Computer Science Tripos – Part II – Project Proposal

Audio to musical note transcriber

Sebastian Ion, Robinson College

Originator: Sebastian Ion

October 16, 2018

Project Supervisors: Dr A. Mycroft & Dr M. Kuhn

Director of Studies: Dr A. Mycroft

Project Overseers: Dr G. Winskel & Dr R. Mortier

Introduction

The basic idea of the project is to employ Digial Signal Processing methods for pitch detection in WAV files representing monophonic music from a flute or recorder. The output would then be a MIDI-style file which will contain a series of *ON* and *OFF* events for the identified musical notes.

The choice of the instrument was made while taking into consideration the characteristics of the soundwave produced by different musical instruments. A flute/recorder produces 'clean' sounds (almost no distortion, harmonics, or overtones) which makes pitch detection easier. For other instruments, such as guitars, the likelihood of the harmonics being 'louder' than the base note is very high.

Starting point

I have gathered some sources (books, previous year's lecture notes) on Digital Signal Processing. I have also looked into and downloaded existing software for pitch detection which will be used for comparison in the evaluation stage: AnthemScore[1], ScoreCloud4[2], Transcribe![3], Melody Scanner[4], IntelliScoreEnsemble[5], AudioScore[6]

As most of the software previously listed is only available as trial versions they will not be used as the primary means of evaluation.

I have downloaded a set of 20 WAV files each containing short riffs (around 10 seconds/riff) played on the recorder. They are fully annotated (with music sheets).

[1]https://www.lunaverus.com/home
[2]https://scorecloud.com/
[3]https://www.seventhstring.com/index.html
[4]https://melodyscanner.com/
[5]http://www.intelliscore.net/index.html
[6]http://www.sibelius.com/products/audioscore/ultimate.html

Resources required

I will be using my personal machine (dual boot Windows 10/Ubuntu 16.04 LTS, Intel Core i7 2.60GHz, 8GB RAM, 1TB SSD + 1TB HDD) for development. The project will be kept under a version control system (Git) and hosted on GitHub and GitLab. A weekly backup will also be stored on the MCS cluster. In case of my laptop breaking I will clone the git repository and continue my work on an MCS machine in the Computer Laboratory. If the third party pitch detection software is non recoverable, then only Melody Scanner will be used for comparison as it is an online tool.

Work to be done

The initial steps of the project would be to research the techniques needed for pitch detection. A major part of the project will consist of implementing and experimenting with different techniques.

To ease implementation and testing the initial code will instead be built to identify the musical notes from pure soundwaves. After this component is working, the code will be tested on the set of WAV files and then changed accordingly to deal with harmonics (present in real instruments/sounds) and noise (caused by the recording device, compression, filetype conversion, etc.)

The basic idea would be:

- Apply FFT to break up the signal and use spectrum density estimation to infer frequency in the sampled signal;
- Apply thresholding to the resulting signal;
- Take the maximum energy level from each time slice and compute the base note;
- Produce a MIDI file with the musical notes that have been identified. *A*dditionally, produce a log of the notes that have been identified in order to compare with the ground truth;
- Play the MIDI file and compare with original. Compare the music sheet (given with the set of WAV files) to the log of notes that have been generated. Run the third party software on the same file and compare results (both sound-wise and note-wise).

Success citerion

The project will be a success if every note is identified correctly in at least 80% of the files in the test suite (i.e. 16/20).

Possible extensions

Extensions will be considered in terms of the amount of time that is still available with each being evaluated separately.

Some potential extensions:

- Identify the musical notes from monophonic music produced by another instrument (e.g. guitar, piano)
- Identify the musical notes from polyphonic music
- Enhance with beat detection (for monophonic music)
- Produce a MusicXML file as output (requires previous extension); this would allow the creation of a standard music sheet for a given audio file

Timetable

Planned starting date is 22/10/2018.

1. **Michaelmas weeks 3–4 (18nd October - 31st October)**
   - General project setup: version-control, development evironment
   - Investigate programming language to be used. The main options would be Julia, Python, and MATLAB.
2. **Michaelmas weeks 5–6 (1st November - 14th November)**
   - Research and plan which DSP techniques will be used
   - Begin implementation of those techniques
3. **Michaelmas weeks 7–8 (15th November - 28th November)** Finish initial implementation and test on pure sine waves
4. **Michaelmas vacation (29th November - 16th January)**
   - Run existing code on the test suite and compare audio with original and notes with the provided music sheet.
   - Start fixing any issues that might arise (might include more research)
5. **Lent weeks 1–2 (17th January - 30th January)**
   - Write progress report and prepare for the presentation.
   - Finish core component of the project
6. **Lent weeks 3–5 (31st January - 20th February)**
   - Run code on more audio files and check results (start fixing if results not satisfactory)
7. **Lent weeks 6–8 (21st February - 13th March)** Prioritise extensions based on remaining time, technical difficulty, and degree of interest/improvement
8. **Easter vacation (14th March - 24th April)** Evaluate extensions and pass full draft dissertation to supervisors and Director of Studies.
9. **Easter weeks 1–2 (25th April - 8th May)** Adjust dissertation in light of comments and run additional performance tests.
10. **Easter week 3 (9th May - 15th May)** Proof reading and then an early submission so as to concentrate on examination revision.