Tecnologías Multimedia - Study Guide - Milestone 6: Compressing the audio data with zlib

Vicente González Ruiz - Depto Informática - UAL

November 10, 2020

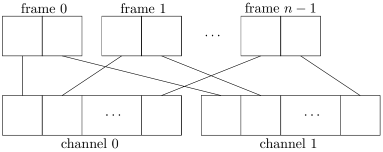

Figure 1: Sample reordering to create two independent channels.

1. Description

It's time to reduce bandwidth comsumption. The `pack()` and the `unpack()` methods can compress and decompress, respectively, the chunks that are handled. To compress and decompress, we will use a free codec named DEFLATE, which is based on LZSS and Huffman Coding [1].

2. What you have to do?

1. Create a class named `Compress`, that inherits from `Buffer` (the class implemented in the previous milestone), in which the methods `pack()` and `unpack()` are overriden to compress and decompress the chunks. Use the Python's standard library `zlib`. Store this class in a module named `compress.py`.

2. Compress (and decompress) each chunk as a unit (each compressed chunk will be transmitted in a different UDP packet). In order to increase slightly the (data) compression ratio, reorder the samples as it is shown in the Figure 1.

3. Timming

Please, finish this milestone at most in one week.

4. Deliverables

Create a Python module named `compress.py` and store it in the root directory of your `intercom`'s repo.

5. Resources

[1] Nelson M. and Gailly J. *The Data Compression Book*. M&T Books, 1996.