A framework for dynamic spatial acoustic scene generation with Ambisonics in low delay realtime

Giso Grimm[a,b], Tobias Herzke[c]

[a] Medical physics institute, Universität Oldenburg, Germany

[b] ORLANDO*viols* – viol consort

[c] HörTech gGmbH, Oldenburg, Germany

LAC2012, April 13th 2012

Outline

Introduction

Venetian polychoral style

Introduction

Venetian polychoral style

- Spatially seperated sources
- Enveloping sound

Introduction

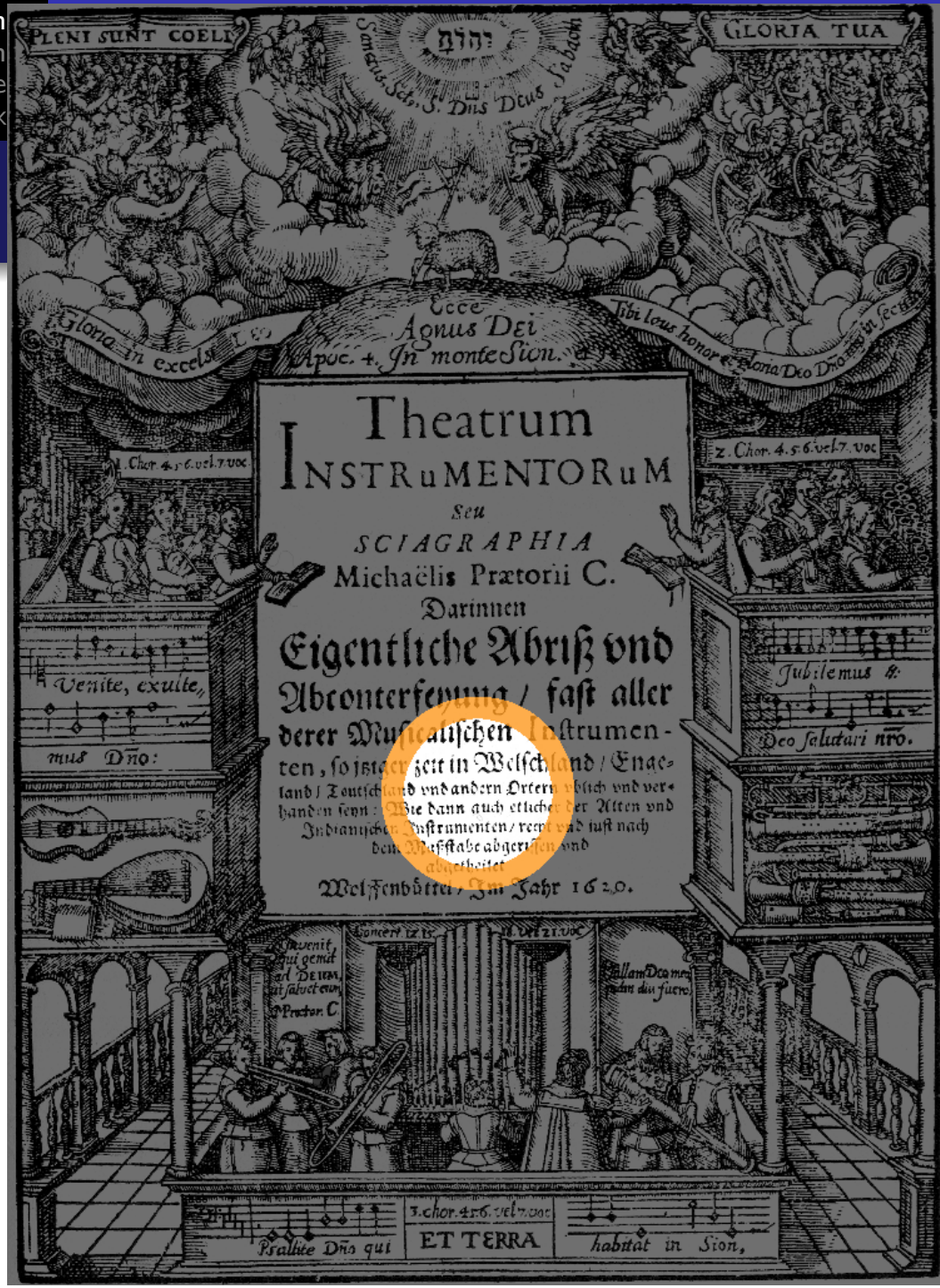
Venetian polychoral style

- Spatially seperated sources
- Enveloping sound
- PLL clock synchronization

Introduction

Venetian polychoral style

- Spatially seperated sources
- Enveloping sound
- PLL clock synchronization
- Small sweetspot, depending on musical complexity



Introduction

Venetian polychoral style

- Spatially seperated sources
- Enveloping sound
- PLL clock synchronization
- Small sweetspot, depending on musical complexity

⇒ We want more!

Toolbox for acoustic scene creation and rendering

- New toolbox: TASCAR
  Toolbox for Acoustic Scene Creation And Rendering
- Open source (GPL/LGPL), http://linuxaudio.vegri.net/
- Inspired by existing tools:
  - SoundScapeRenderer (jack, TU Berlin)
  - AMB plugins (ladspa, Fons & Jörn)
- Focus: Dynamic spatial content creation
  - Initially developed for music performance
  - Extended within research project

Hearing Research

Research targets

- Spatial psychoacoustics, e.g.
  - Precedence effect of moving sources
  - Localization, Source tracking
- Speech perception, e.g.
  - Speech intelligibility in complex scenes
  - Effect of head- and torso movements
- Hearing aids, e.g.
  - Directional microphone algorithm evaluation
  - Gesture recognition for hearing aid control

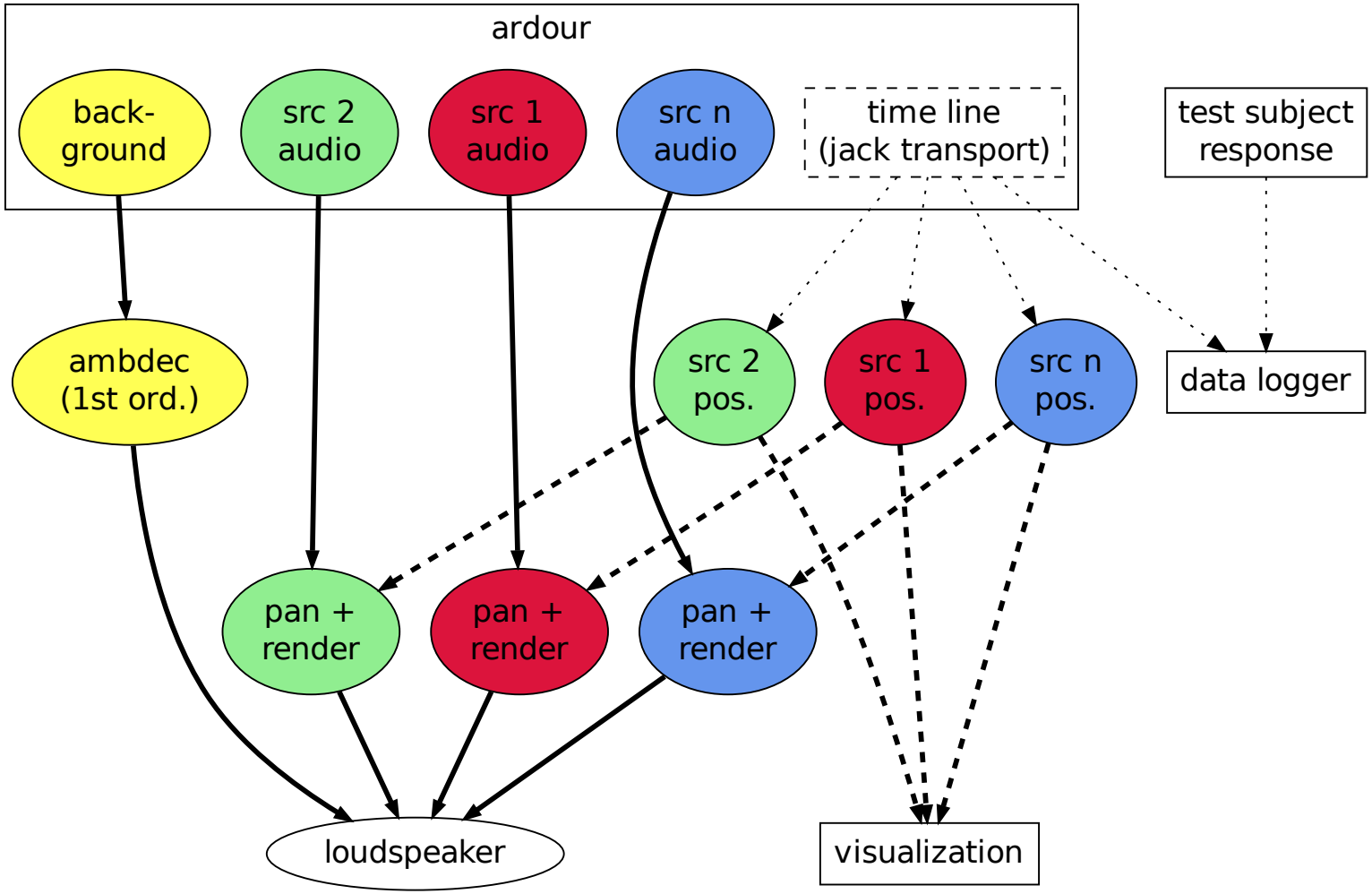
What we would like to have:

- Realistic acoustic scenes with high spatial complexity (e.g., 'cafeteria problem', public transort)
- Physically correct sound field within a certain sweetspot (human and machine listening)
- Spatial dynamics of sources are controllable
- Contents exchangable independently from spatial configuration

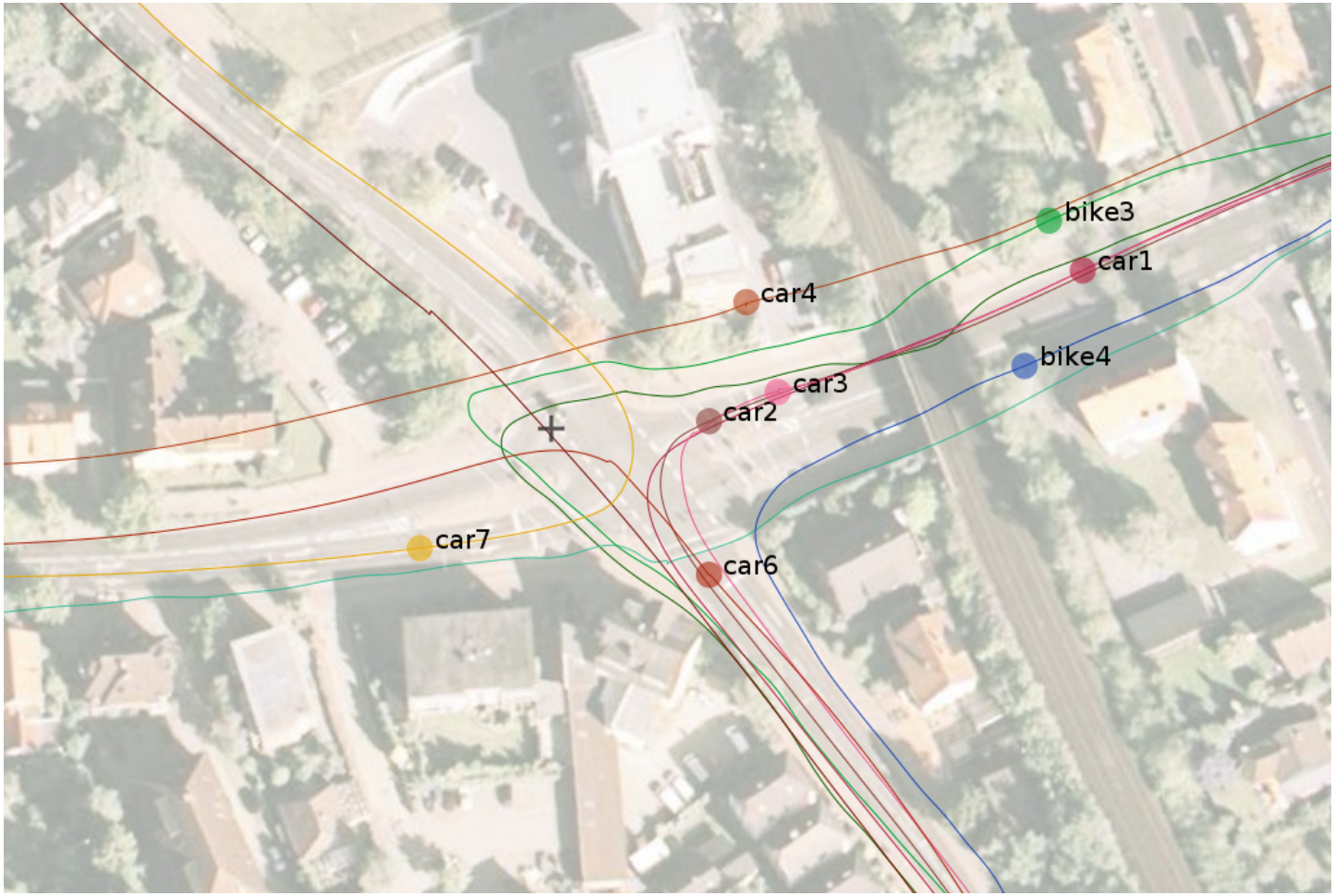
Acoustic scene creation and rendering

- Create tracks (csv format: t, x, y, z): Record sound and GPS track, project to local coordinates
  → `tascar_gpx2csv`
- Play position as jack ports, use jack transport
  → `tascar_jpos`
- Panning module takes position and audio as jack ports
- Render to speaker given layout in several formats (`nearest`, `vbap`, `amb_basic`, `amb_inphase`, `wfs`, `cardiod`)
  → `tascar_multipan`
- Add ambient background noise (first order ambisonics), use ambdec for rendering to speaker layout

Setup

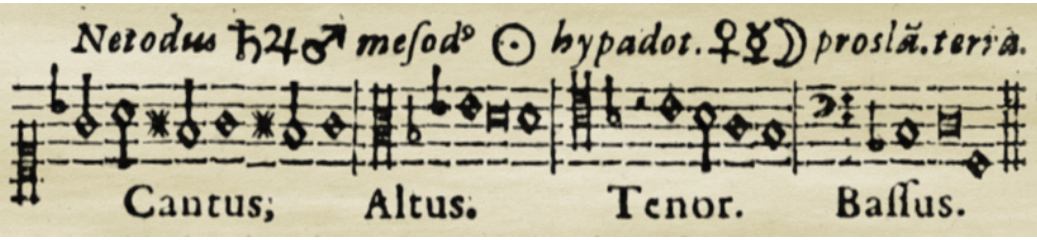

ardour
back-ground
src 2 audio
src 1 audio
src n audio
time line (jack transport)
test subject response
ambdec (1st ord.)
src 2 pos.
src 1 pos.
src n pos.
data logger
pan + render
pan + render
pan + render
loudspeaker
visualization

Example scene: Outdoor (street) situation

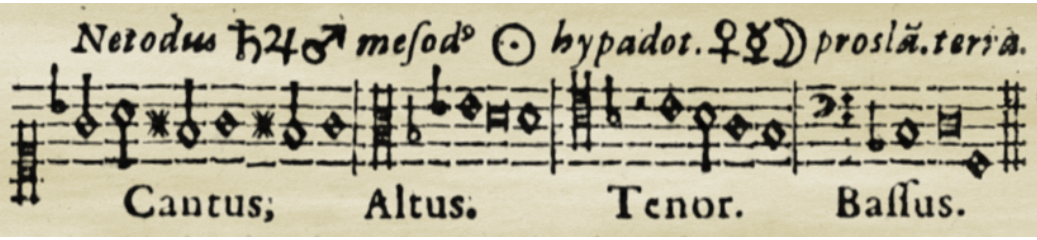


Music performance

Harmony of the Spheres – concert

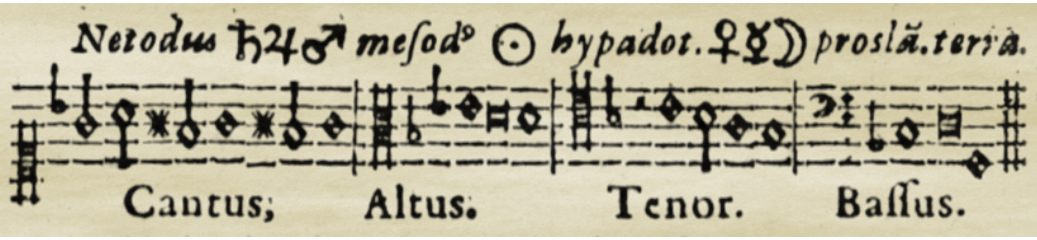


Harmony of the Spheres – concert



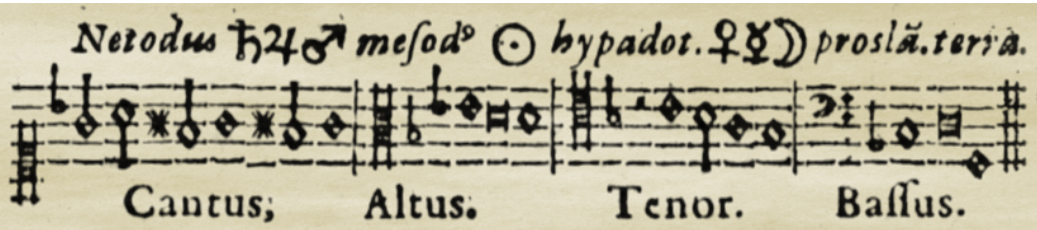
- Harmony of the Spheres: Philosophical concept of projecting celestial relationships (renaissance)

Harmony of the Spheres – concert



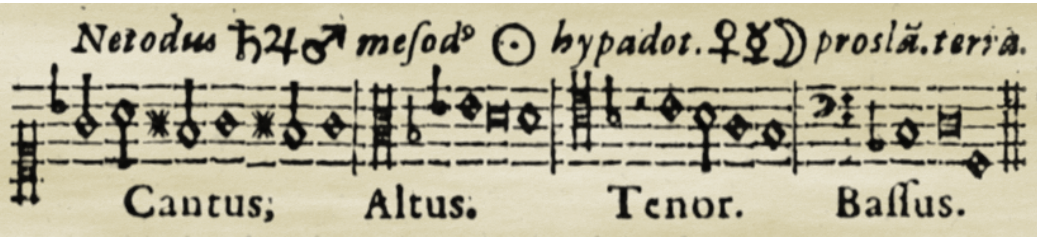
- Harmony of the Spheres: Philosophical concept of projecting celestial relationships (renaissance)
- Music theory devised from celestial motions (e.g., Bartolus 1614, Kepler 1619, Kircher 1650)

Harmony of the Spheres – concert



- Harmony of the Spheres: Philosophical concept of projecting celestial relationships (renaissance)
- Music theory devised from celestial motions (e.g., Bartolus 1614, Kepler 1619, Kircher 1650)
- Number aesthetics and symmetry affected compositions (musical form, number of voices, rules of algorithmic composition)

Harmony of the Spheres – concert



- Harmony of the Spheres: Philosophical concept of projecting celestial relationships (renaissance)
- Music theory devised from celestial motions (e.g., Bartolus 1614, Kepler 1619, Kircher 1650)
- Number aesthetics and symmetry affected compositions (musical form, number of voices, rules of algorithmic composition)
- Subtle deviation from regularities create musical content in minimal music

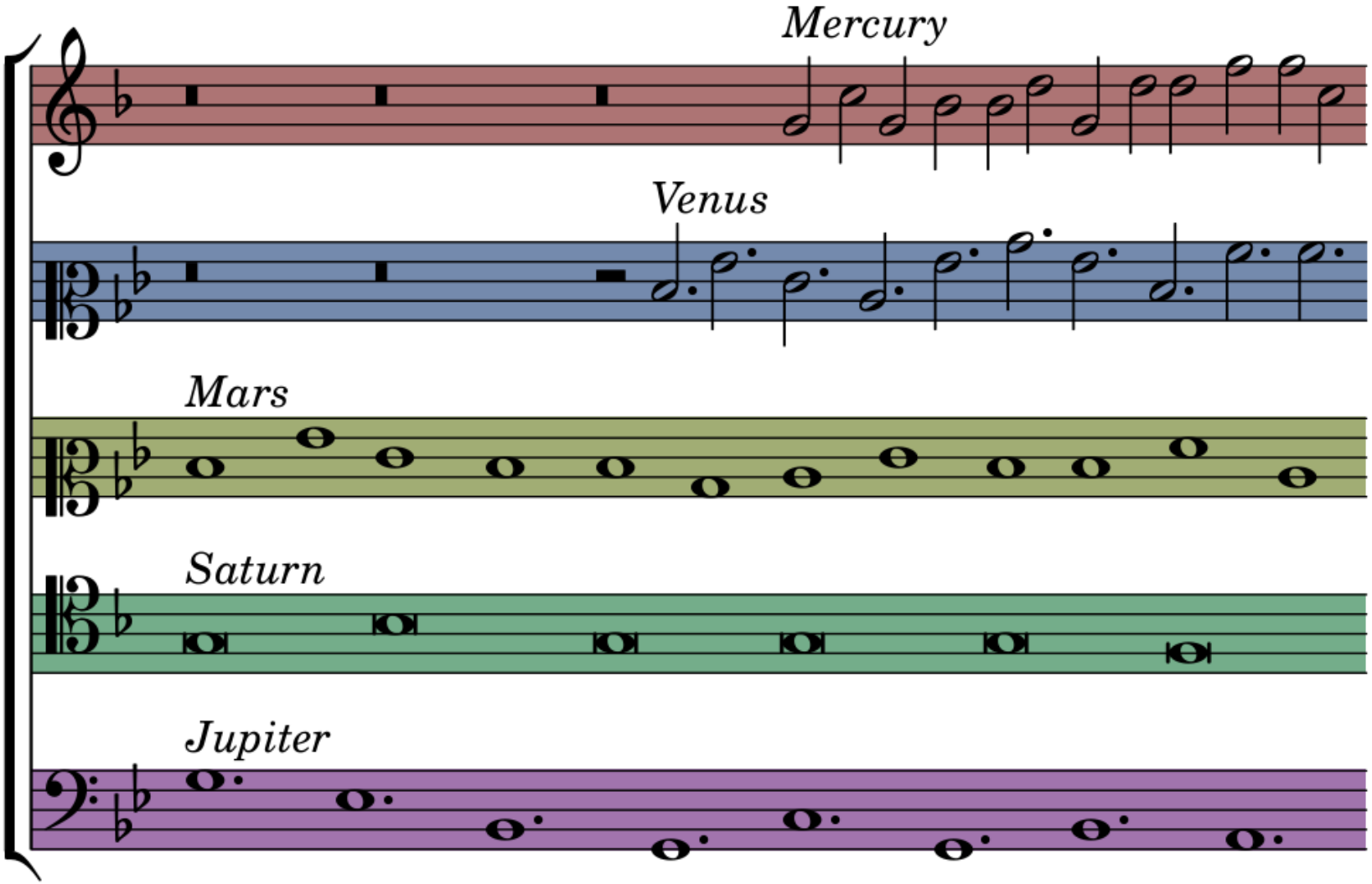
Harmony of the Spheres – spatialization

- Create virtual sources from acoustic inputs
- Let them move around the audience

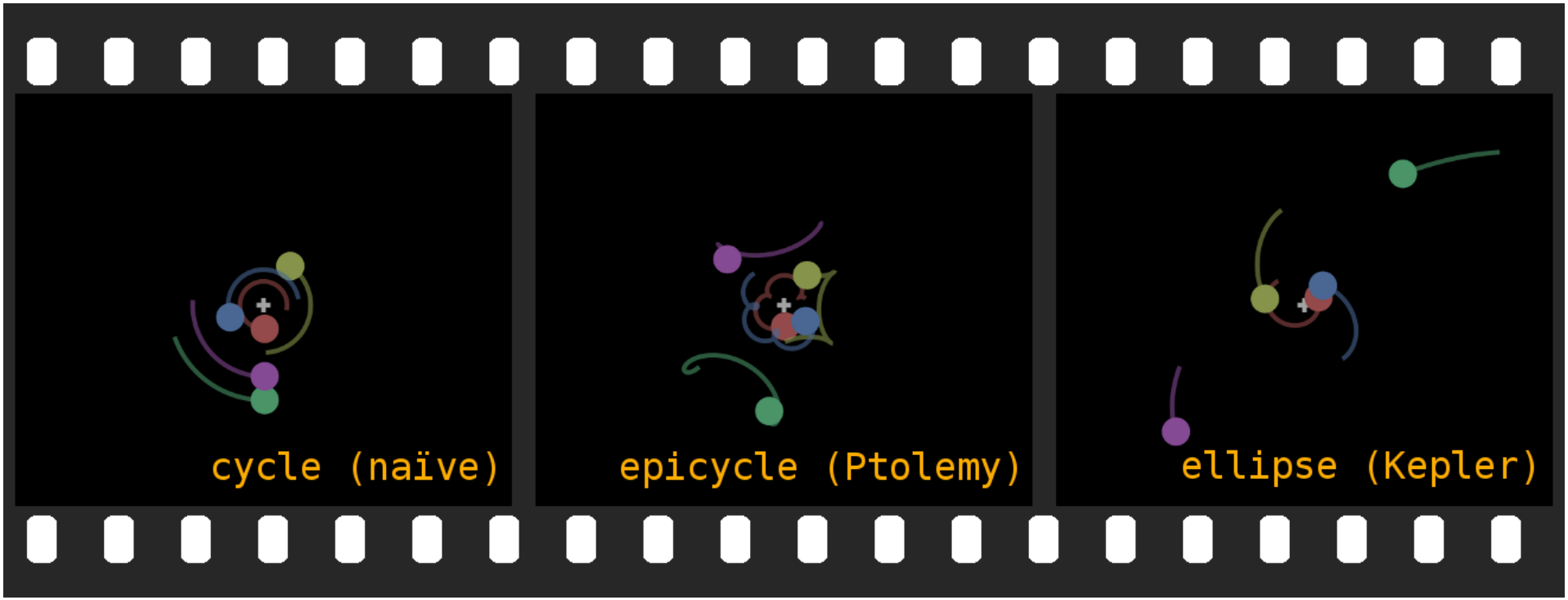
Harmony of the Spheres – spatialization

- Mr. Picforth: *In Nomine (16. century)*
  Perfect proportion, voices representing planets

Harmony of the Spheres – spatialization



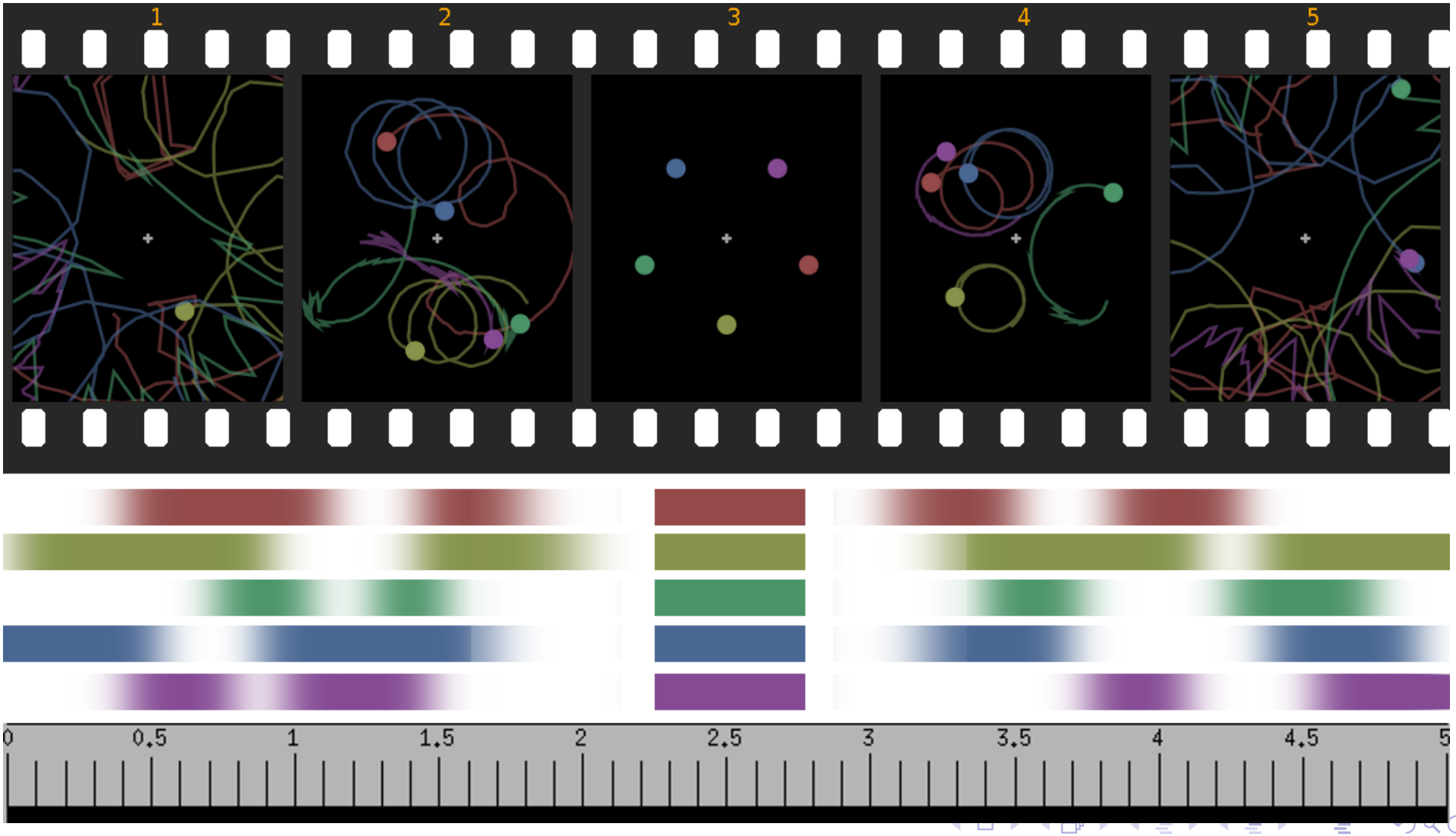
Harmony of the Spheres – spatialization



Harmony of the Spheres – spatialization

- Mr. Picforth: *In Nomine (16. century)*
  Perfect proportion, voices representing planets
- John Cage: *Five (1988)*
  Transition from disorder to harmony and back to disorder

Harmony of the Spheres – spatialization



Harmony of the Spheres – spatialization

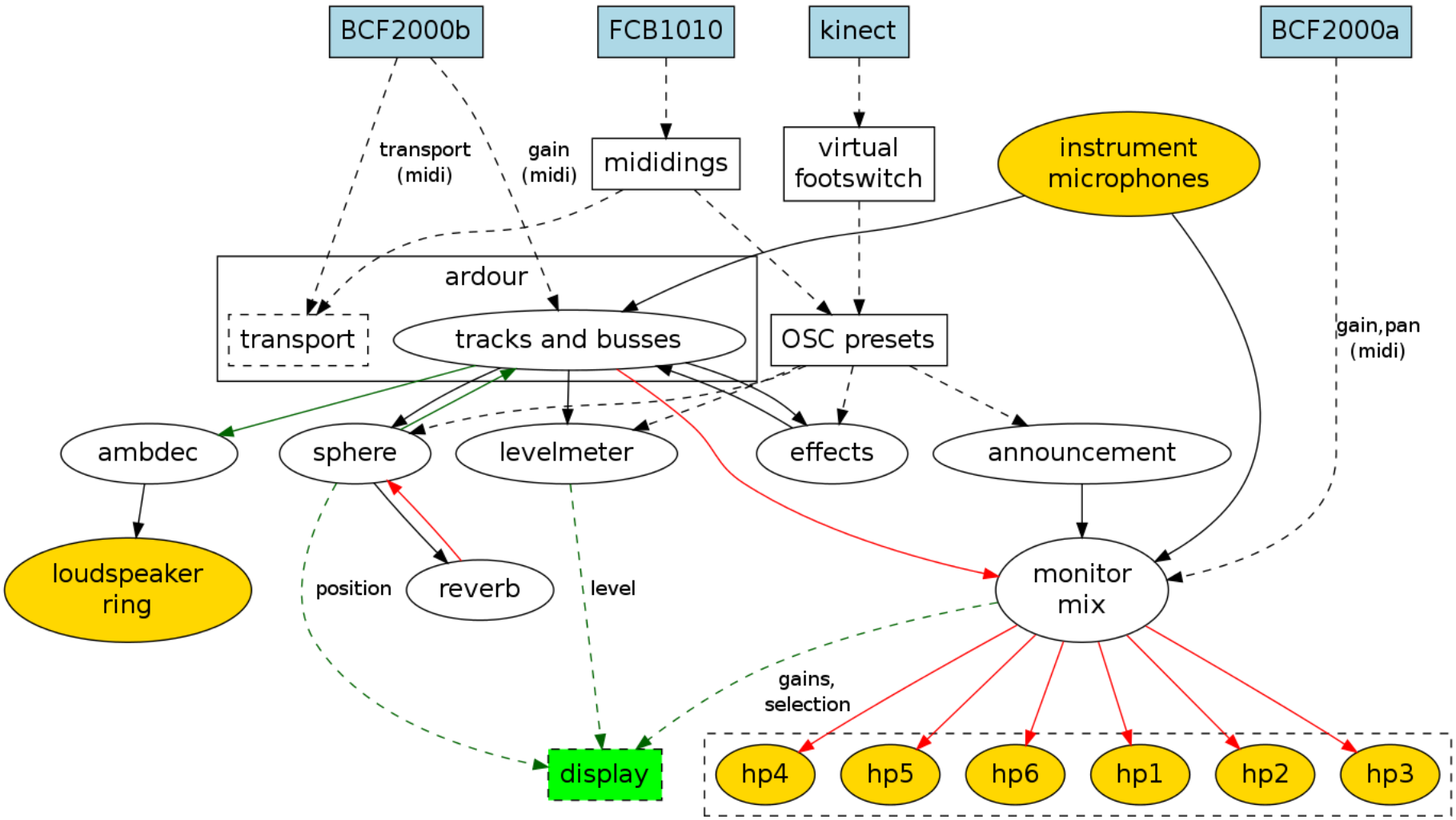
- Mr. Picforth: *In Nomine (16. century)*
  Perfect proportion, voices representing planets
- John Cage: *Five (1988)*
  Transition from disorder to harmony and back to disorder
- Lee Santana: *Music for Viols (1988)*
  Late minimal music, 'choreography' amplifies composer's intentions

Harmony of the Spheres – spatialization

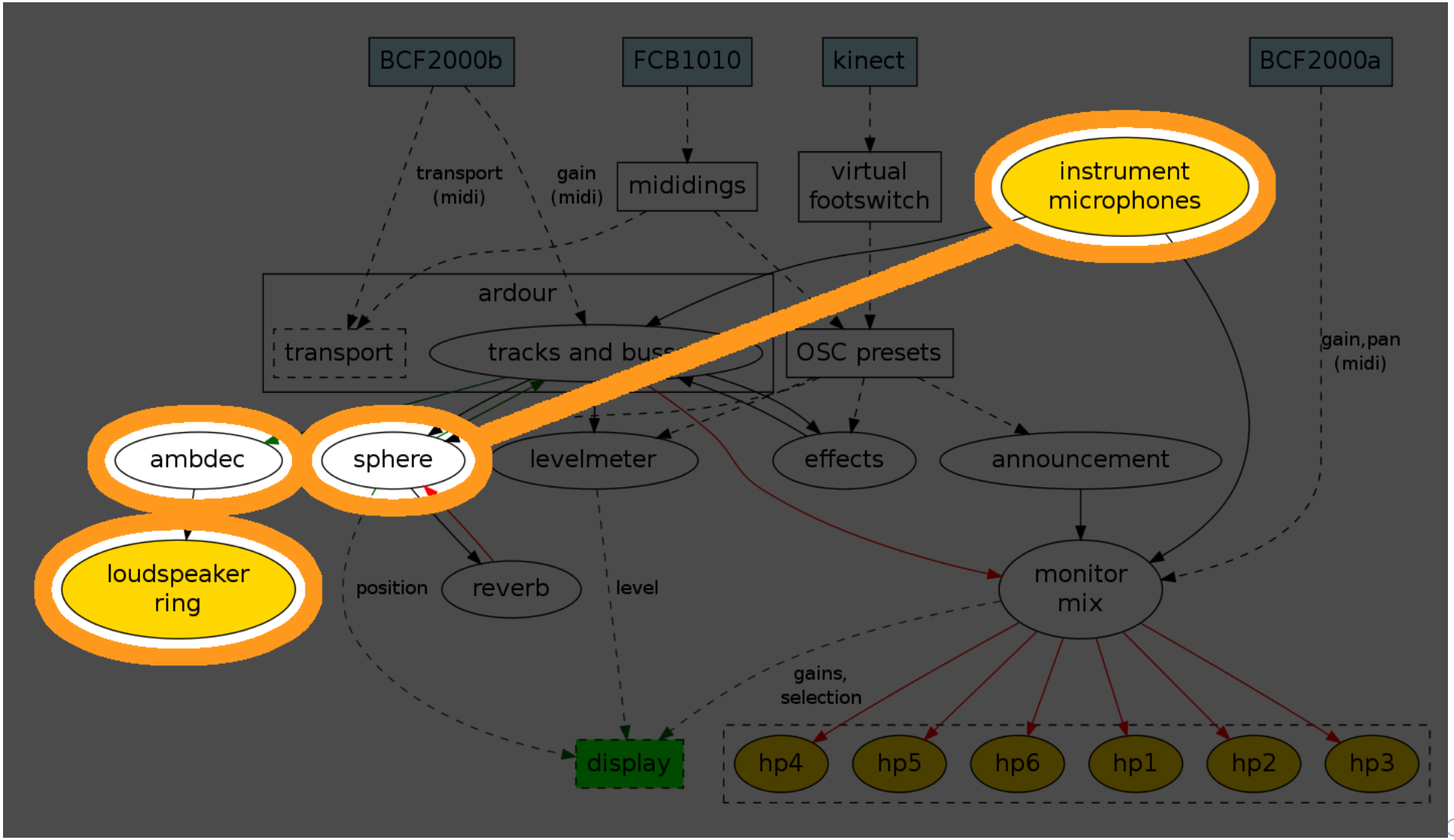
- Mr. Picforth: *In Nomine (16. century)*
  Perfect proportion, voices representing planets
- John Cage: *Five (1988)*
  Transition from disorder to harmony and back to disorder
- Lee Santana: *Music for Viols (1988)*
  Late minimal music, 'choreography' amplifies composer's intentions

Christopher Tye (c. 1505-1572), Giovanni Pierluigi Palestrina (1525-1594), Nicholas Strogers (†1575), Arvo Pärt (*1935), Steve Reich (*1936)

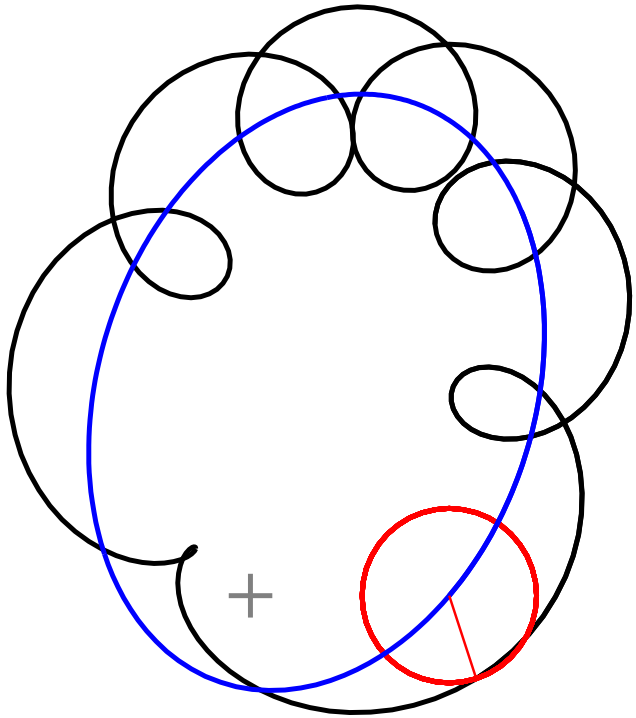
Harmony of the Spheres – signal flow



Harmony of the Spheres – signal flow



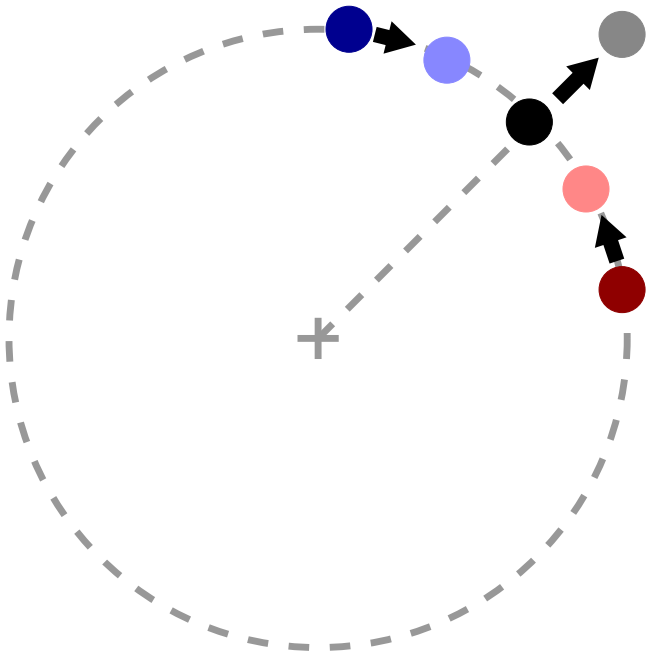
hos_sphere: Trajectory generator



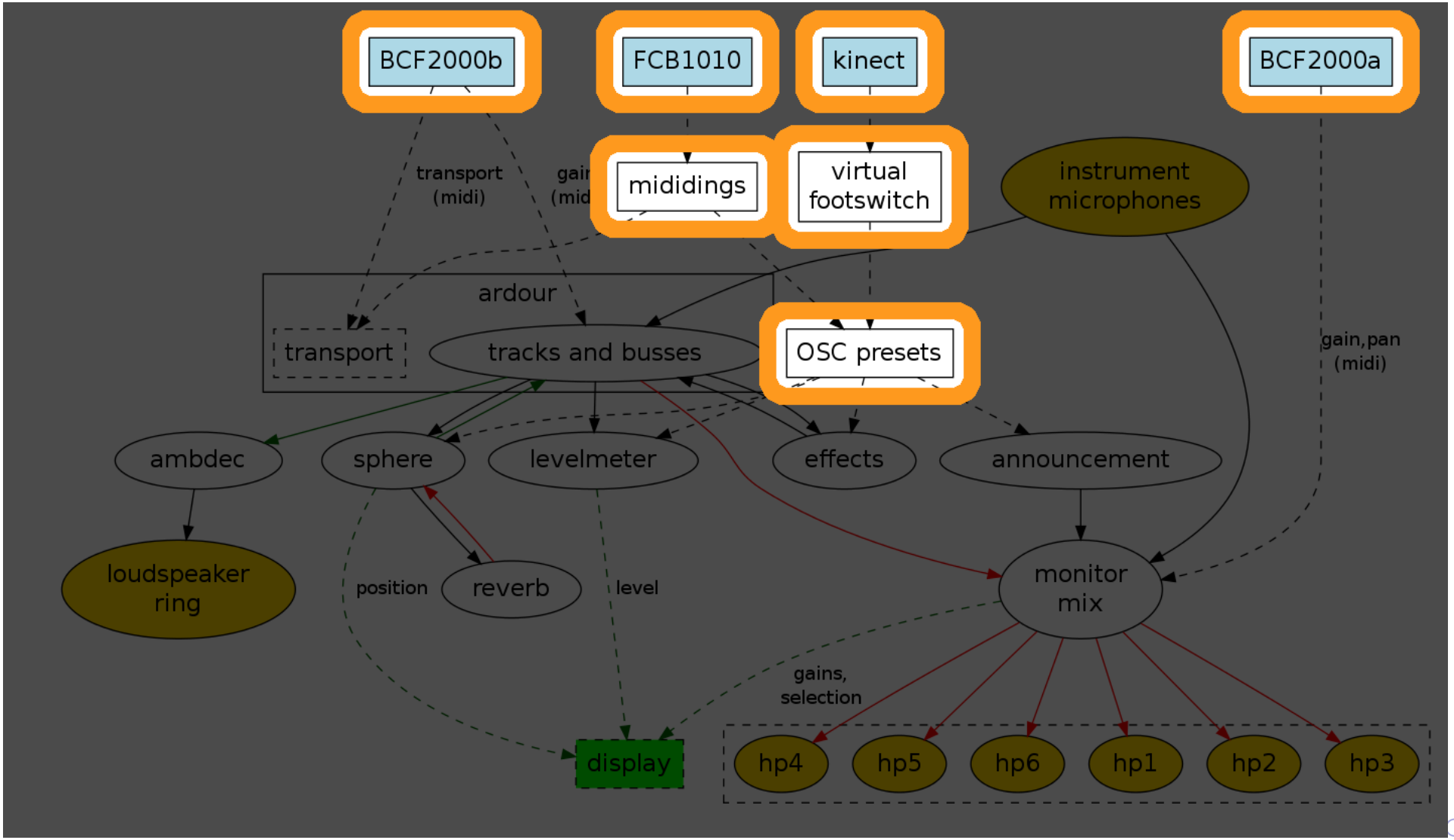
- Superposition of Kepler ellipse and epicycle
- Ellipse
  - ρ nominal radius
  - ε eccentricity
  - θ main axis rotation
  - ω nominal angular velocity
- Epicycle
  - ρ_{epi} radius
  - ω_{epi} angular velocity

hos_sphere: Distance coding

- Stereo reverberation
  convolution reverb 'chapel' from Fons Adriaensen
- Virtual stereo source around source direction
- Distance r affects
  - amount g of reverberation
    $g(r) = \frac{r}{1+r}$
  - width w of stereo source
    $w(r) = w_{max} \frac{2r}{r^2+1}$



Control interfaces and parameter selection



Parameter selection with real footswitch

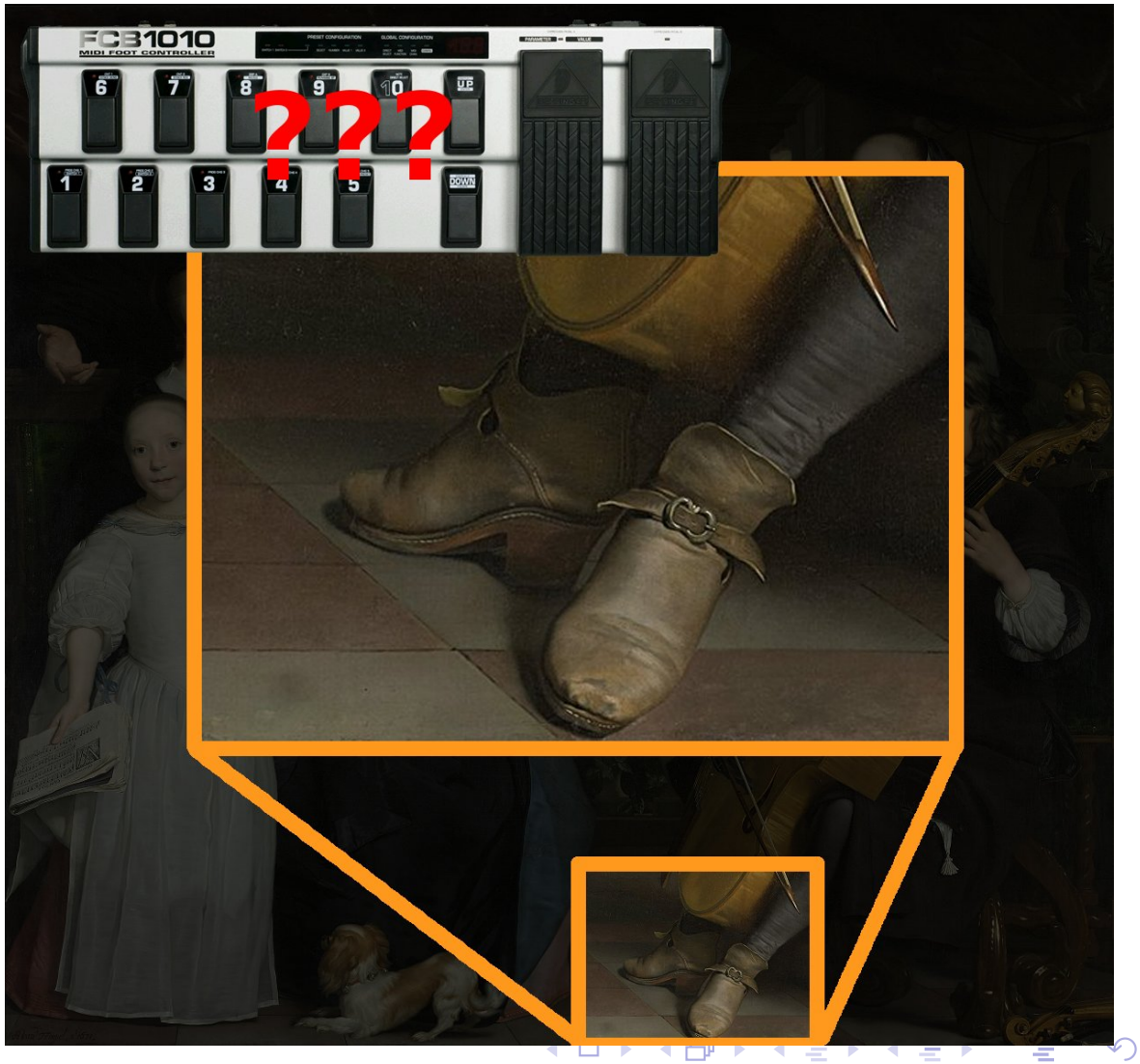
- MIDI foot switch (FCB1010) connected to 'mididings'
- Invoke shell script on program change events
- Read preset files, send as OSC messages to 'hos_sphere'
- Control ardour transport with OSC messages

Parameter selection with virtual footswitch

- Viola da gamba is held between the legs

Parameter selection with virtual footswitch

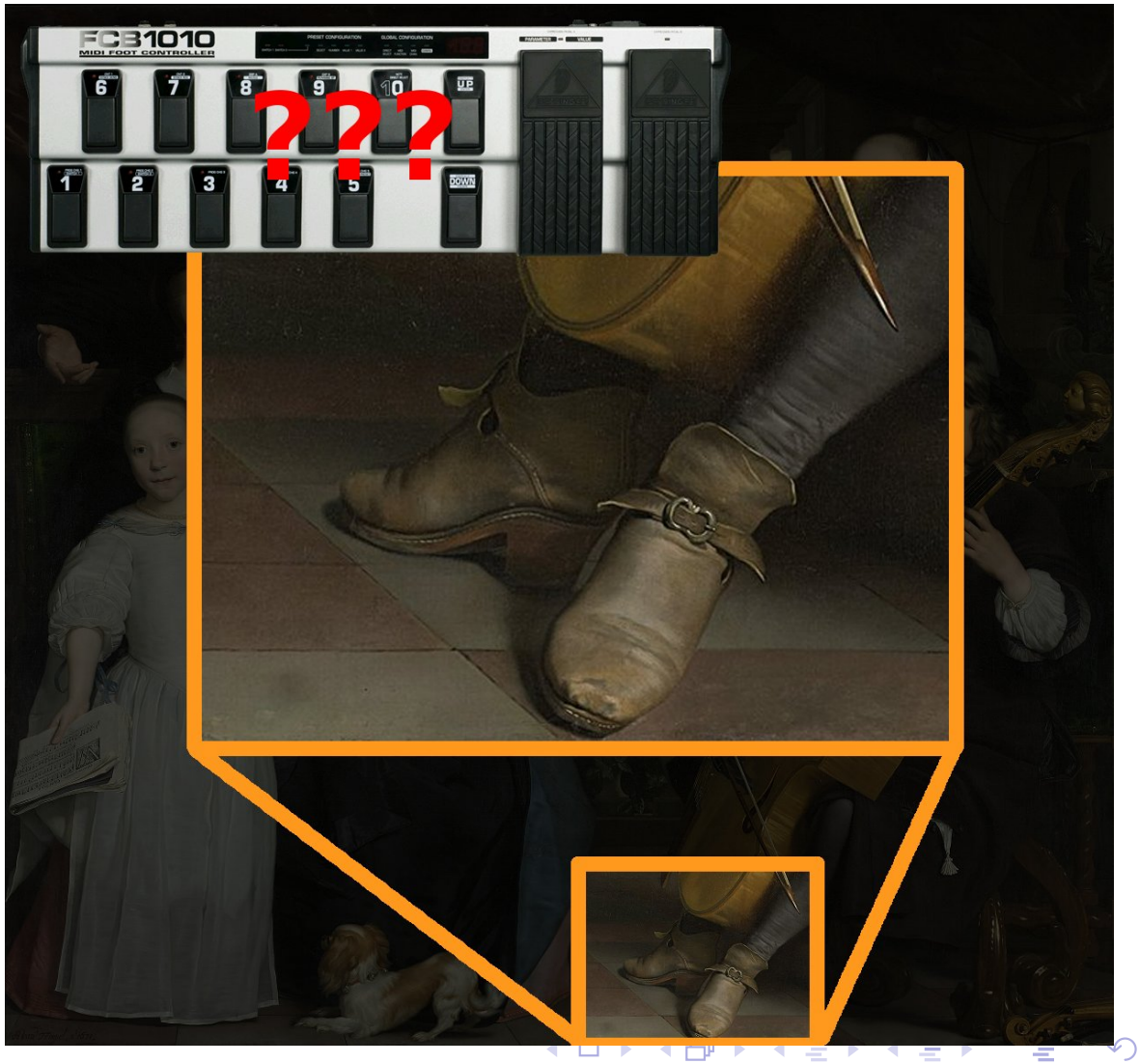
- Viola da gamba is held between the legs
- Lifting the heel is difficult



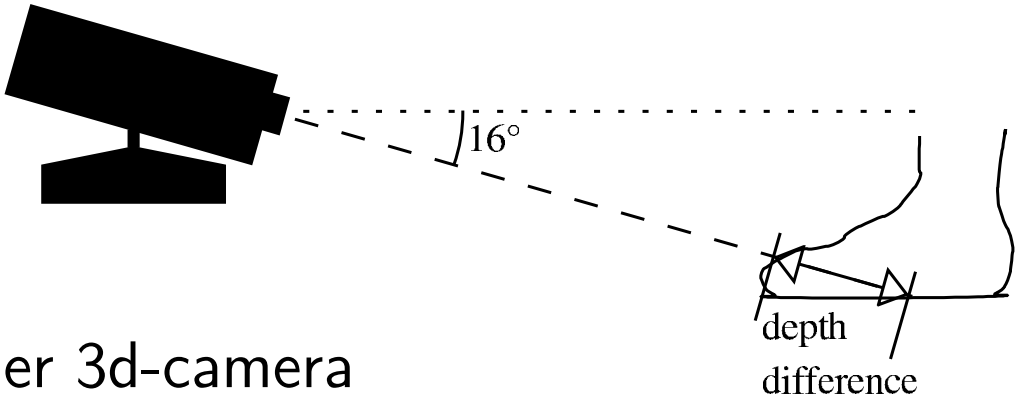
Parameter selection with virtual footswitch

- Viola da gamba is held between the legs
- Lifting the heel is difficult

⇒ Optical foot scanner

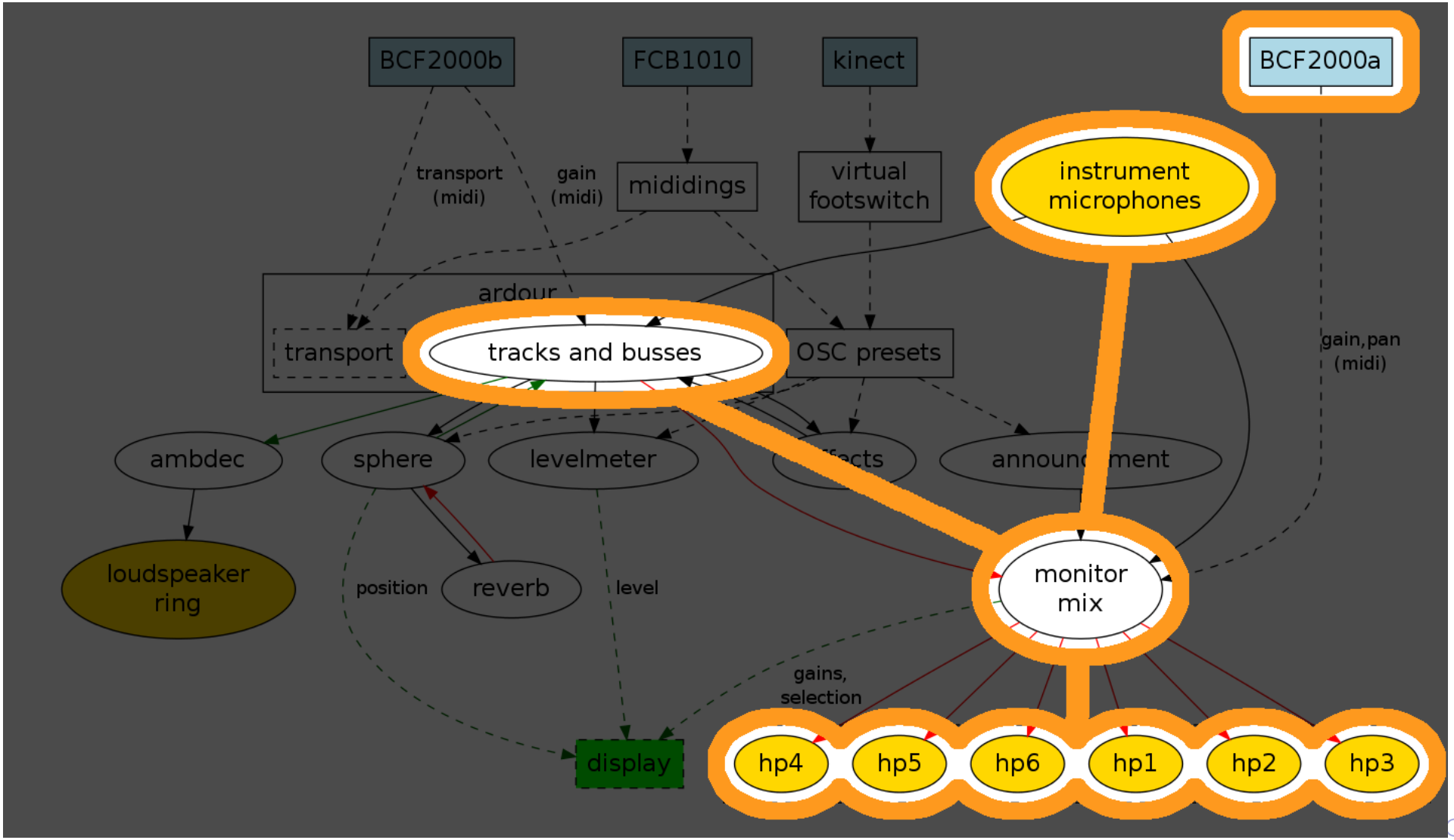


Parameter selection with virtual footswitch



- kinect game controller 3d-camera
- Track position of foot tip
- Four states can be detected:
  - empty: no foot present
  - straight: neutral foot position
  - right: increase program number
  - left: decrease program number
- Shell scripts executed on program change

Monitoring


BCF2000b
FCB1010
kinect
BCF2000a
transport (midi)
gain (midi)
mididings
virtual footswitch
instrument microphones
ardour
transport
tracks and busses
OSC presets
gain,pan (midi)
ambdec
sphere
levelmeter
effects
announcement
loudspeaker ring
position
reverb
level
monitor mix
gains, selection
display
hp4
hp5
hp6
hp1
hp2
hp3

Monitoring: Requirements

- Concert setup with huge distances between musicians
  $\rightarrow$ 'Zero latency' in-ear monitoring is necessary
- Only dry sound monitor
- Pre-recorded announcements
  (program change, time, count in)
- Spatial cues by interaural level differences
- Real-time modifications during sound check & concert

Monitoring: Hard- and software

- RME hdsp9652 matrix mixer
- 'hdspmixer' is missing hardware interface and grouping features
  $\Rightarrow$ own software
- OSC based network solution:
  - 'mm_hdsp': access to hardware mixer
  - 'mm_midicc': interface to MIDI controller (BCF2000)
  - 'mm_file': load, save and manipulate mixer matrix
  - 'mm_gui': visualize gain matrix

Monitoring: Hard- and software

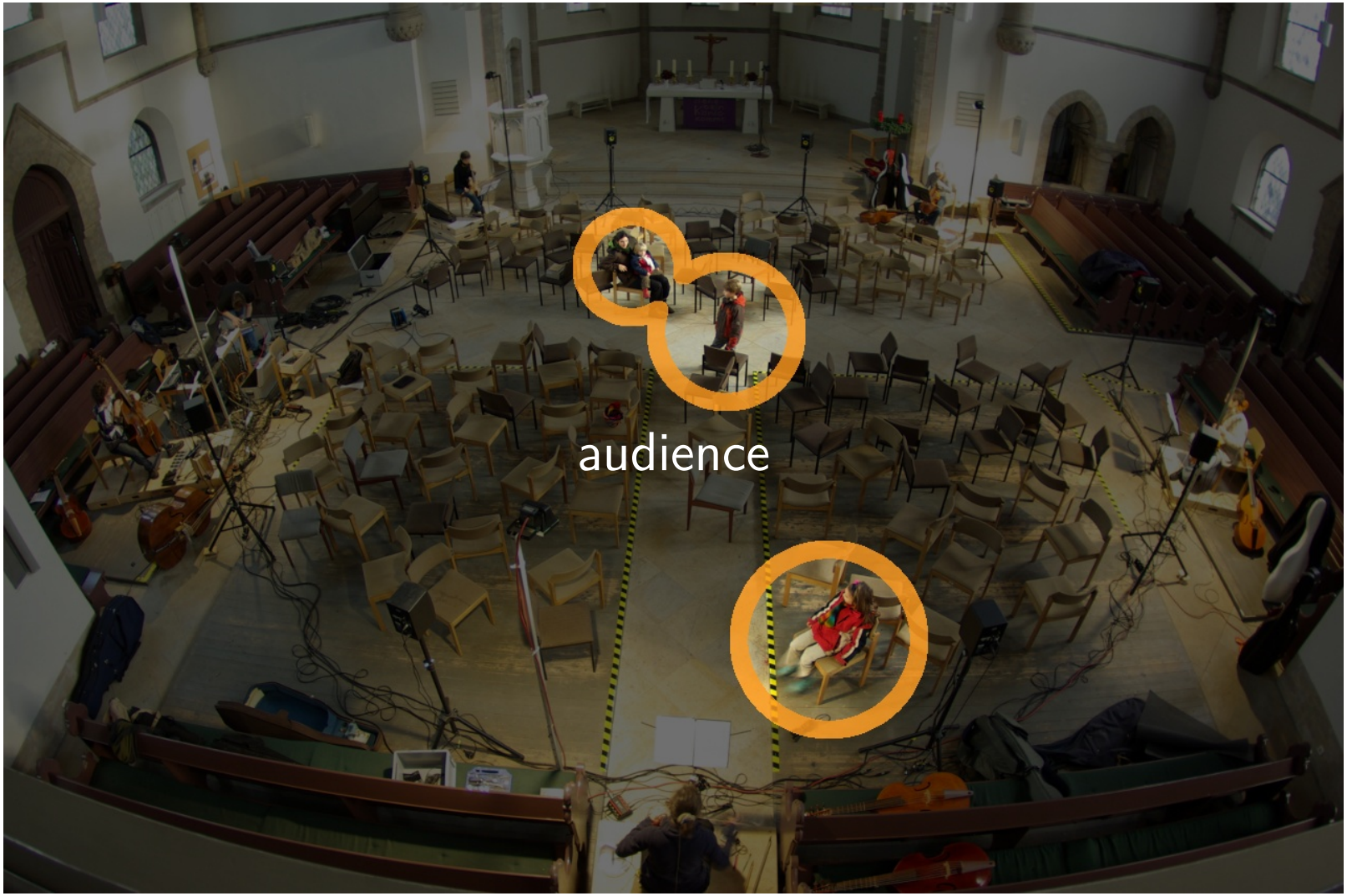
- sub-set of gain matrix
  $\rightarrow$ no conflict with other mixing software
- Assign names to inputs and outputs
- Routes can be multi-channel

Concert setup

Concert setup

Concert setup

Concert setup



Concert setup

Concert setup

Demonstration video

today 4:45pm → listening room

Thank you and all other LADs!

Supported by

- German research funding organization DFG, Research Unit 1732 "Individualisierte Hörakustik"
- "klangpol – Neue Musik im Nordwesten" (Contemporary Music in north-west Germany).

Appendix

'tascar_gpx2csv'

- Tangent projection of GPS coordinates to local coordinates
- Smoothing and resampling to correct errors and jitter
- Time shift to match jack transport time

Example configuration

```
<origin src="trkpt" mode="tangent">
  <trkpt lat="53.151367" lon="8.200514"/>
</origin>
<scale x="1" y="1" z="0"/>
<resample dt="0.25"/>
<smooth n="11"/>
```

'tascar_multipan'

- Multiple panning methods & render, switchable via OSC messages over UDP multicast (`/*/mode`):
  - Nearest speaker (`nearest`)
  - Vector based amplitude panning (`vbap`)
  - Higher Order Ambisonics (`amb_basic`, `amb_inphase`)
  - Wave field synthesis (`wfs`)
  - Cardioid microphone array, e.g. ORTF (`cardiod`)
  - Direct HOA output planned
- Speaker/microphone configuration in XML

'tascar_multipan'

- Multiple panning methods & render, switchable via OSC messages over UDP multicast (`/*/mode`):
  - Nearest speaker (`nearest`)
  - Vector based amplitude panning (`vbap`)
  - Higher Order Ambisonics (`amb_basic`, `amb_inphase`)
  - Wave field synthesis (`wfs`)
  - Cardioid microphone array, e.g. ORTF (`cardiod`)
  - Direct HOA output planned
- Speaker/microphone configuration in XML
- Distance coding with delay and gain simulation
  $\Rightarrow$ Doppler effect
- Delay- and gain compensation of speaker distances

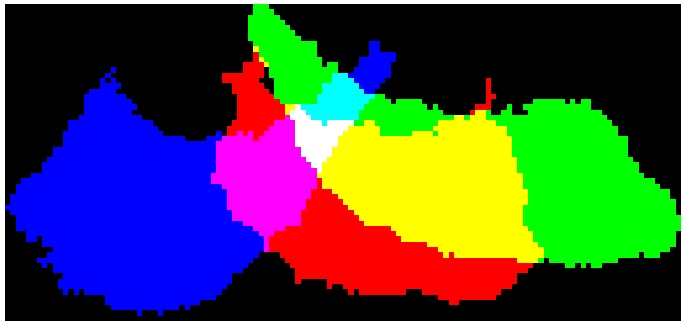
hos_sphere: Trajectory generator

Kepler ellipse with epicycle component:

$$
\begin{aligned}
\rho_\varepsilon(\varphi) &= \frac{\rho_0\sqrt{1-\varepsilon^2}}{1-\varepsilon\cos(\varphi-\theta)} \\
z(\varphi,\varphi_{epi}) &= \rho_\varepsilon e^{i\varphi} + \rho_{epi} e^{i\varphi_{epi}} \\
\Delta\varphi &= \frac{\omega\rho^2}{\rho_\varepsilon^2} \\
\Delta\varphi_{epi} &= \omega_{epi}
\end{aligned}
$$

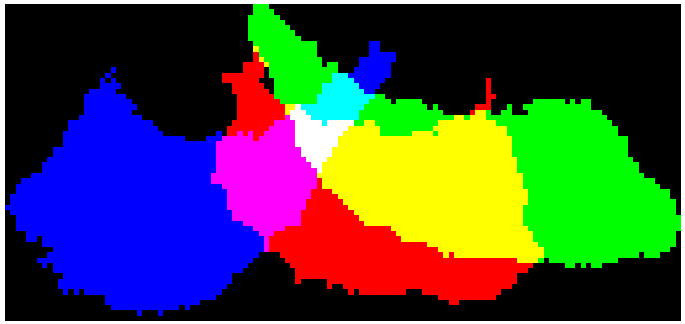
Also: Random component, φ proportional to signal envelope

Parameter selection with virtual footswitch



- Training phase (before concert)
  - Calculating depth difference to 'empty' state
  - Find clusters with depth differences between states

Parameter selection with virtual footswitch



- Training phase (before concert)
  - Calculating depth difference to 'empty' state
  - Find clusters with depth differences between states
- Test phase (during concert)
  - Calculate error difference to training data
  - Select state with least error
  - State change: left/right detected during 0.25 seconds
  - Predefined shell command is evoked on program change
  - Shell commands trigger OSC messages