A SYMBOLIC KEY FINDING ALGORITHM

Yongwei Zhu

Institute for Infocomm Research
21 Heng Mui Keng Terrace
Singapore 119613
ywzhu@i2r.a-star.edu.sg

ABSTRACT

We describe a symbolic key finding algorithm, which reads in a MIDI file as input, and outputs the name of the first key that is established in the piece of music. This algorithm is based on a rule-based approach. The key is claimed based on the detection of a number of supporting evidences of a particular key in the music notes. The supporting evidence includes the pitch class of the tonic, the triad chord, the pitch set of the diatonic scale, and etc. The rules that claim a key is derived from the training examples.

Keywords: Key Finding

1 THE ALGORITHM

The algorithm is presented as a number of processing steps:

In step 1, The MIDI file is converted to note matrix using MIDITOOLBOX [1], which represent each note by the following format:

onset time, onset duration, channel, pitch, and velocity

The time and duration is measured in the unit of quarter note. And the notes are listed in sequential order based on the onset time.

In step 2, the onset time of the notes are quantized to the precision of 1/16 note, i.e. onset times that are less than 1/16 note away are set to a same onset time value.

In step 3, the note matrix is segmented to groups of notes based on the onset time, where each note group has a single onset time.

In step 4, each group of notes is translated to a pitch class feature, which specifies the pitch classes of the notes in the group.

In step 5, a structure called *pitch class set growing track* is computed. This structure depicts the accumulated pitch classes of the notes versus the number of pitch classes of the notes, which is up to 12. For example, from onset time 0 to onset time, say, 1.5, the pitch class of the notes is a single pitch class, say, C. And from onset time 0 to onset time 4, the pitch classes are C and G. So in the structure contains 12 pitch classes features, and the first pitch classes feature is C, and the second pitch class feature is C and G, and so on until the 12th pitch class feature is computed. Certain pitch class feature may be empty in the structure, either because the number of pitch classes jump more than 1 or because the total number pitch classes for the entire MIDI file is less than 12.

In step 6, certain cues of the key are detected based on the computed structure.

The tonic note is claimed in the first pitch class feature, and the number of note occurrence is larger than a threshold value 3.

A major mediant note or minor mediant note is claimed if the 2 pitch class in the second pitch class feature forms a major 3rd or minor 3rd interval.

A major tonic chord or minor tonic chord is claimed if third pitch classes feature depicts a major triad or minor triad.

A diatonic scale pitch set is claimed if the seventh pitch class feature corresponds exactly to a diatonic scale.

In step 7, the key is detected based on all the extracted evidence using a set of rules. The rules and the importance sequence of the rules are derived based on the training data.

The list of rules and the sequences will be presented in the final draft of the abstract.

In the case the first 3 pitch class features are all empty, the structure of pitch class set growing track is re-computed by skipping the first group of onset notes.

2 IMPLEMENTATION

This algorithm is implemented using Windows PC, MATLAB platform. We use MIDITOOLBOX to read in the notes from MIDI files. With Intel 1.66GHz, the execution time of key finding of 96 MIDI files is about 2 minutes.

3 DISCUSSION

The algorithm detects the key based on the beginning notes of the music, and not all notes are utilized in the detection. A key is claimed when the accumulated evidence is adequate.

REFERENCES

[1] http://www.jyu.fi/musica/miditoolbox/