Musical Micro-Timing for Live Coding

Introducing metre and style-specific probabilistic micro-timing to Sonic Pi

Max Johnson[1] Mark Gotham[1,2]

[1] University of Cambridge, [2] Durham University

Introduction

Consider metre as a mental "grid" for events

Expressive micro-timing

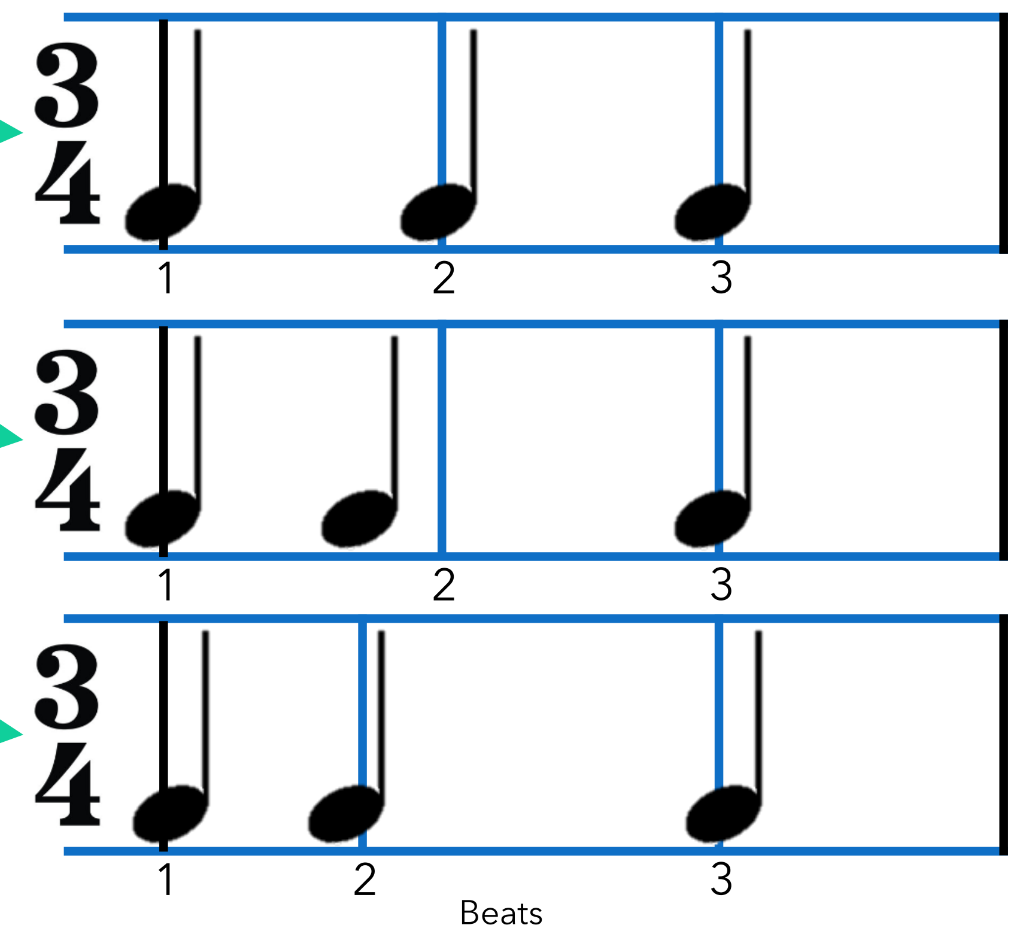
- Events expressively altered from categorical positions

Categorical micro-timing

- Uneven spaces between grid lines (the metre itself is uneven)

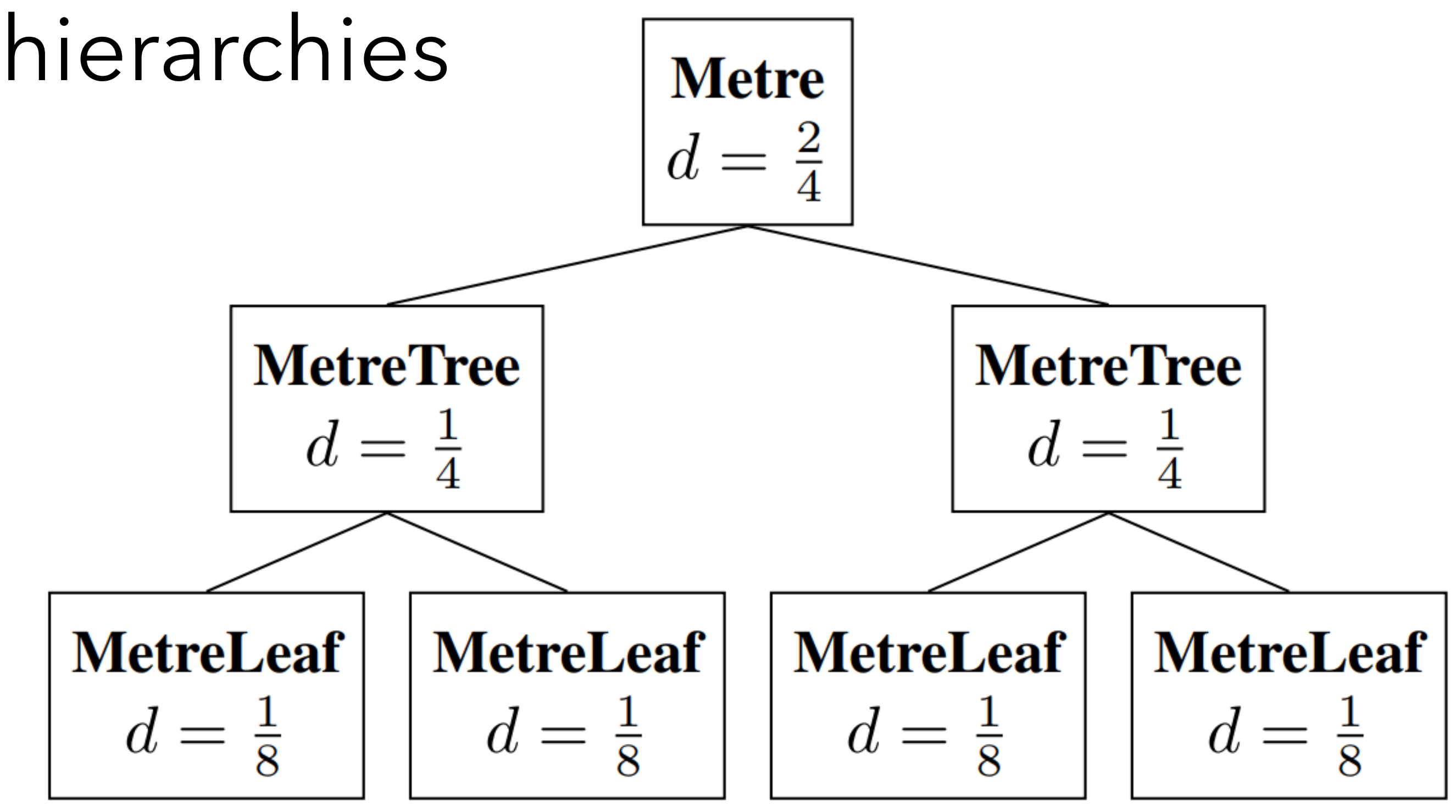
Sonic Pi live coding language

- How can we improve the "life-likeness" for specific styles? π)))

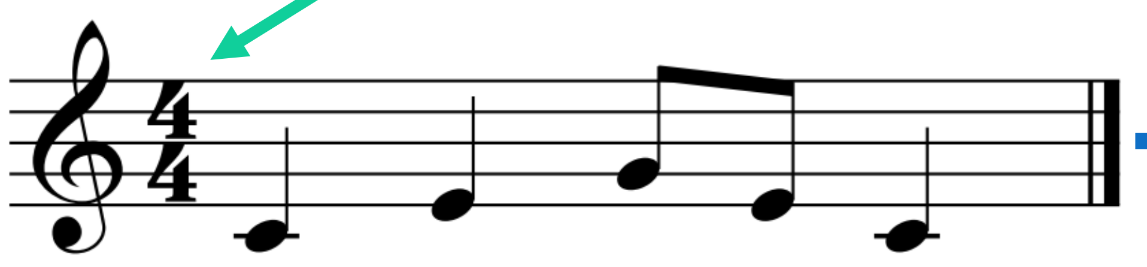


Metre

Nested objects allow for arbitrary metrical hierarchies



Retains information about the metre!



equivalent to

Western music notation

```
use_metre '4/4'
bar do
     add_note :C4, 0, 1
     add_note :E4, 0, 1
     add_note :G4, 1, 1
     add_note :E4, 1, 1
     add_note :C4, 0, 1
end
```

New Sonic Pi commands

(Specify note durations in terms of position in metrical hierarchy)

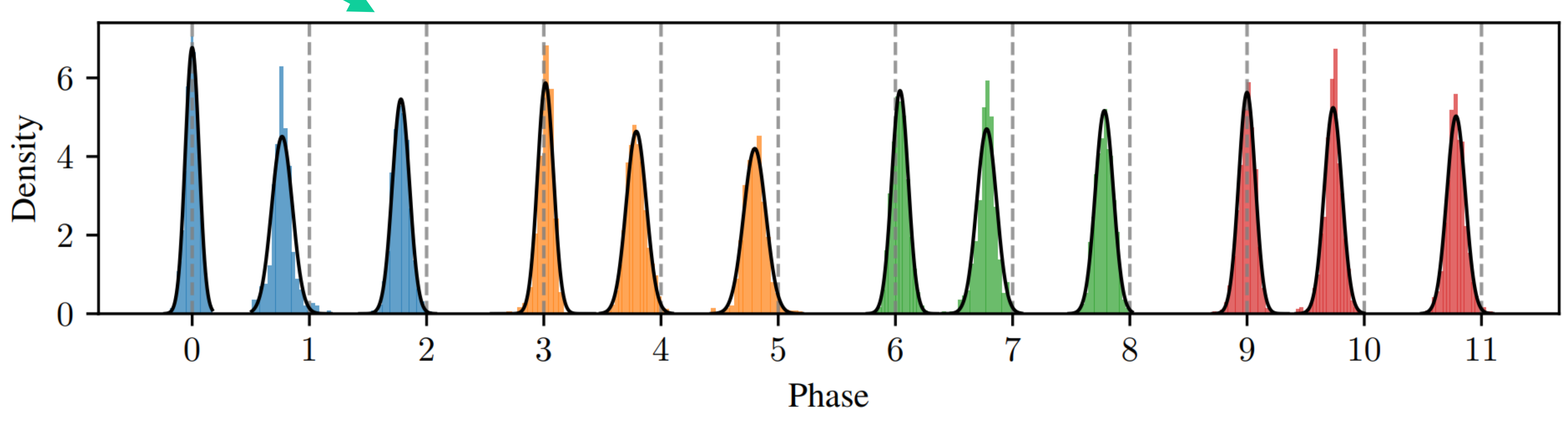
Micro-timing

Probabilistically alter the timing of each note according to its metrical location

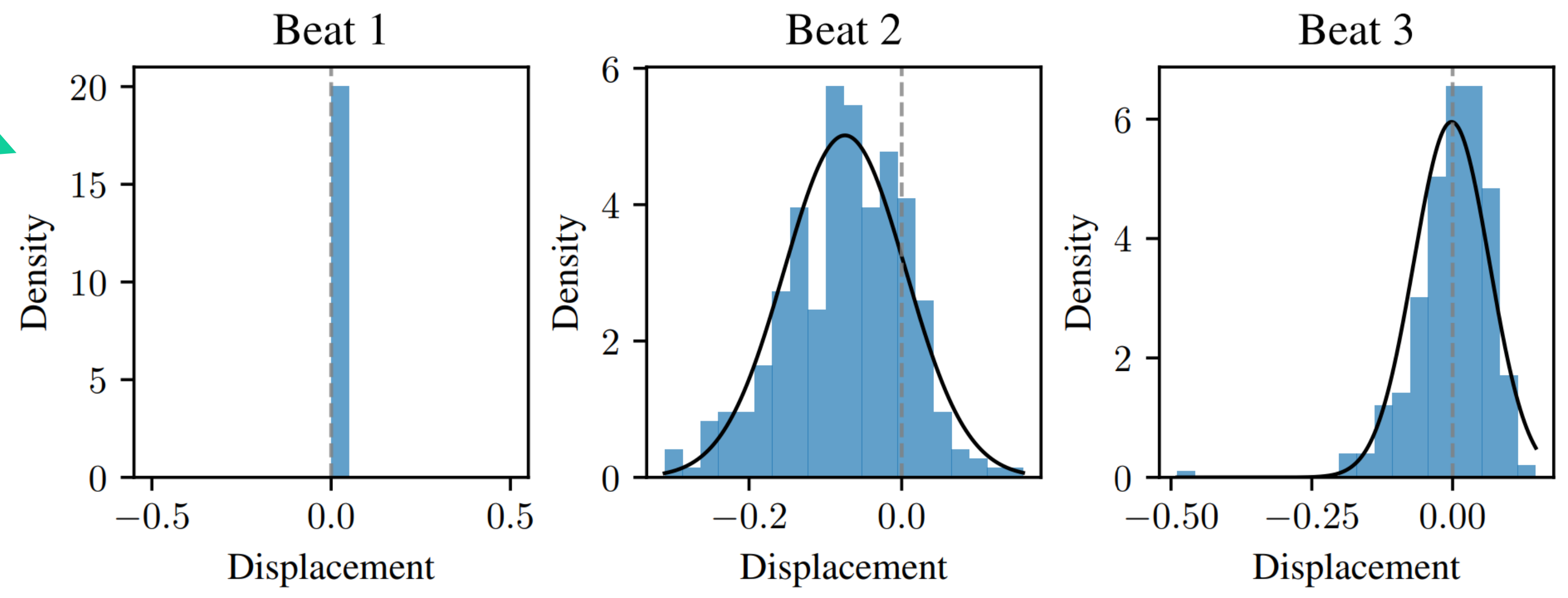
Analyse timings from performances relative to isochronous grid → Fit probability distribution with MLE (these encode the *actual*, non-isochronous metre) → Store a distribution for each event at each level → When playing a note, draw random samples and adjust timing accordingly

Results

Micro-timing data analysis results for two case study styles: Viennese waltz and Malian jembe



Shows a short-medium-long pattern. Black curves show the PDF of the probability distributions.



The second beat in each bar is slightly "early" (short-long-medium pattern).

Acknowledgements

- References and full details can be found in the paper.
- Thanks to Rainer Polak for his invaluable advice and feedback on this project.
- This research was conducted in the context of author MG's time as an Affiliated Lecturer at the Department of Computer Science and Technology, University of Cambridge. We thank Alan Blackwell and others for their role in organising and supporting this.

Source code