Automated Transcription of Electronic Drumkits

Martin Digard

Florent Jacquemard

Lydia Rodriguez-de la Nava

4th International Workshop on Reading Music Systems
WoRMS’22
November 18th 2022
Objective & Outline

**source**
MIDI drumkit

**symbolic representation**
of performance
piano roll (MIDI file)
- only NOTE-ON Events
- pitches = Drum Elements
- unquantized onsets & durations

**target representation**
music score (XML file)
- rhythm quantization
- voice separation
- score engraving

Outline

1. Elements of Drum notation

2. Parsing Drums MIDI Recording
   - Prior Weighted Tree Grammar
   - Vertical Alignments

3. Score Engraving
   - Building an Intermediate Score Representation
   - Term Rewriting

4. Experiments and Results
   with Google Groove MIDI dataset
Drum notation emerged with drum schools for teaching. For the preservation of improvisations, no standard was established. There were 2 schools (few differences):
- Europe (Agostini)
- US ("Universal")

Readability is a crucial issue; the number of elements in drumkits can overload the notation.
Drumkit Elements

Principles:

- 1 pitch = 1 drum kit element
  height = position in space

- note head = mode

European notation

US notation
Dynamics, Ornaments

accent marcato ghost note  SD  FT  flam between HT and SD  drag (not used) 16th-note roll 32nd-note roll

corresponding to velocity  2 strikes  denoted explicitly (in extenso)
• 1 staff
• but several voices (for readability)
• voices denoted with stem direction

voice separation is performed according to a schema fixed for the whole score

• top voice:
  hand-played notes
• bottom voice:
  feet-played notes

• top voice:
  repeated rhythmic patterns (RC)
• bottom voice:
  other (sporadic) elements
Input MIDI file (unstructured)

$k$-best parsing
- rhythm quantization
- score structuring (hierarchical groups of notes, beams)

Weighted Regular Tree Grammar
defines a prior rhythm notation language
evaluation of a cost of readability

Parse Tree

Term rewriting

Parse Tree

Intermediate Score Representation (structured)

XML/MEI score file

Music 21 structure

qparse framework
https://gitlab.inria.fr/qparse/qparselib
https://qparse.gitlabpages.inria.fr

75 Kloc C++

- Command lines tools
- Python binding - Lydia Rodriguez-de la Nava
Rhythm Quantization “on Grid”

Rhythm quantization with grids, e.g. MIDI files import
• in score editors (Finale, Sibelius, Dorico, Musescore…),
• or in DAWs (Ableton Live, Logic…)
Grid based Approaches to Rhythm Quantization

Alignment of every input time point (onset) to the closest position in a *grid* = sequence of equidistant time position.

- **Grid 16th note**
  - Input: 0, 1/4, 1/2, 3/4, 1 beat
  - Alignment: 0, 1/4, 1/2, 3/4
  - Illustration: poor fit, good readability

- **Grid 32th note**
  - Input: 0, 1/8, 1/4, 3/8, 5/8, 3/4, 1
  - Alignment: 0, 1/8, 1/4, 3/8, 5/8, 3/4
  - Illustration: good fit, bad readability

- **Grid 64th note**
  - Input: 0, 1/8, 1/4, 3/8, 5/8, 7/8, 1
  - Alignment: 0, 1/8, 1/4, 3/8, 5/8
  - Illustration: good fit, bad readability

*Steinway Grand Piano* on Track Steinway Grand Piano

**Region**
- **Time Quantize**
  - 1/16 Note

**Velocity**
- Value: 26
Hierarchical (irregular) Grids

regular grids
- search of a best quantization is possible by a brute-force enumeration:
  8th note grid, 16th, 32th, 64th...
- result not always optimal
- problems with tuplets (so called “irrationals” 3, 5, 7…)

hierarchical grids = trees
- more “natural” results
- brute force enumeration impossible

- prior specification of the language of acceptable hierarchical grids
  with quantitative formal language models:
  Weighted Tree Grammars

- Dynamic Programming parsing algorithm
  to extract 1-best tree, optimal wrt
  - weight wrt grammar (readability)
  - distance to input (accuracy)
Parse Trees & Vertical Alignments

Several events might be aligned at the same position of a hierarchical grid (= leaf of parse tree). They are considered simultaneous.

The FSM also detect (and fix) some MIDI captation errors. e.g. confusion between rim-shot and cross-stick.
Transcription Workflow

Input MIDI file (unstructured)

- k-best parsing
  - rhythm quantization
  - score structuring
    (hierarchical groups of notes, beams)

Parse Tree

Intermediate Score Representation (structured)

term rewriting

Voice separation

XML/MEI score file

Music 21 structure

Weighted Regular Tree Grammar

defines a prior rhythm notation language

evaluation of a cost of readability
Post-processing the intermediate score representation

Theoretical issues:
- rewrite strategies (e.g. IO or OI),
- conflicts (overlapping rules)…
with **Groove MIDI Dataset**

- by Google Magenta
  https://magenta.tensorflow.org/datasets/groove
- 13.6 hours, 1150 MIDI files ~ 22000 measures recorded by professional drummers on a electronic drum kit
- audio (wav) files synthesized from (and aligned to) MIDI files for evaluation of audio-to-MIDI drum transcription
- no score files!  
  *i.e.* no ground truth for evaluation

**Results**

- 25 score transcriptions to XML/MEI as proof of concept
  https://gitlab.inria.fr/transcription/gmdscores
- score length: 24 to 261 4/4 bars, tempo is known transcription time ~ 2s per score
- all score files (XML) produced from the MIDI files with the same generic tree grammar (4/4 measure)
- FSM for specific drumming constraints (hands ≤ 2, feet ≤ 2)
- processing of flams
- processing of errors from MIDI sensors
Experiments (comparison with MusScore)
Experiments (comparison with Steinberg’s Dorico)

qparse output

Dorico output

swing issue?
Transcription procedure for MIDI drumkit input, based on
- prior quantitative language models (weighted tree grammars)
- techniques of parsing by Dynamic Programing
- post-processing with Term Rewriting

Future plans:
• Complete transcription of Google’s **Groove MIDI Dataset**
  with manual corrections by a drummer
  → dataset of drum scores

• Backend of **Audio2MIDI** drum transcription

• **Guitar & Piano** MIDI2score transcription
  voice separation
  several piano datasets (aligned audio/MIDI/scores)
  based on recording from Yamaha Piano Competition