Transforming Audio Files to MIDI Files

Jake Fox, Siobhan Plouffe, Luke Nash, Aine Ryhn

UNIVERSITY of ROCHESTER

Introduction

As musicians, it is vital to be able to dictate the music being played into writing. Checking dictations solely through your ear is not enough, which is why a transcription MatLab program has been implemented. This program takes a wave file as an input and exports a midi file, which then can be converted to sheet music through third party applications such as MuseScore.

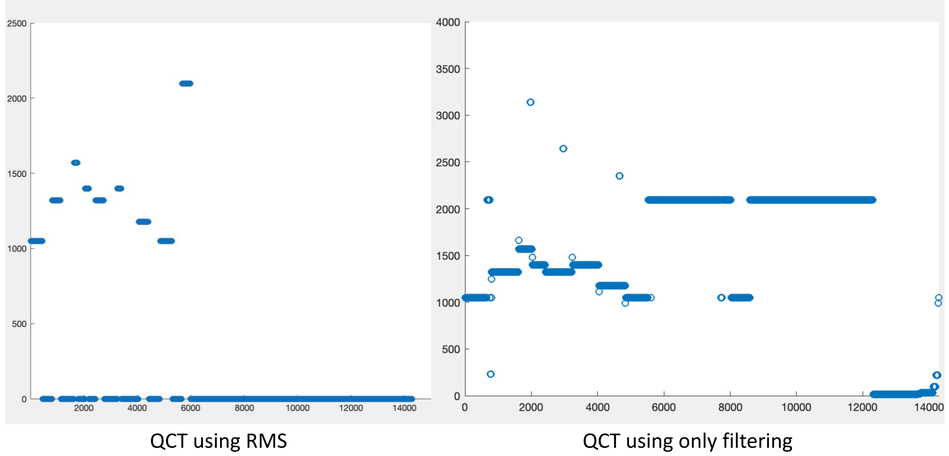
Methods

Pitch detection was implemented based on the Constant Q Transform (CQT), which is a variation of the DFT.

Quantity	CQT	DFT
Center frequencies	exponential in k $f_k = f_0\left(\sqrt[b]{2}\right)^k$	linear in k $f_k = k \cdot \Delta f$
Window length	variable $= N(k)$	constant = N
Filter bandwidth	variable $= f_k/Q$	constant $= f_s/N$
Resolution	constant = Q	variable = k

Table 3.1: Comparison of CQT and DFT.

The CQT array was normalized and filtered using the RMS of the peak amplitude of the signal. A comparison of using RMS vs solely using filtering can be seen below.



QCT using RMS

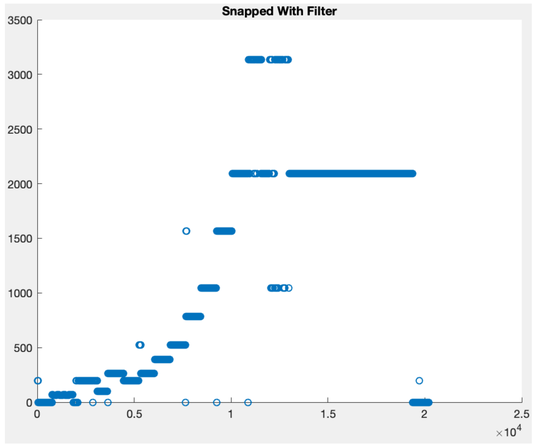
QCT using only filtering

Pitch was determined in two ways, depending on whether the input was monophonic or polyphonic : (1) pattern recognition and cross correlation methods or (2) bispectral analysis.

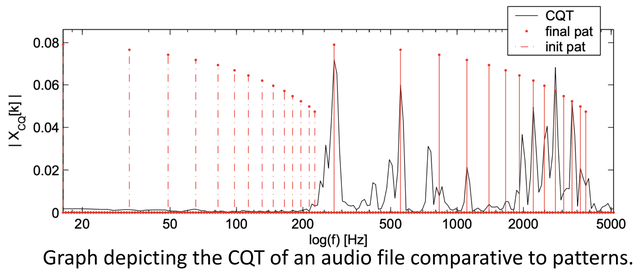
$$B_x(\eta_1,\eta_2) = X(\eta_1)X(\eta_2)X^*(\eta_1+\eta_2) = \\ = \left(\sum_{k=1}^{4}\delta(\eta_1 \pm f_k)\right)\left(\sum_{l=1}^{4}\delta(\eta_2 \pm f_l)\right) \\ \left(\sum_{m=1}^{4}\delta(\eta_1+\eta_2 \pm f_m)\right).$$

Equation used to implement polyphonic transcription

Methods (cont.)

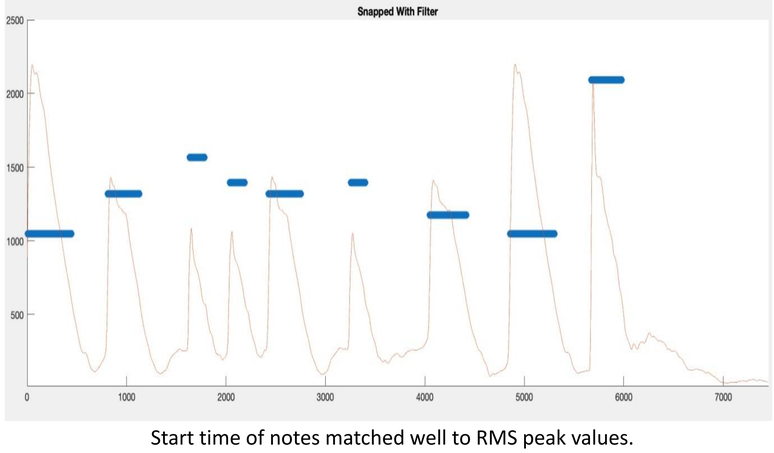


Monophonic detection using pattern recognition and cross correlation



Graph depicting the CQT of an audio file comparative to patterns.

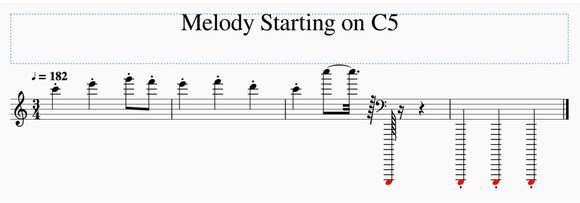
Starting times of notes were determined by RMS peaks, and end times were determined by zeros following the frequency. The note duration was then set to be the difference between the two.



Start time of notes matched well to RMS peak values.

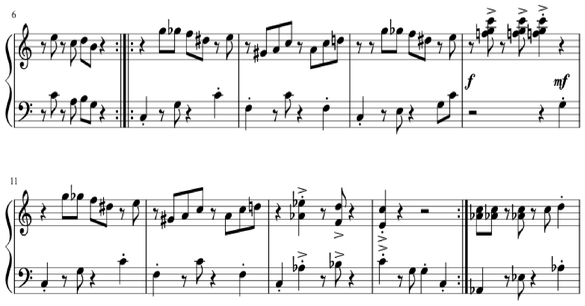
Results

Monophonic transcription was obtained, as seen below.



Future Work

Future goals include completion of polyphonic detection, as a velocity detection. These can make the MIDI files more accurate, and can improve the overall characteristic of the file.



References

[1] Schutte, Ken. “Matlab and MIDI.” *Ken Schutte*, 2012, kenschutte.com/midi.

[2] Wolfe, Joe. “Note Names, MIDI Numbers, and Frequencies.” *Note Names, MIDI Numbers and Frequencies*, 2005, newt.phys.unsw.edu.au/jw/notes.html.

[3] “Note Input.” *MuseScore.org*, 2018, musescore.org/en/handbook/note-input#enter-pitch.

[4] Vass, Jiří. “Automatic Transcription of Audio Signals.” *Czech Technical University in Prague*, Czech Technical University in Prague, 2204, pp. 1–66.

[5] Brown, Judith. “Musical Fundemental Frequency Tracking Using a Pattern Recognition Method.”*Musical Fundamental Frequency Tracking Using a Pattern Recognition Method*, academics.wellesley.edu/Physics/brown/pubs/cqptrv92P1394-P1402.pdf.