CMPT 365

Final Project: FLAC Spectrogram Generator

Mackenzie Craig (301 371 202)
mlcraig@sfu.ca

Motivation
My goal for this project was to create a tool that could be used by music fans and digital audio collectors like myself to quickly and easily detect lossy transcodes in music visually. I wanted to create a command line program to quickly create a spectrogram image for viewing with little hassle and configuration parameters.

Problem
Transcoded audio can often have problems that are not immediately visible by simply looking at the file metadata, or by quickly listening to a section of the file. For example, a lossless file could be transcoded to a lossy format and then back into a lossless format. While the final file is stored in a lossless format, loss of quality has still occurred during this transcode. Most people who collect or archive high fidelity audio would not accept this type of file for their collection, so they need tools to spot these types of transcodes.

Solution
One solution to this problem is to visually analyze the file in question. This can be done by generating a spectrogram of the song and looking for visual anomalies showing transcodes. A spectrogram charts frequencies over time. The x-axis of a spectrogram is time, while the y-axis represents each individual frequency present in the file. A heat-like coloration is used to show intensity or volume of each given frequency (the lighter the color the more intense).

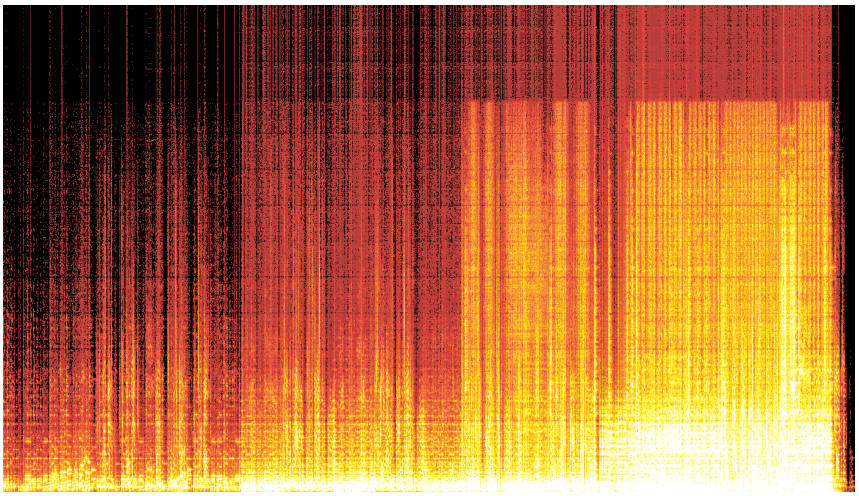

Figure 1: A lossy transcode of Led Zeppelins "Stairway to Heaven"

Note the clear horizontal cutoff at a frequency about 4/5 up the spectrogram pictured in Figure 1. This is a dead giveaway that a lossy transcode has occurred.

This solution is not novel. Spectrograms have been commonly used for this purpose throughout high fidelity audio communities for years. My solution is an implementation of generating spectrograms for the FLAC audio format.

Evaluation
When compared to other spectrogram generation software, my program generates consistent visual results. While its runtime is admittedly slower on average, I was more concerned with the accuracy of the generated results for this assignment.

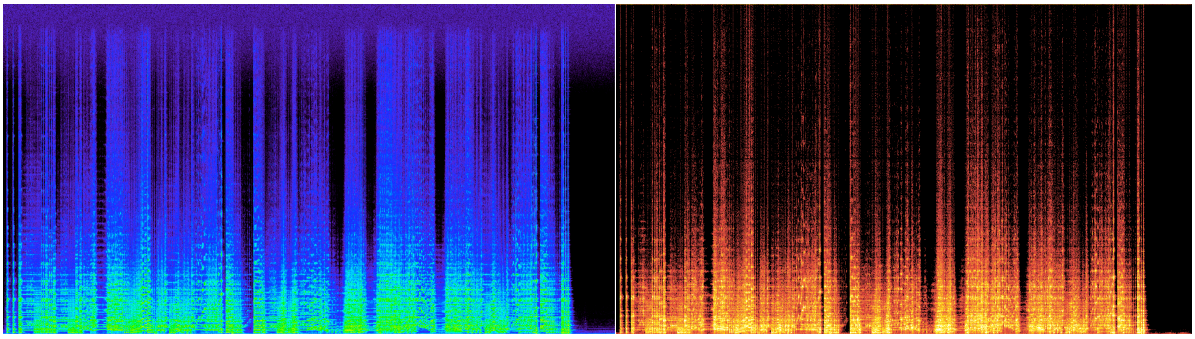

Figure 2 Results of a song from "Spek" open source spectrogram software (left), and my program (right) for the same song

While other programs use various different color schemes, my programs results are structurally consistent with those of other popular spectrogram generators. Figure 2 shows the spectrograms generated from the file "music/classic.flac" from the provided sample audio.

Instructions

To run the software, no dependencies need to be installed. All library code used is included in the project as a single header file. Note: library code is only used to read samples from FLAC files. No libraries are used for any other aspect of the program.

Please download the full project from: https://github.com/tehp/flacspec (includes sample audio).

To compile the program, run "make".
To run the program with a given song, type "./flacspec [filename]"
For example: "./flacspec music/classic.flac"
or: "./flacspec music/zeldas_theme.flac"

The output is stored in the file "output.ppm".
To view this file, either install or use a PBM image viewer for your OS, or use the online PBM image viewer at: http://paulcuth.me.uk/netpbm-viewer/ .

Here are the reproducible results of the following run:
"make && ./flacspec music/zeldas_theme.flac"