Using Data Visualization to Pinpoint Mistakes in MIDI Practice Recordings

Jeremy Grifski and Stephen Wu

Abstract— When a musician wants to practice their instrument, they often have to rely on their peers or an instructor to help them isolate mistakes in their technique. As an alternative solution, we are proposing a system to answer the following question: how can we leverage data visualization to pinpoint mistakes in music data? For the sake of scope, we have chosen to focus on MIDI recordings.

Index Terms—Music, Data Visualization, MIDI.

1 Introduction

Music is a profession and hobby enjoyed by many people. Unfortunately, the field hasn't received a lot of attention from the technology community. To this day, musicians still practice their instruments with little to no benefit from technology.

One area of music that could really benefit from a technological upgrade would be practice. After all, practice is usually something that occurs alone without a lot of feedback. Without access to an instructor, musicians may find it difficult to self-assess their abilities. They could all benefit from some sort of tool to help pinpoint their mistakes.

In this project, we intend to build a data visualization dashboard which can be used to compare practice MIDI files with professional MIDI files. The goal is to isolate areas in the practice file which are most unlike the professional file for the sake of improvement.

2 Research Questions

As mentioned previously, the major research question we will be looking to address is the following: how can we leverage data visualization to pinpoint mistakes in MIDI practice recordings?

Naturally, this question raises several underlying questions such as:

- What are practice areas and quantifiable data (pitch, tempo, etc.) that we can gleam from MIDI recordings?
- What are the most effective ways of visualizing those practice areas?
- What are our options in analyzing MIDI files to visualize MIDI events in a useful manner for musicians?
- How useful is comparing MIDI recordings via velocity, sustain, and note frequency over time graphs
- Can we algorithmically generate useful automated feedback from analysis of these MIDI recordings and graphs

In an effort to pinpoint mistakes, we will need to find the best ways to represent our musical data, so the user will see value in the tool.

- *Jeremy Grifski is a student at The Ohio State University. E-mail: grifski.1@osu.edu.*
- *Stephen Wu is a student at The Ohio State University. E-mail: wu.2719@osu.edu.*

Manuscript received xx xxx. 201x; accepted xx xxx. 201x. Date of Publication xx xxx. 201x; date of current version xx xxx. 201x. For information on obtaining reprints of this article, please send e-mail to: reprints@ieee.org. Digital Object Identifier: xx.xxxx/TVCG.201x.xxxxxxx

3 Design Goals

At a high level, we intend to construct a dashboard split into two panes: the file pane and the graph pane.

The file pane will contain a list of active MIDI files which are each given a color for encoding purposes. That means the dashboard will be able to support about 20 simultaneous MIDI files due to the limits of color perception. This should be more than enough considering the practicality of comparing that many recordings.

Each file in the file pane will be able to be selected for viewing purposes in the graph pane. When unselected, the file's background color will be neutral. When selected, the file's background color will mirror its color in the graph pane.

Meanwhile, the graph pane will contain several graphs:

- Notes versus Time (master graph)
- Notes versus Frequency
- Velocity versus Time
- Sustain versus Time

As a stretch goal, each graph will be connected with the master graph for filtering purposes. When a section of time is selected in the master graph, all other graphs will be updated to reflect the new subsection of data. This will allow a user to hone in on specific mistakes.

In addition, graphs will contain tooltips which will highlight areas with the highest amount of mistakes. These tooltips will include high level notes to assist the user in understanding the data.

Finally, the dashboard can be extended to include realtime recording and sheet music comparison.

4 Hardware and Software Requirements

To complete this project, we require the following software:

- JavaScript: a web-based programming language
- D3.js: a data visualization library
- MIDI.js: a MIDI processing library
- GarageBand: a MIDI editing tool
- GitHub: a version control and project management tool
- Travis CI: a continuous integration tool for testing

With this software, we should be able to build and test the entire system.

5 Tasks and Metrics

In order to verify the success of this project, we will be tracking several tasks in GitHub. Namely, we will be designing and implementing the following:

- MIDI File Upload
- MIDI File Pane
- Notes versus Time Graph
- Notes versus Frequency Graph
- Velocity versus Time Graph
- Sustain versus Time Graph
- Mistake Analysis for Tooltips
- Realtime Recording
- Sheet Music Rendering and Comparison

Each of these tasks can probably be broken down into smaller tasks as they all need to be designed, prototyped, and tested.

As for verifying that our design is good, we'll be testing it on musicians of varying abilities. They will first play a song to generate a MIDI file, then we will ask them to indicate any mistakes they felt they made. Finally, we will compare their personal insight to the tool.

To take the research a step further, we would need to also gather insight from professional reviewers to see how their thoughts vary from the results in the tool.

6 Conclusion

It may seem odd to want to think of music in a visual way, but we feel our system will have a positive impact on musicians who want to improve their practice sessions.