- Listen to the original 16 bit and the 8 bit version
- Can you hear the quantization noise in the 8 bit version?

Rules

1) Use of functions (for quantization, processing).
2) The code has to be reusable and flexible.
3) Encoded data has to be stored into *.bin file
4) Use the same audio file for comparisson
5) You have to able to compare *.bin file with previous HW
6) Plots should containe: A title, axis names with correct measure units; if there are several signals on the plot – use of Legend