

## シンボリックドメインにおける音楽分析

Alexander Adli<sup>1</sup>, 仲尾善勝<sup>2</sup>

<sup>1</sup>琉球大学大学院理工学研究科、<sup>2</sup>琉球大学電気電子工学科

### Music Analysis in Symbolic Domain

Alexander Adli<sup>1</sup>, Zensho Nakao<sup>2</sup>

<sup>1</sup>Graduate School of Engineering & Science, University of the Ryukyus

<sup>2</sup>Department of Electrical & Electronics Engineering, University of the Ryukyus

#### Abstract

This paper is a summary of music analysis tools for high-level features extracting in MIDI files. The usage of MIDI file was motivated by the rapid advance of automatic transcription systems which is aiming to reliably get the musical events from the audio files. This means that audio files can be analyzed for musically meaningful features which can benefit various applications such as musical similarity, music recommendation, music visualizers and the like.

The paper emphasizes the created by the authors MIDI classes and their ability to extract tonality, bass, etc. and write this information inside the MIDI file as meta events. This can be used by standard MIDI players or by an advanced player like Synclayer to apply various real-time effects including visualization.

#### 1. Introduction

By music analysis we mean extracting the characteristic pieces of information known as features. Typically there are three categories of features – the low-level features related to the spectral of time-domain information, high-level features contains information that is musically meaningful, and cultural features sociocultural information out of the scope of music itself[1].

It happened that the majority of the research in music information retrieval (MIR) focused on the low-level features, probably because of the difficulties in getting reliable musical information from the audio files. However, when high-level features extraction is applied in the symbolic domain like MIDI it become much more reliable. With the advances of automatic transcriptions one can convert the audio file into MIDI format, and extract high-level features from it. The next section describes some of the available tools of high-level extraction and their purposes.

#### 2. High-level features extractors

Probably the most advanced MIDI music processing tool is the MIDIToolBox for Matlab which is an extension for the Matlab environment from MathWorks. It is designed for handling and processing MIDI files. It was introduced by Petri Toiviainen and Tuomas Eerola, and there are many publications describing its possibilities and advances[3]. The users of MIDIToolBox can analyze a MIDI file and get musical features by running the realized algorithms in the toolbox.

JSymbolic is a high-level feature extractor in JAVA. It analysis a MIDI file for 160 features and writes the result into a ACE XML or ARFF file[2]. This can be used for applications of musical similarity where a database of such XML files can be compared.

Back in late 90s, Robert Rowe[7] created a number of classes and structures for MIDI music, and realized dozens of interesting algorithms. Unfortunately most of them were written in a platform dependant way using CodeWarrior for Macintosh. Rowe himself claimed he started rewriting them for the newer XCode of MacOS.

Commercial notation programs like (Finale, Cakewalk, SoundForge...) provide tools in MIDI mainly for music compositions. There are algorithms for creating harmony from melody, or even suggesting the melodic line from databases.

### 3. MIDI Classes

The MIDI Classes created by the authors of this paper are technically described in another paper[6]. The algorithms they are based on are also describes in corresponding researches. However, what does interest us here is that what these classes provide for the music analysis. As obvious from the name they are oriented on MIDI files analysis. Unlike previously mentioned feature extractors, they include the result information inside the MIDI file itself as additional meta events. The advantage of this is that prepared MIDI players can read this information along with the music and react on it while simple players will just ignore the unknown messages to perform the music as usual. A MIDI player for real time visualization purposes were created previously by the authors and Ermakov. A. and is described in [5].

### 4. Conclusions and future research

This is a short summary of some of the tools used for high-level features extracting from symbolic music represented in MIDI.

### References

- [1] McKay, C. Automatic genre classification of MIDI recordings. *M.A. Thesis*. McGill University, Canada, 2004.
- [2] McKay, C., and I. Fujinaga. jSymbolic: A feature extractor for MIDI files. *Proceedings of the International Computer Music Conference*. 302–5, 2006
- [3] Tuomas Eerola and Petri Toivainen, "MIR in Matlab: The MIDI Toolbox," *Proceedings of the 5th International Conference on Music Information Retrieval conference(ISMIR2004)*, 2004.
- [4] Alexander Adli & Zensho Nakao, "Computer music analysis in MIDI files," *Proceedings of the computational science symposium*, Nagoya University, Japan, 2005.
- [5] Alexander Adli, Zensho Nakao and Yasunori Nagata, "A Content Dependent Visualization System for Symbolic Representation of Piano Stream," *Lecture Notes in Computer Science*, Volume 4693/2007, September 2007.
- [6] Alexander Adli, Zensho Nakao, "MIDI classes for symbolic music processing," under review.
- [7] R. Rowe, "Machine Musicianship," The Mit Press, 2001.