Real-time Detection of Finger Picking Musical Structures

Dale E. Parson

Consulting Member of Technical Staff, Agere Systems, dparson@ptd.net

banjo picker (35 years), computer scientist (29 years), electronic musician (1+ year)

The MIDIME Software Pipeline (Laboratory)

MIDI in from guitar or banjo ->

Stage 1

extracts timed state of guitar / banjo strings from input MIDI stream.

Stage 2

extracts meter, tempo, accents, scale(s), chords, drone(s), melody from stage 2.

Stage 3

matches a map of previous stage 2 scale-chord-drone tuples (+time+melody) to the current stage 2 scale-chord-drone tuple. This is a guess on where we are at and where we are headed in a composition.

Stage 4

consists of agents that read previous stages and generate MIDI accompaniment.

MIDI out to software or hardware synthesizer ->

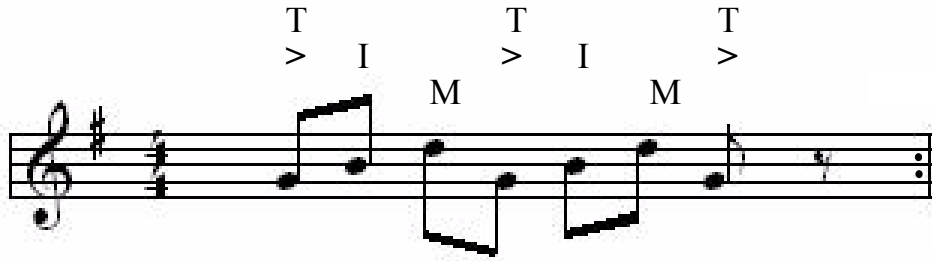
Stage 1: Map MIDI events to state of guitar / banjo strings

Detect **plucks** as explicit noteon events.
Convert flood of pitch bend messages to semitone crossing note events (**slurs**).
Interpret MIDI channel and guitar or banjo **string number**.
Superposition onto semitone pitch yields **3-to-1 to 5-to-1 reduction** in events.

```
0: bend,c=4,b=0x2068 2549
1: noteon,c=4,p=E4,v=122 2550
2: bend,c=4,b=0x2074 2569
3: bend,c=4,b=0x2088 2598
4: bend,c=4,b=0x2098 2619
5: bend,c=4,b=0x20a2 2729
6: bend,c=4,b=0x2092 2789
7: bend,c=4,b=0x2070 2820
8: bend,c=4,b=0x2042 2849
9: bend,c=4,b=0x22b8 2879
```

```
 0: string 4, stage1 row 0:
--- --- E4 --- --- ---
~0 ~0 ~8192 ~0 ~0 ~0
^0 ^0 ^122 ^0 ^0 ^0
<0 <0 <2550 <0 <0 <0
0> 0> 0> 0> 0> 0>
 1: string 4, stage1 row 1:
--- --- E4 --- --- ---
~0 ~0 ~8258 ~0 ~0 ~0
^0 ^0 ^122 ^0 ^0 ^0
<0 <0 <2550 <0 <0 <0
0> 0> 329> 0> 0> 0>
 2: string 4, stage1 row 2:
--- --- F4 --- --- ---
~0 ~0 ~8206 ~0 ~0 ~0
^0 ^0 ^122 ^0 ^0 ^0
<0 <0 <2879 <0 <0 <0
0> 0> 0> 0> 0> 0>
```

Stage 2, Part 1: Extract meter and tempo from pluck patterns



4/4 Finger pattern with accents at 1, 2.5, and 4

1	3	1	**3**	2	1	1	3	1	3	2	1	1	1	1	**1**	3	1	**1**	1	1	**1**	1	3	**binned raw**
8	5	5	**3**	0	4	9	4	4	3	0	4	7	7	9	**2**	3	9	**5**	7	7	**8**	4	3	**intervals**
6	8	3	**3**	1	8	4	8	4	7	3	6	8	9	3	**5**	9	1	**6**	7	3	**9**	1	3	
1	3	1	3	1	1	1	3	1	3	1	1	1	1	1	1	3	1	1	1	1	1	1	3	**average within**
7	4	4	4	9	4	9	4	4	4	9	4	7	7	9	4	4	9	7	7	7	9	4	4	**each bin**
5	1	3	1	5	3	5	1	3	1	5	3	5	5	5	3	1	5	5	5	5	5	3	1	
1	3	1	3	1	1	1	3	1	3	1	1	1	1	1	1	3	**1**	**1**	**1**	**1**	**1**	**1**	**3**	**snap bin to**
6	2	6	2	6	6	6	2	6	2	6	6	6	6	6	6	2	**6**	**6**	**6**	**6**	**6**	**6**	**2**	**2, 3, or 2 x 3**
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**play time division**
1	2	1	2	**1**	**1**	**1**	**2**	**1**	**2**	1	1	1	1	1	1	2	**1**	**1**	**1**	**1**	**1**	**1**	**2**	**normalize**

Stage 2, Part 2: Extract scale, chord and drones

Scale analysis uses repeated (non-transient) notes across last 4 measures.

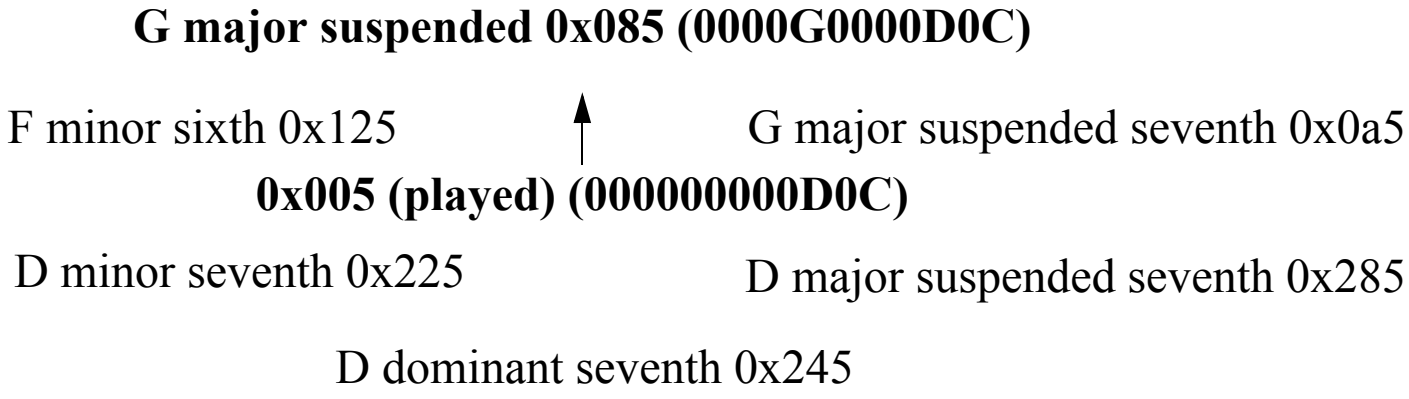
Chord analysis uses current notes sounding on strings, discarding low-velocity notes.

Drone (pedal point) analysis looks for open string plucks across last 4 measures.

Scale and drone analysis can use optional, top-down hints.

These analyses use octave-invariant bit vectors and Hamming distance maps.

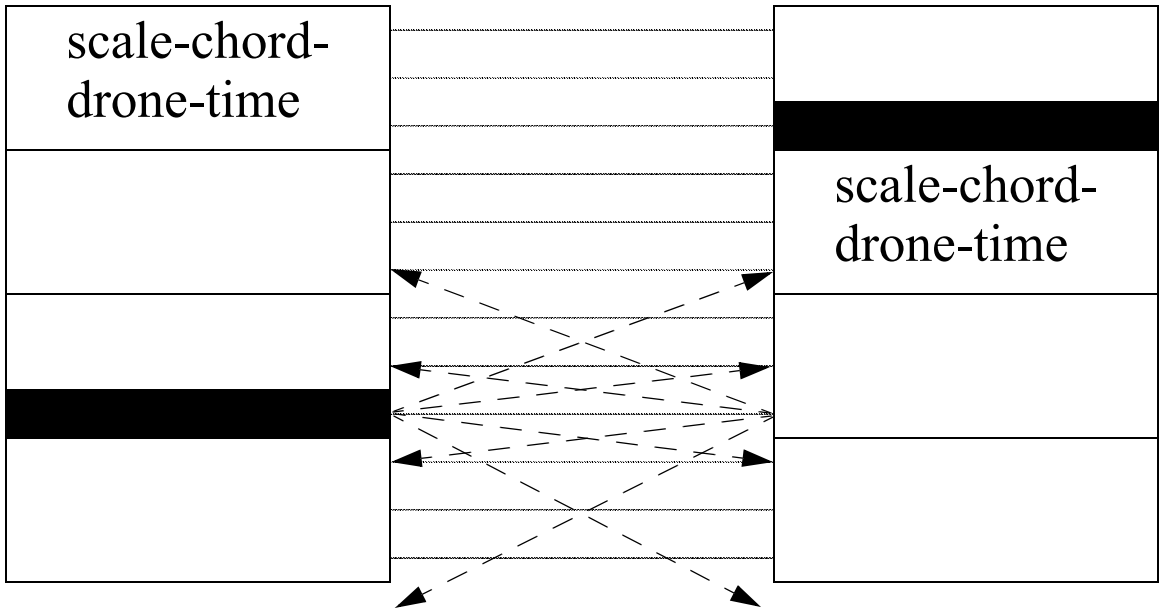
2-to-1 event reduction ratio from stage 1 due to discarding slurs.



Stage 3: Match current stage 2 state to Map of previous stage 2

Stage 3 map is a merger of scale-chord-drone-(time)-(melody) tuples from previous runs. Its use is to predict upcoming scale-chord-drones-melody across future time.

A performer builds a map by repeatedly playing the piece, capturing stage 2 traces, and then merging the traces. Merging uses the *Union Jack Algorithm* to remove transients.



Stage 3 Example: The Polarized Shuffle

Step 1: “Freeze dry” a chorus of *Seven of Eight* into a Stage 3 map.

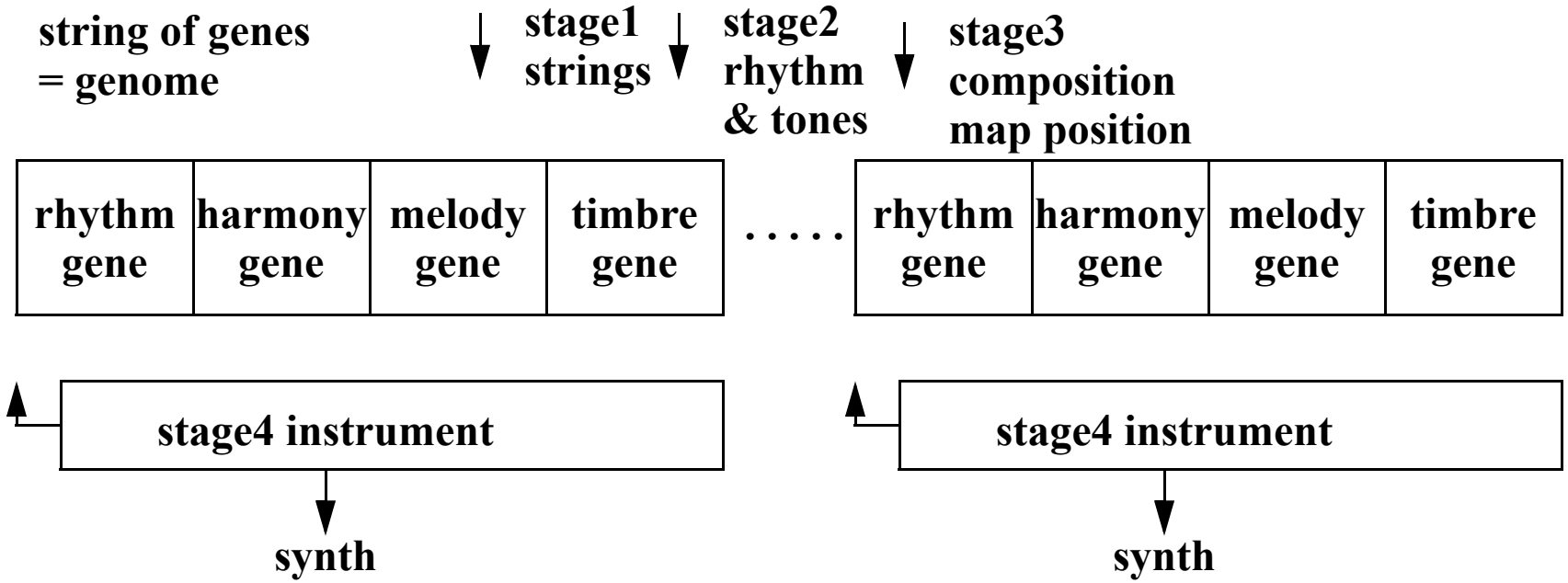
Step 2: “Rehydrate” this map by binding unbound variables within StringStudio.

The first attempt was like mixing blue, red & yellow pigments: mud.

The second attempt feeds chords, drones & melody notes into a 32-voice StringStudio arpeggiated harp configured for portamento. Meter and scale are not used.

The arpeggiator “swallows” consonant (“in phase”) melody notes.
The arpeggiator “spits out” dissonant (“out of phase”) melody notes that lie between chord notes in pitch and time, generating percussive noise and microtonal slides.

Stage 4 consists of genetically programmed agents.



Each Gene is a unit of typed code that matches input events & its state to output.

A Genome is a string of Genes (~32) that provide an Instrument Player's behavior.

Genetic operations of duplication, crossover, inversion & point mutation, steered by performer-driven selection, provide trials of performance space & convergence.

http://home.ptd.net/~dparson has additional information on MIDIME.