Computer Science Workshop (AINT101): Computer-Aided Composition (1)

Torsten Anders
Interdisciplinary Centre for Computer Music Research (ICCMR)
University of Plymouth
http://cmr.soc.plymouth.ac.uk/

20 November 2007

Warm-Up

Question

How can we use computers for making music?

Outline

What is Computer-Aided Composition (CAC)? I

Computer-Aided Composition (CAC)

User employs (and usually also writes) a computer program which outputs music: *score* synthesis, in contrast to *sound* synthesis. Alternative term: algorithmic composition.

Motivation

- Speeding up of formalisable compositional decisions
- Working on a more abstract level (e.g., changes to single algorithm argument may change whole composition)
- Composer may surprise herself

What is Computer-Aided Composition (CAC)? II

Note

Composers formalise musical intentions, and implement them as computer programs.

Ways of working

Automatic composition: the full composition process in formalised and implemented – approach used by scientists

Computer-aided composition: only subtasks formalised, other aspects composed 'by hand', and and output possibly hand-edited – approach used by artists

Intermediate approach: hand-selecting output from automatic composition

Existing Computer-Aided Composition Systems I

- General-purpose programming languages
  - Tendency towards very high-level languages.
  - Lingua franca in CAC: Lisp.
- Programming systems designed for music composition
  - Often extend general purpose programming language by music-specific constructs
  - Users with less technical background often prefer visual languages

Existing Computer-Aided Composition Systems II

Widely used CAC systems

System	Programming language	Special features
OpenMusic, PWGL	Visual language and Common Lisp	
Common Music	Common Lisp	
SuperCollider	Own language (Smalltalk-like semantics, C-like syntax)	Sound synthesis
Max/MSP, PD	Visual language	Sound synthesis

Existing Computer-Aided Composition Systems III

System links

- OpenMusic: `http://recherche.ircam.fr/equipes/repmus/OpenMusic/`
- PWGL: `http://www2.siba.fi/PWGL/`
- Common Music: `http://commonmusic.sourceforge.net/doc/cm.html`
- SuperCollider: `http://supercollider.sourceforge.net/`
- Max/MSP: `http://www.cycling74.com/products/maxmsp`
- PD: `http://puredata.info/`

Existing Computer-Aided Composition Systems IV

- Many more systems exist: `http://www.flexatone.net/algoNet/` lists 115 composition systems!
- We will focus on concepts and techniques, not on systems.

History: Musikalisches Würfelspiel

- Composer creates table with several alternative solutions per bar
- User composes a minuet by throwing the dice to select a solution per bar

Example

Wolfgang Amadeus Mozart (1756 – 1791): *Musikalisches Würfelspiel*

`http://sunsite.univie.ac.at/Mozart/dice/`

`http://sunsite.univie.ac.at/Mozart/dice/rules.html`

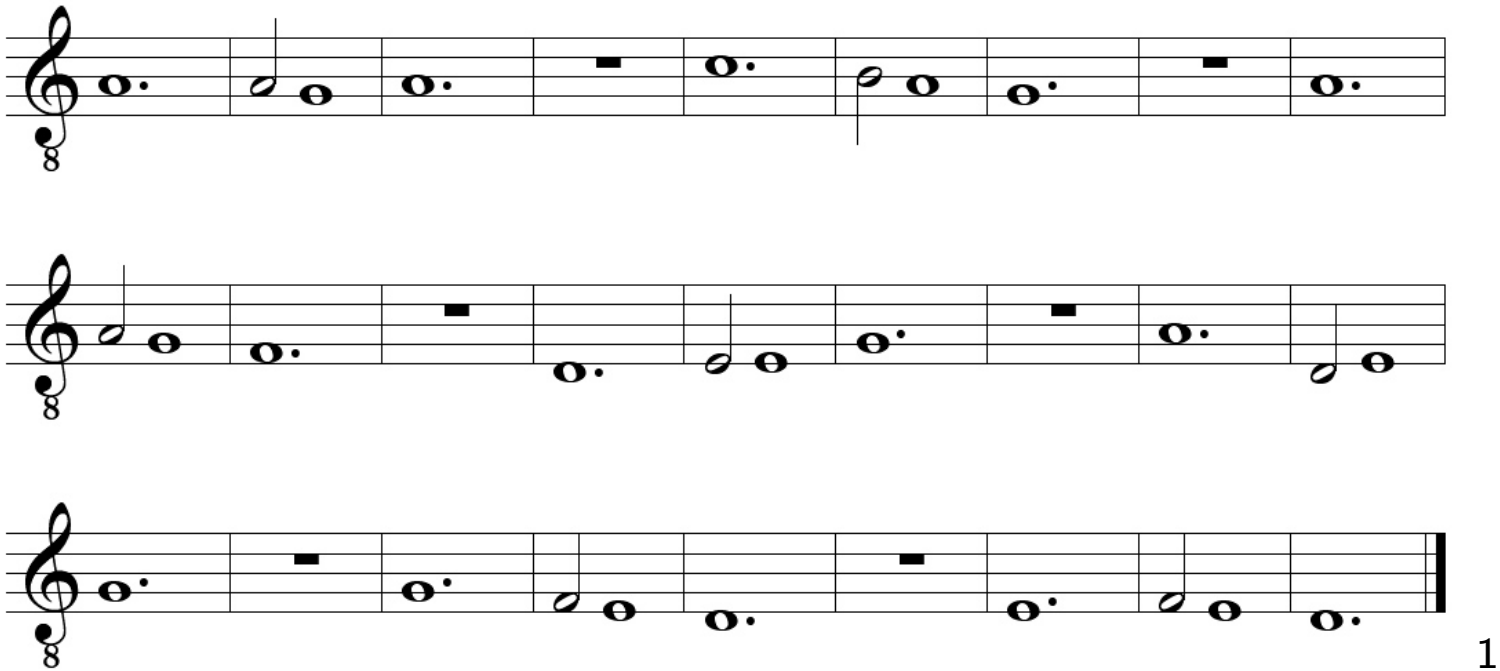
History: Isorhythmic Music I

- Organisation of rhythmic structure with: cyclic repetition of a note duration sequence
- Repeated note duration sequence called *talea*
- Similar organisation of pitches possible (pitch sequence called *color*)

Example

Guillaume de Machaut (ca. 1300 – 1377): *Messe de Nostre Dame, Kyrie*

History: Isorhythmic Music II



[1]

Figure: Isorhythmic tenor from the Kyrie of Machaut's Messe de Nostre Dame

[1] Source: Wikipedia, *Isorhythm*.

Music Representation: Motivation

Note

When using the computer for composition, music must be represented in a way suitable for computer programs.

Question

Any ideas how we can represent music for use by computer programs?

Music Representation: Parameterisation of Music I

- Musical events (e.g., a note) are represented by a set of *parameters* (e.g., start time, duration, pitch, loudness)
- Parameters can be stored and processed indepedently in a computer
- Example: a note can be represented as a *composite data structure*

Music Representation: Parameterisation of Music II

- Note: such a music representation allows also for new musical results

Example (realised mechanically, i.e. pre-computer!)

Conlon Nancarrow (1912 – 1997): *Player Piano Study #21, Canon – X*

Task

Listen closely: what is going on here musically?

Event-List: a Basic Representation I

Definitions

An *event* is score object which produces sound. An event features a set of parameters.
A *parameter* is a basic magnitude in a music representation.
Examples: start time, duration and pitch.

Event-list example

```
[note(start:0, dur:1, pitch:60, amp:0.5),
 note(start:1, dur:1, pitch:64, amp:0.5),
 note(start:2, dur:1, pitch:67, amp:0.5)]
```

Event-List: a Basic Representation II

- Event-lists successfully applied in sound synthesis systems for decades
- Music representation is a research field in its own – we will leave it now

Fractals Introduction I

- Music often contains dense net of similarity relations (e.g., motivic relations)
- A *fractal* is a geometric shape which is *self-similar*
- Composers are therefore interested in applying fractals in a CAC process

Fractals Introduction II

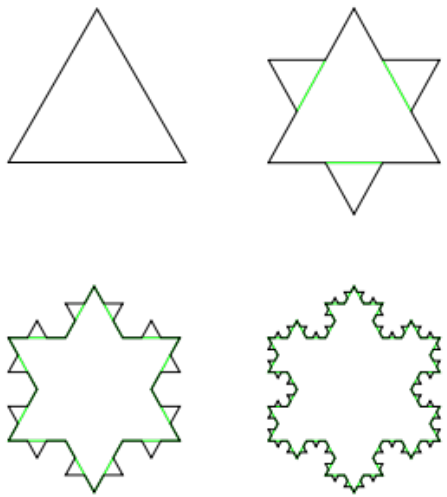
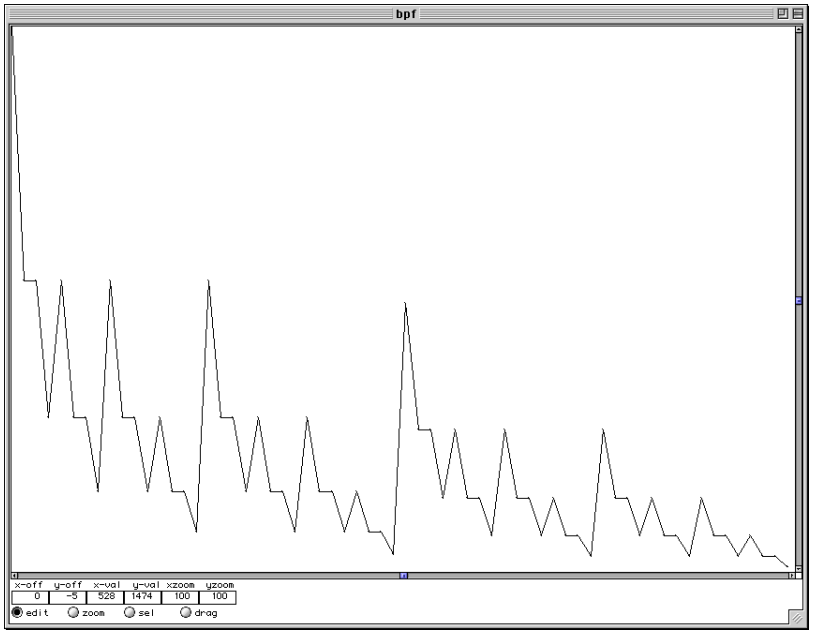


Figure: The first four iterations of the *Koch snowflake* fractal[2]

[2]Source: Wikipedia, *Koch snowflake*.

Creating Music with Fractal Data I

- Basic raw material: a number sequence created with a fractal, similar to the Koch curve
- This number sequence is then *mapped* to musical parameters (see below)



Creating Music with Fractal Data II

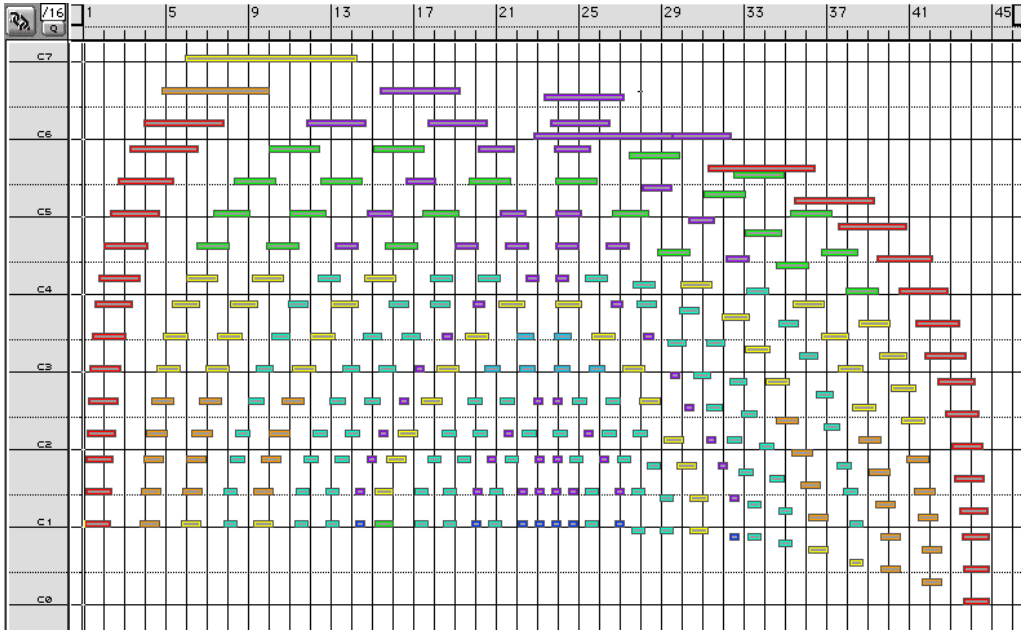


Figure: Torsten Anders: *Kunststoff*, 5th movement, in piano-roll notation

Creating Music with Fractal Data III

- Music consists of many layers (voices, with initially always repeat their pitch)
- Fractal data is almost directly mapped to rhythmic structure and the dynamics of each layer, only simple transformations occur
- Lowest layer uses the full fractal data sequence extended by its reverse (forming a palindrome)
- Upper layer always use a value less of the fractal data sequence, their start time is delayed, and all durations are longer

Cellular Automata: the Concept I

- Cellular automata are a very simple formalism with can show surprisingly complex and interesting behaviour
- This model has a time dimension, which makes it even more suitable for musical applications

Example: Game of Life

`http://en.wikipedia.org/wiki/Conway%27s_Game_of_Life`

Cellular Automata: the Concept II

A discrete model: discrete cells and discrete time

- A regular grid of cells (one-dimensional case: a sequence of cells)
- Each cell has finite number of states (e.g., two states: 0 and 1)
- For each cell, a rule specifies the state of the cell in the next generation – depending on the current state of the cell and its neighbours

Cellular Automata: the Concept III

One-dimensional cellular automaton example: Rule 30 cellular automaton

- First table line shows the state of three cell neighbours at current stage
- Second line shows the state of the center cell at next stage

111	110	101	100	011	010	001	000
0	0	0	1	1	1	1	0

- Rule name follows so-called Wolfram notation: sequence of new states read as binary number

Cellular Automata: the Concept IV

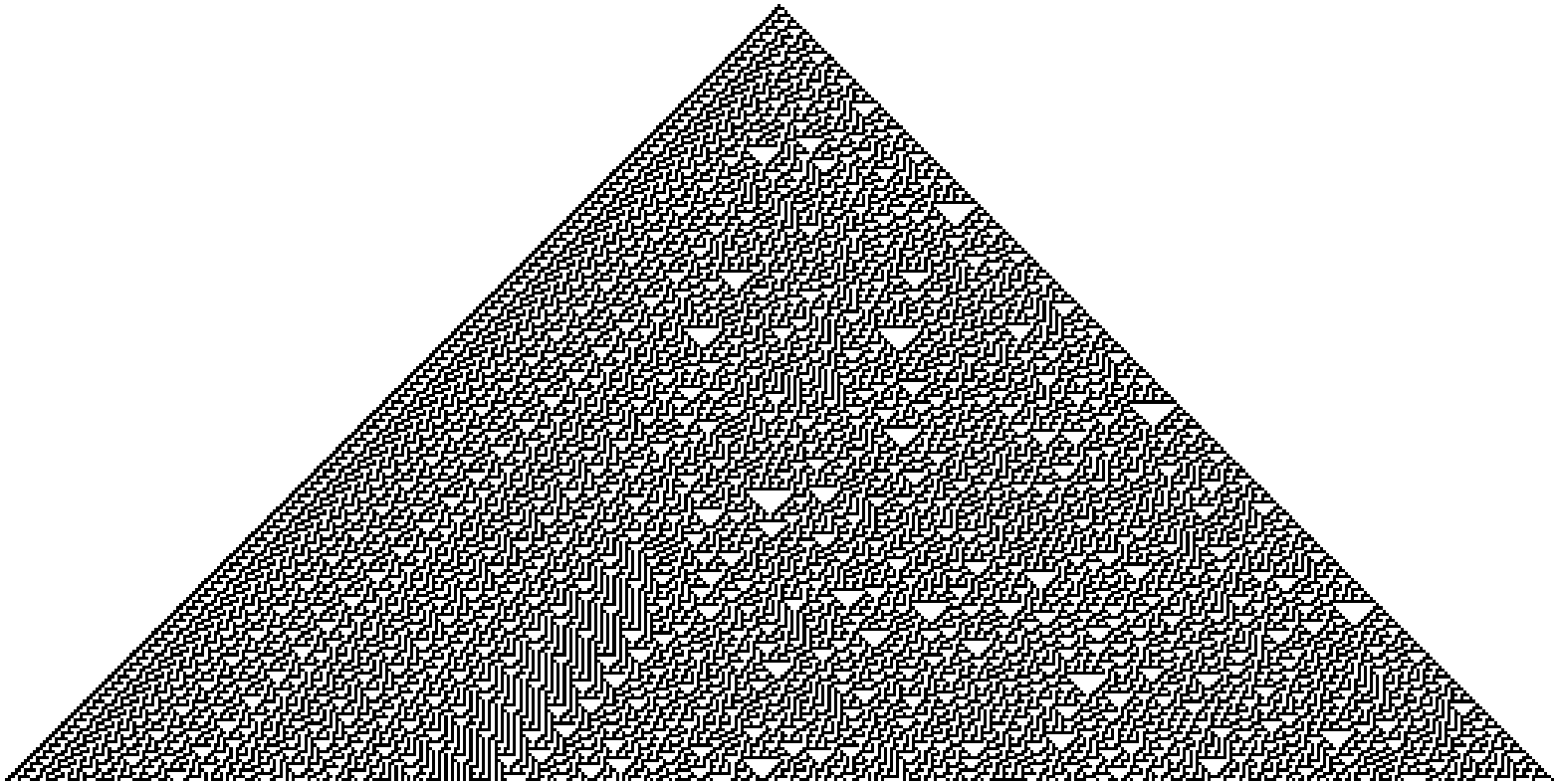


Figure: Rule 30 cellular automaton result[3]

[3] Source: Wikipedia, *Cellular Automata*.

Creating Music with Cellular Automata I

Example: Cellular Automata: Wolfram Tones:
`http://tones.wolfram.com/`

Recommended Literature

- Miranda, E. R (2001). *Composing Music with Computers*. Focal Press.
- Roads, C. (1996). *The Computer Music Tutorial*, Chap. 18 "Algorithmic Composition Systems" and 19 "Representation and Strategies for Algorithmic Composition". MIT press.
- Dodge, C. and Jerse, T. A. (1997). *Computer Music: Synthesis, Composition, and Performance*. Schirmer Books.

Summary

- Introduction: computer-aided composition
- Historical examples: Würfelspiel and isorhythmic music
- Music representation: event-list
- CAC techniques: fractals and cellular automata