High Order Ambisonics, generating and diffusing full surround 3D soundfields

Fernando López Lezcano
CCRMA, Universidad de Stanford
https://ccrma.stanford.edu/~nando

EMW2015

High Order Ambisonics, generating and diffusing full surround 3D soundfields

Fernando López Lezcano
CCRMA, Universidad de Stanford
https://ccrma.stanford.edu/~nando

(thanks to Jörn Nettingsmeier for many inspirational slides)

EMW2015

contents

- problems, solutions
- introduction to ambisonics
  - first order
  - higher orders
- decoding ambisonics
- free software for ambisonics
- examples
  - systems, concerts, pieces

which hat do I wear?

- the composer hat:
  - use all space around the audience for sound
  - forget there are speakers
- the sound engineer hat:
  - transparent and accurate sound
- the concert producer hat:
  - easily accomodate pieces created for many different speaker configurations

creating surround

- normal workflow:
  - multichannel DAW
  - mixing bus, as many channels as speakers
  - panners to place sound in between speakers

your piece is a multichannel soundfile, each channel goes to a speaker

creating surround

forget that!

spatial audio woes...

spatial audio woes...

great! people want to hear my music!

spatial audio woes...

great! people want to hear my music!

… only there are fewer speakers available than my piece requires

spatial audio woes...

great! people want to hear my music!

… only there are fewer speakers available than my piece requires
… or they are in the wrong places and can't be moved

spatial audio woes...

great! people want to hear my music!

… only there are fewer speakers available than my piece requires
… or they are in the wrong places and can't be moved
… or there are more, but using them properly would require a remix session and studio time

more spatial audio woes...

more spatial audio woes...

- why does my sound change when I move it around?

more spatial audio woes...

- why does my sound change when I move it around?
- why does my sound stick to the speaker, then jump across, when I want uniform motion?

more spatial audio woes...

- why does my sound change when I move it around?
- why does my sound stick to the speaker, then jump across, when I want uniform motion?
- how do I create convincing (or even correct) reverbs in surround?

more spatial audio woes...

- why does my sound change when I move it around?
- why does my sound stick to the speaker, then jump across, when I want uniform motion?
- how do I create convincing (or even correct) reverbs in surround?
- how do I create stereo fold-downs for home use or grant applications, without doing a full remix?

ambisonics to the rescue!

ambisonics to the rescue!

- you can decode your mix to various speaker layouts without manual intervention.

ambisonics to the rescue!

- you can decode your mix to various speaker layouts without manual intervention.
- your music will be downwards compatible, and degrade gracefully all the way down to mono.

ambisonics to the rescue!

- you can decode your mix to various speaker layouts without manual intervention.
- your music will be downwards compatible, and degrade gracefully all the way down to mono.
- your music will be upwards compatible, and make good use of all available speakers.

ambisonics to the rescue!

- sources will sound the same on or between speakers.

ambisonics to the rescue!

- sources will sound the same on or between speakers.
- panning will be perfectly smooth, and speaker locations inaudible.

ambisonics to the rescue!

- sources will sound the same on or between speakers.
- panning will be perfectly smooth, and speaker locations inaudible.
- using ambisonic IRs and convolution, you can recreate natural ambience perfectly.

ambisonics to the rescue!

- sources will sound the same on or between speakers.
- panning will be perfectly smooth, and speaker locations inaudible.
- using ambisonic IRs and convolution, you can recreate natural ambience perfectly.
- stereo and 5.0 fold-downs can be created automatically.

so, are you sold?

ambisonics is free

ambisonics is free

- invented in the 70's, all relevant patents have expired

ambisonics is free

- invented in the 70's, all relevant patents have expired
- very well designed (british engineering)

ambisonics is free

- invented in the 70's, all relevant patents have expired
- very well designed (british engineering)
- but forgotten for 20 years (british marketing)

how does it work?

- ambisonics is a spatial sampling technique

how does it work?

- ambisonics is a spatial sampling technique
- tries to be physically correct where possible, exploits phychoacoustic effects otherwise

how does it work?

- ambisonics is a spatial sampling technique
- tries to be physically correct where possible, exploits phychoacoustic effects otherwise
- it has a solid mathematical foundation

how does it work?

- ambisonics is a spatial sampling technique
- tries to be physically correct where possible, exploits phychoacoustic effects otherwise
- it has a solid mathematical foundation
- no need to panic:
  - *we are going to skip the math*

how does it work?

- how do we “sample” a soundfield?

how does it work?

- how do we “sample” a soundfield?

  microphones can be thought of as spatial sampling instruments... so...

how does it work?

- how do we “sample” a soundfield?

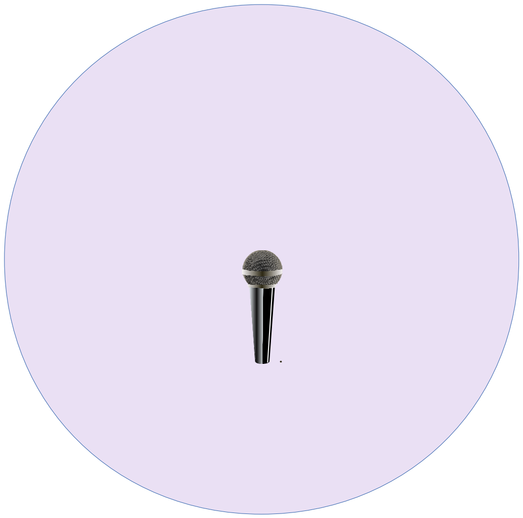
    microphones can be thought of as spatial sampling instruments... so...

    - we need microphones with polar patterns that will cover the sphere uniformly

how does it work?

- how do we "sample" a soundfield?
  - an omnidirectional microphone: it covers the whole sphere uniformly, but it does not have directional information

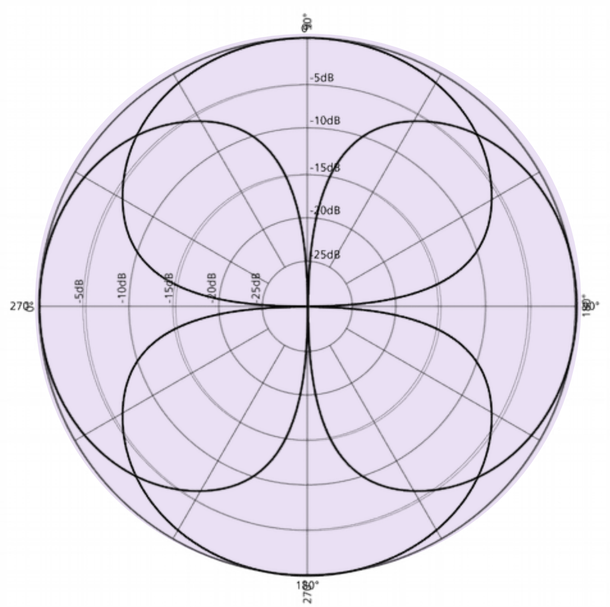
W



how does it work?

- how do we “sample” a soundfield?
  - three figure of 8 microphones at right angles to each other also cover the sphere uniformly

X,Y,Z



how does it work?

- how do we “sample” a soundfield?
  - these four microphones capture the entire soundfield without redundancy

    this is what is called “first order ambisonics”, or “b-format” for short

W,X,Y,Z

recording ambisonics

- first order ambisonics microphone
  - native: use an omni and figure of 8's
  - in real life: four capsules in the vertices of a tetraedron, we derive the ambisonics signals from them

W = LFD + RFU + LBU + RBD
X = LFD + RFU – LBU – RBD
Y = LFD – RFU + LBU – RBD
Z = - LFD + RFU + LBU – RBD

WXYZ = B-format

generating ambisonics

- first order ambisonics panner
  - these are the formulas that generate a first order ambisonics signal from a mono signal “S” and its azimuth and elevation angles:

W = S * 0.707
X = S * cos(AZ) * cos(EL)
Y = S * sin(AZ) * cos(EL)
Z = S * sin(EL)

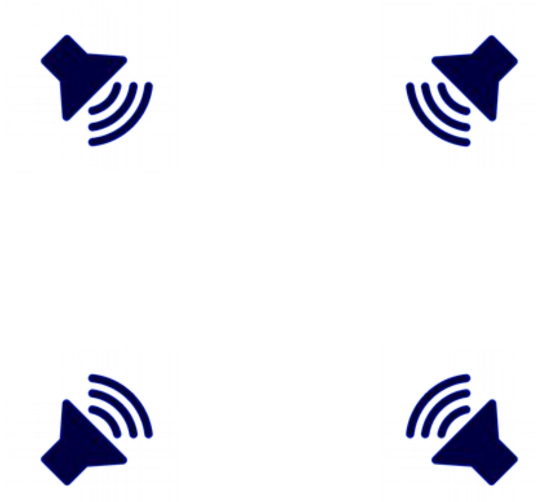
playing ambisonics

- how do we play an ambisonics signal?
  - we need a decoder that projects the components of the ambisonics signal to the directions of the speakers we have

playing ambisonics

- if we have just four speakers:

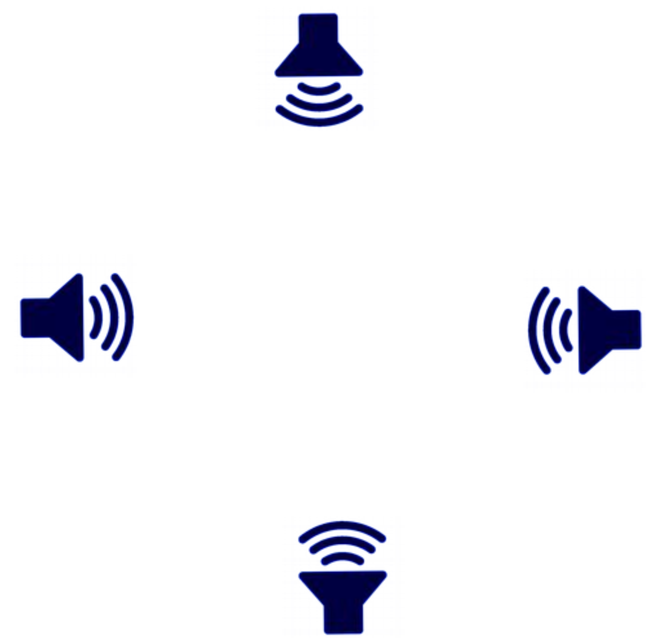
LF = W + X + Y
RF = W + X – Y
LB = W – X + Y
RB = W – X – Y



playing ambisonics

- if we have just four speakers (different layout):

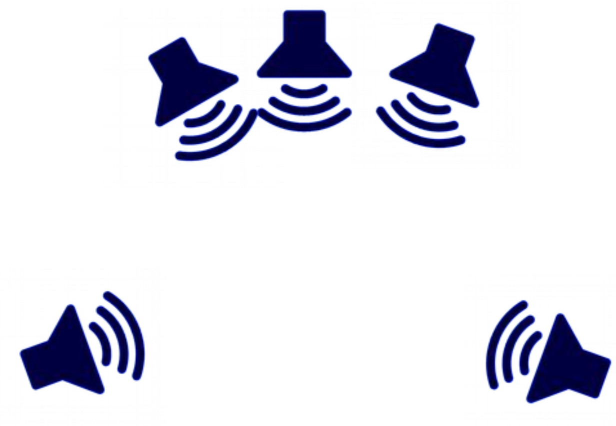
$$F = W + X$$
$$L = W + Y$$
$$R = W - Y$$
$$B = W - X$$



playing ambisonics

- if we have five speakers in weird positions:

irregular layouts lose some spatial resolution and will introduce slight localisation errors



ambisonics problems

- very low angular resolution (same as a figure of 8 microphone – basically a cardiod)
- small “sweet spot” when decoding to speakers

ambisonics problems

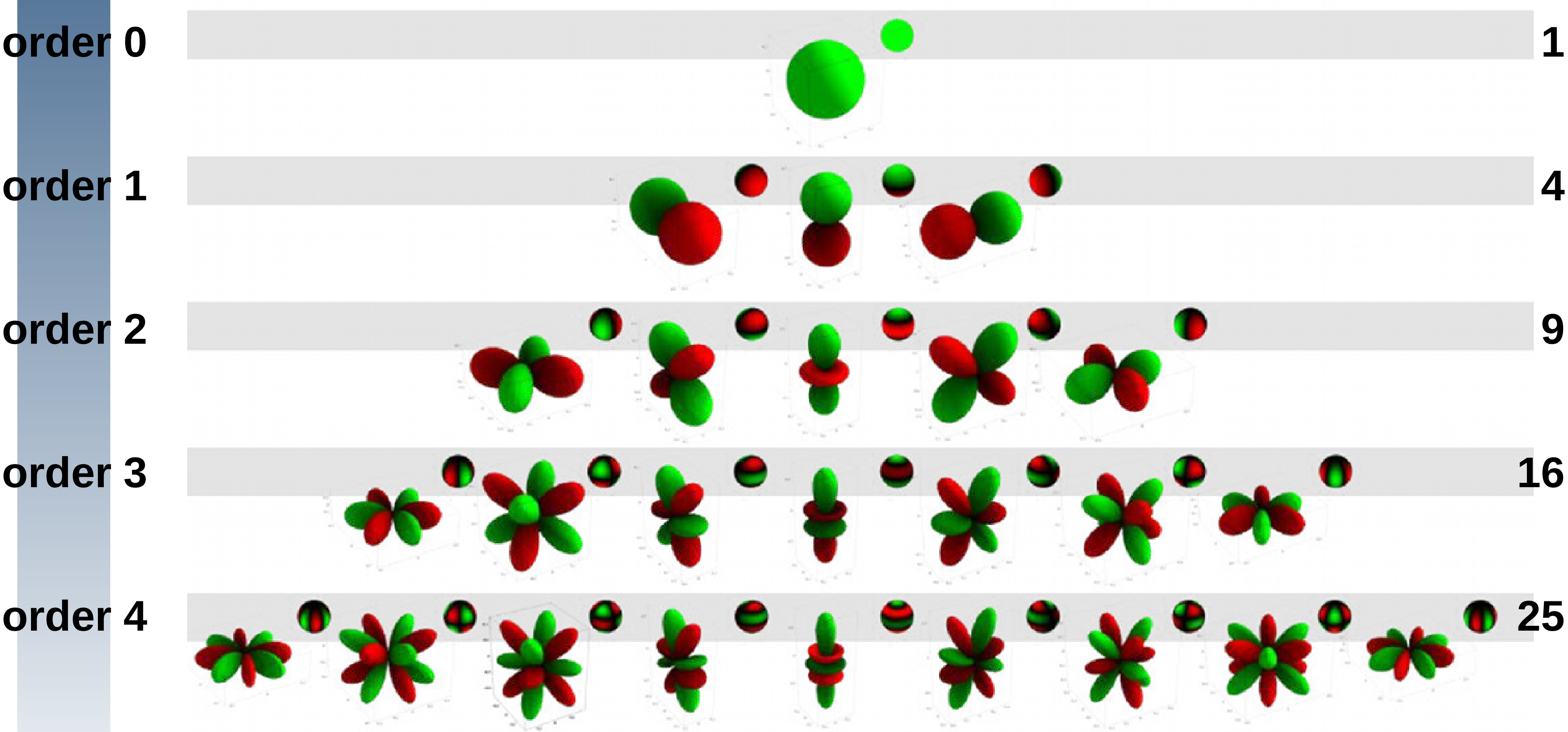
- how can we get a better angular resolution?

ambisonics problems

- how can we get a better angular resolution?
  - we need “microphones” with better angular resolution that sample the whole sphere uniformly and without redundancy

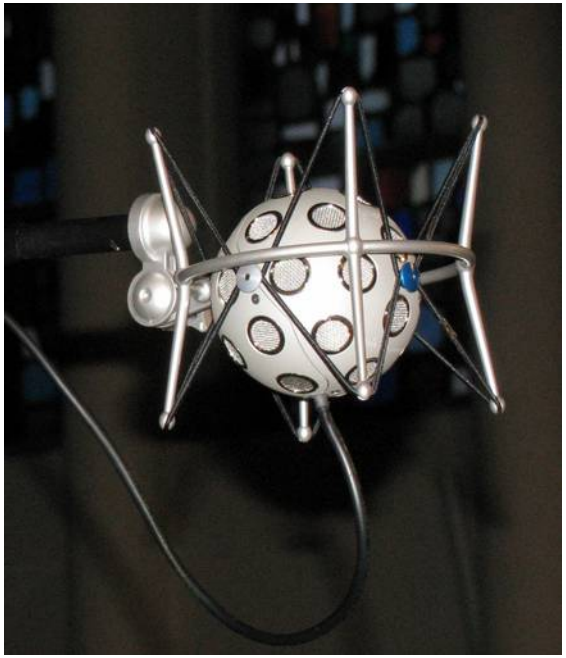
high order ambisonics

- here they are, the spherical harmonics:



recording high order ambisonics

- high order microphones...
  - there aren't any, but we can derive ambisonics signals from microphones like this one:



recording high order ambisonics

- high order microphones...
  - and here is one that a group at ccrma is building:

generating high order ambisonics

- high order panner
  - easy, just a matter of implementing the equations

    these are the equations that generate a third order Ambisonics signal (16 channels) in ACN order and with SN3D normalization

$$1$$
$$\sin(\theta)\cdot\cos(\psi)$$
$$\sin(\psi)$$
$$\cos(\psi)\cdot\cos(\theta)$$
$$\sqrt{3}\cdot\sin(\theta)\cdot\cos^2(\psi)\cdot\cos(\theta)$$
$$\frac{\sqrt{3}\cdot(\cos(2\cdot\psi-\theta)-\cos(2\cdot\psi+\theta))}{4}$$
$$\frac{3\cdot\sin^2(\psi)}{2}-\frac{1}{2}$$
$$\frac{\sqrt{3}\cdot(\sin(2\cdot\psi-\theta)+\sin(2\cdot\psi+\theta))}{4}$$
$$\frac{\sqrt{3}\cdot\cos^2(\psi)\cdot\cos(2\cdot\theta)}{2}$$
$$\frac{\sqrt{10}\cdot\left(-4\cdot\sin^2(\theta)+3\right)\cdot\sin(\theta)\cdot\cos^3(\psi)}{4}$$
$$\sqrt{15}\cdot\sin(\psi)\cdot\sin(\theta)\cdot\cos^2(\psi)\cdot\cos(\theta)$$
$$\frac{\sqrt{6}\cdot\left(5\cdot\sin^2(\psi)-1\right)\cdot\sin(\theta)\cdot\cos(\psi)}{4}$$
$$\frac{\left(5\cdot\sin^2(\psi)-3\right)\cdot\sin(\psi)}{2}$$
$$\frac{\sqrt{6}\cdot\left(5\cdot\sin^2(\psi)-1\right)\cdot\cos(\psi)\cdot\cos(\theta)}{4}$$
$$\frac{\sqrt{15}\cdot\sin(\psi)\cdot\cos^2(\psi)\cdot\cos(2\cdot\theta)}{2}$$
$$\frac{\sqrt{10}\cdot\left(-4\cdot\sin^2(\theta)+1\right)\cdot\cos^3(\psi)\cdot\cos(\theta)}{4}$$

generating high order ambisonics

think about it:

a linear combination of those very weird shapes (the spherical harmonics) can "point" a very narrow microphone in any direction!

$$1$$
$$\sin(\theta)\cdot\cos(\psi)$$
$$\sin(\psi)$$
$$\cos(\psi)\cdot\cos(\theta)$$
$$\sqrt{3}\cdot\sin(\theta)\cdot\cos^2(\psi)\cdot\cos(\theta)$$
$$\frac{\sqrt{3}\cdot(\cos(2\cdot\psi-\theta)-\cos(2\cdot\psi+\theta))}{4}$$
$$\frac{3\cdot\sin^2(\psi)}{2}-\frac{1}{2}$$
$$\frac{\sqrt{3}\cdot(\sin(2\cdot\psi-\theta)+\sin(2\cdot\psi+\theta))}{4}$$
$$\frac{\sqrt{3}\cdot\cos^2(\psi)\cdot\cos(2\cdot\theta)}{2}$$
$$\frac{\sqrt{10}\cdot\left(-4\cdot\sin^2(\theta)+3\right)\cdot\sin(\theta)\cdot\cos^3(\psi)}{4}$$
$$\sqrt{15}\cdot\sin(\psi)\cdot\sin(\theta)\cdot\cos^2(\psi)\cdot\cos(\theta)$$
$$\frac{\sqrt{6}\cdot\left(5\cdot\sin^2(\psi)-1\right)\cdot\sin(\theta)\cdot\cos(\psi)}{4}$$
$$\frac{\left(5\cdot\sin^2(\psi)-3\right)\cdot\sin(\psi)}{2}$$
$$\frac{\sqrt{6}\cdot\left(5\cdot\sin^2(\psi)-1\right)\cdot\cos(\psi)\cdot\cos(\theta)}{4}$$
$$\frac{\sqrt{15}\cdot\sin(\psi)\cdot\cos^2(\psi)\cdot\cos(2\cdot\theta)}{2}$$
$$\frac{\sqrt{10}\cdot\left(-4\cdot\sin^2(\theta)+1\right)\cdot\cos^3(\psi)\cdot\cos(\theta)}{4}$$

playing high order ambisonics

- decoding hoa to speakers

ambisonics decoders

- an ambisonics decoder has to have:
  - rV: particle velocity vector (ITD) <400Hz

    should point to the virtual source

    should have amplitude 1 or close to 1
  - rE: energy vector (related to ILD) 400Hz .. 4KHz

    should point to the virtual source (and coincide with rV)

    maximum possible amplitude (will be always less than 1)

    - implemented with a “phase matched shelf filter”
  - near field correction filters (for each speaker if speaker distances vary)

ambisonics decoders

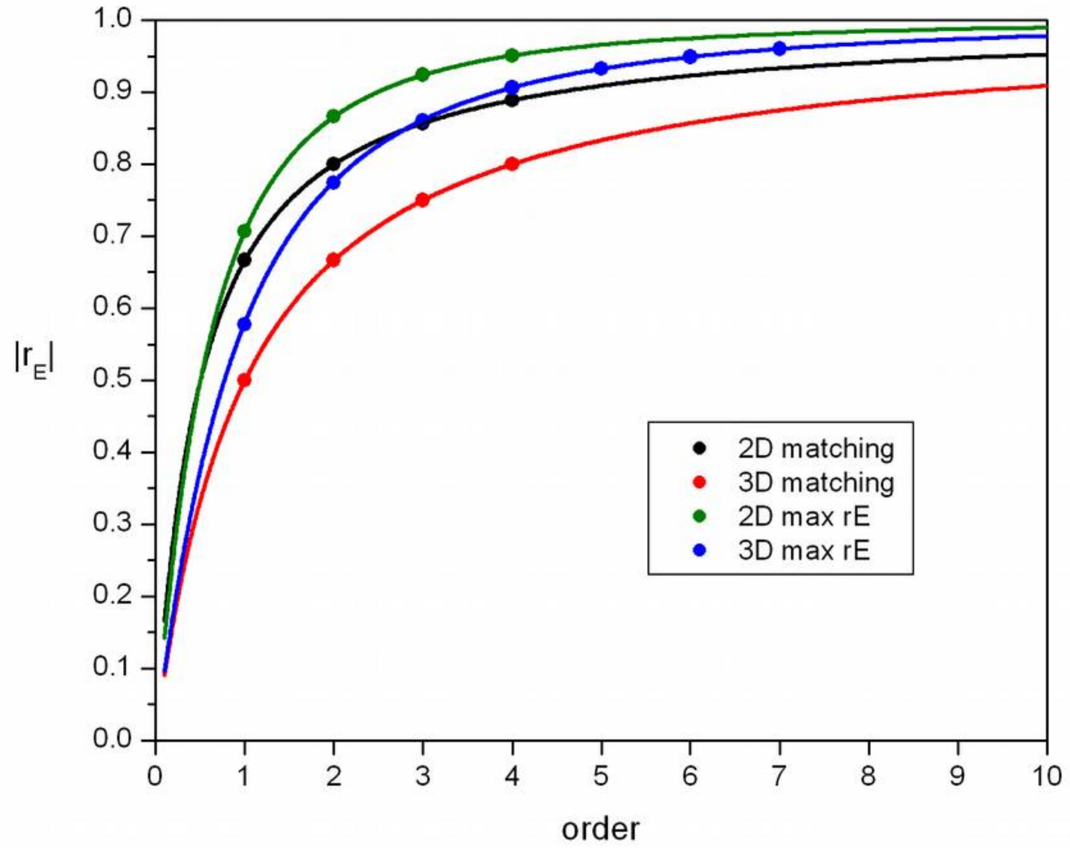
- regular speaker arrays:
  - pseudoinverse method

    we invert the matrix created by sampling the spherical harmonics in the directions of the speakers, invert the matrix

    - this gives us an rV decoder (ITD)
    - for rE (ILD) we adjust the pressure/velocity ratio (relationship between W and the higher order components)

ambisonics decoders

- max rE vs decoder order (and method)



ambisonics decoders

- irregular speaker arrays :
  - in general the rV and rE vectors don't match...

    we have to see how to invert the matrix

    (tradeoff between uniform energy and angular errors)
- domes
  - no speakers below the audience

ambisonics decoders

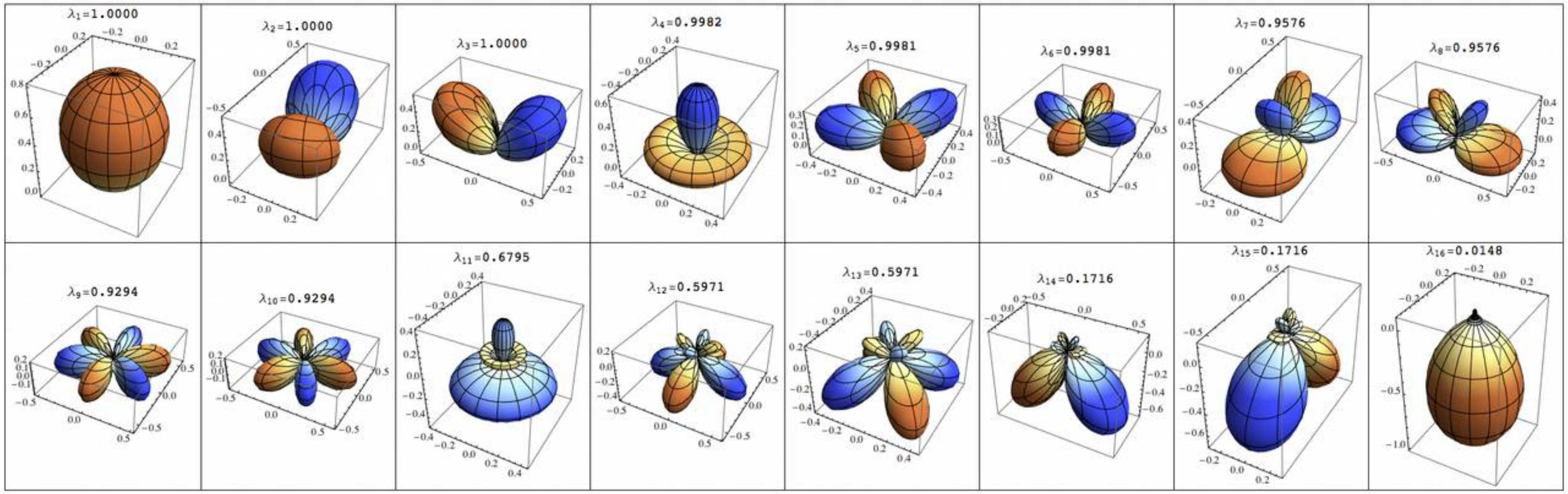
- solutions:
  - start from an array of speakers that covers the sphere uniformly, design a decoder for that and then map the result to the real speakers:

    AllRad method (Zotter / Frank)

    - use a t-design, design the decoder
    - use vbap to map the virtual speakers to the real ones

ambisonics decoders

- solutions:
  - use other functions that are *not* the spherical harmonics and cover only the area of the sphere that we are interested in
    - “slepian” functions



ambisonics decoders

- solutions:
  - all these methods are implemented in ADT (Ambisonics Decoder Toolkit), free software written by Aaron Heller

(free) ambisonics software

(free) ambisonics software

- ardour + LADSPA AMB-plugins (FUMA), Fons Adriansen, Joern Nettingsmeier
  - create a session with a 16 channel master bus
  - use the 3,3 ambisonics panner plugin (replacing the native panner)

    http://cec.sonus.ca/econtact/11_3/nettingsmeier_ambisonics.html
- howto: ambisonics at home (2008), Joern Nettingsmeier
  - paper:

    http://stackingdwarves.net/public_stuff/linux_audio/ambi_at_home/joern_nettingsmeier-ambisonics_at_home.pdf
  - slides:

    http://stackingdwarves.net/public_stuff/linux_audio/ambi_at_home/AMBI@Home.pdf

(free) ambisonics software

- ambi-X: cross-platform ambisonics plugins Matthias Krolachner
  - http://www.matthiaskronlachner.com/?p=2015
  - ambix convention (ACN, SN3D), see this:

    http://iem.kug.ac.at/fileadmin/media/iem/projects/2011/ambisonics11_nachbar_zotter_sontacchi_deleflie.pdf
  - they can work with mcfx, plugin collection for multichannel work:
  - http://www.matthiaskronlachner.com/?p=1910

(free) ambisonics software

- ATK, Ambisonics Toolkit (SuperCollider), Joe Anderson (first order only – includes soundfield transforms)
  - http://www.ambisonictoolkit.net/
- Ambisonics Externals for Max/MSP, UDOs for Csound, ICST (Institute for Computer Music and Sound Technology)
  - http://www.zhdk.ch/index.php?id=71547
- Csound: panning and spatialization (includes panning, vbap and ambisonics examples)
  - http://en.flossmanuals.net/csound/b-panning-and-spatialization/

(free) ambisonics software

- Ambdec, ambisonics decoder, Fons Adriaensen

  dual band, near field compensated, includes presets for most common speaker configurations

- ADT, Ambisonics Decoder Toolbox, Aaron Heller

  decoder design, Octave + Faust

  - https://bitbucket.org/ambidecodertoolbox/adt.git
  - read: The Ambisonic Decoder Toolbox: Extensions for Partial-Coverage Loudspeaker Arrays

    http://lac.linuxaudio.org/2014/papers/17.pdf

example: the GRAIL

- CCRMA's 3D concert speaker array

example: the GRAIL

- CCRMA's 3D concert speaker array
  - Giant Radial Array for Immersive Listening

    named by Alison Rush (PhD student)

example: the GRAIL

- CCRMA's 3D concert speaker array
  - Giant Radial Array for Immersive Listening

    named by Alison Rush (PhD student),

    but it also means
  - GRAIL Renders Ambisonics in Linux

    discovered by Chris Chafe (CCRMA Director)

example: the GRAIL

- CCRMA's 3D concert speaker array
  - Giant Radial Array for Immersive Listening

    named by Alison Rush (PhD student),

    but it also means
  - GRAIL Renders Ambisonics in Linux

    discovered by Chris Chafe (CCRMA Director)

    it is of course a holistic system, so it is
  - the HOLI GRAIL

example: the GRAIL

example: the GRAIL hardware

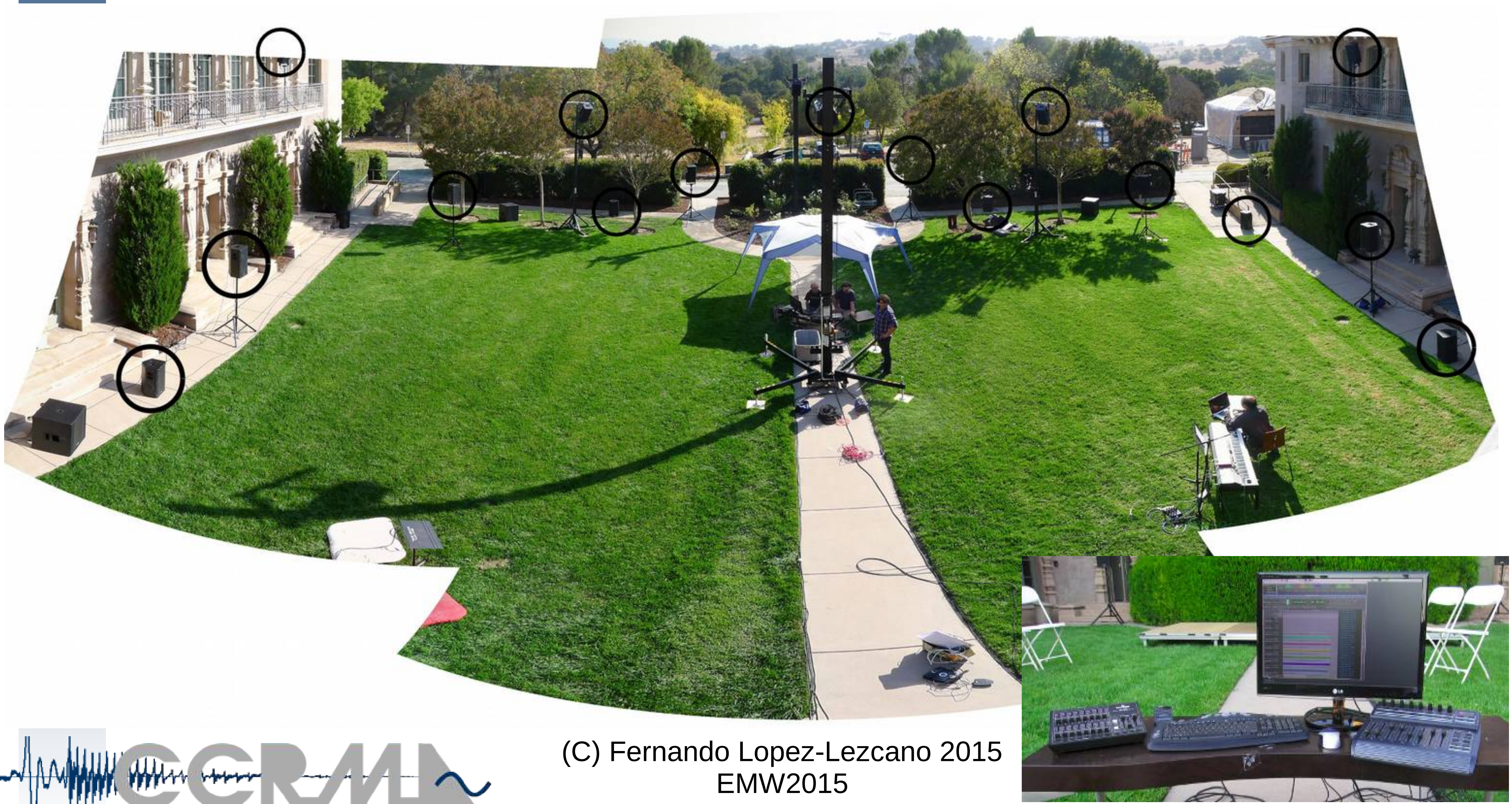
- PC (6 core / 64G ram) running Linux
  - remoted display and usb peripherials
- NetworkSound Digital Snake (1/2)
  - analog I/O (32/32)
- RME RayDAT soundcard
  - digital I/O (4 x ADAT pipes, optional)
- BCF2000 x 2 usb control surfaces
  - control

example: the GRAIL software

- supercollider dsp core
  - delay and volume compensation, LR4 crossover
  - ADT generated ambisonics decoders (2 decoders, one for the main speakers, one for the subwoofers)
- jconvolver backend
  - renders DRC filters for each speaker
- jack-mamba (interface a/d d/a 32 channels), jackd
- ardour2/3
  - GUI front-end for live mixing or playback

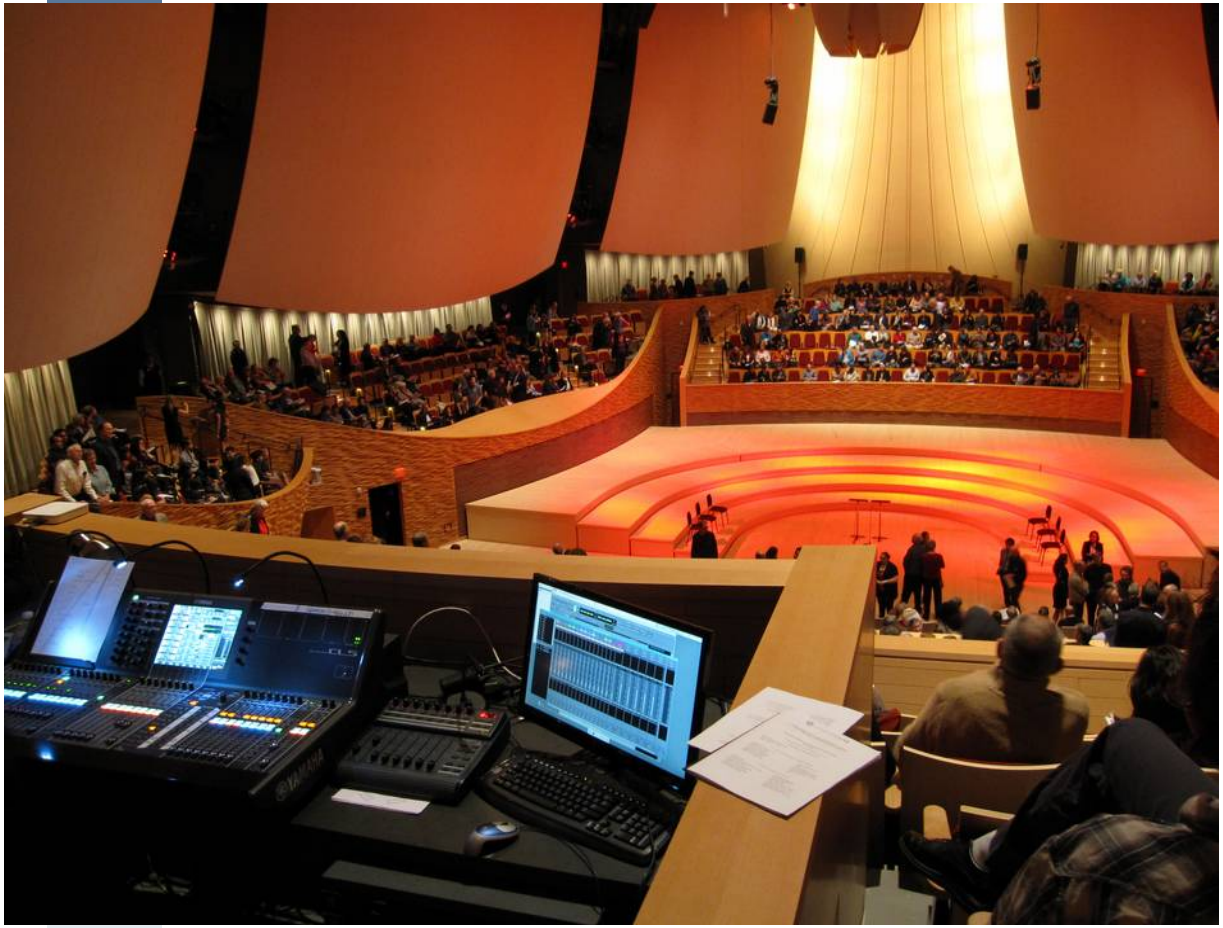
concerts

- 2012: Transitions (24.6)



concerts

- 2013: “From Constantinople to California”



concerts

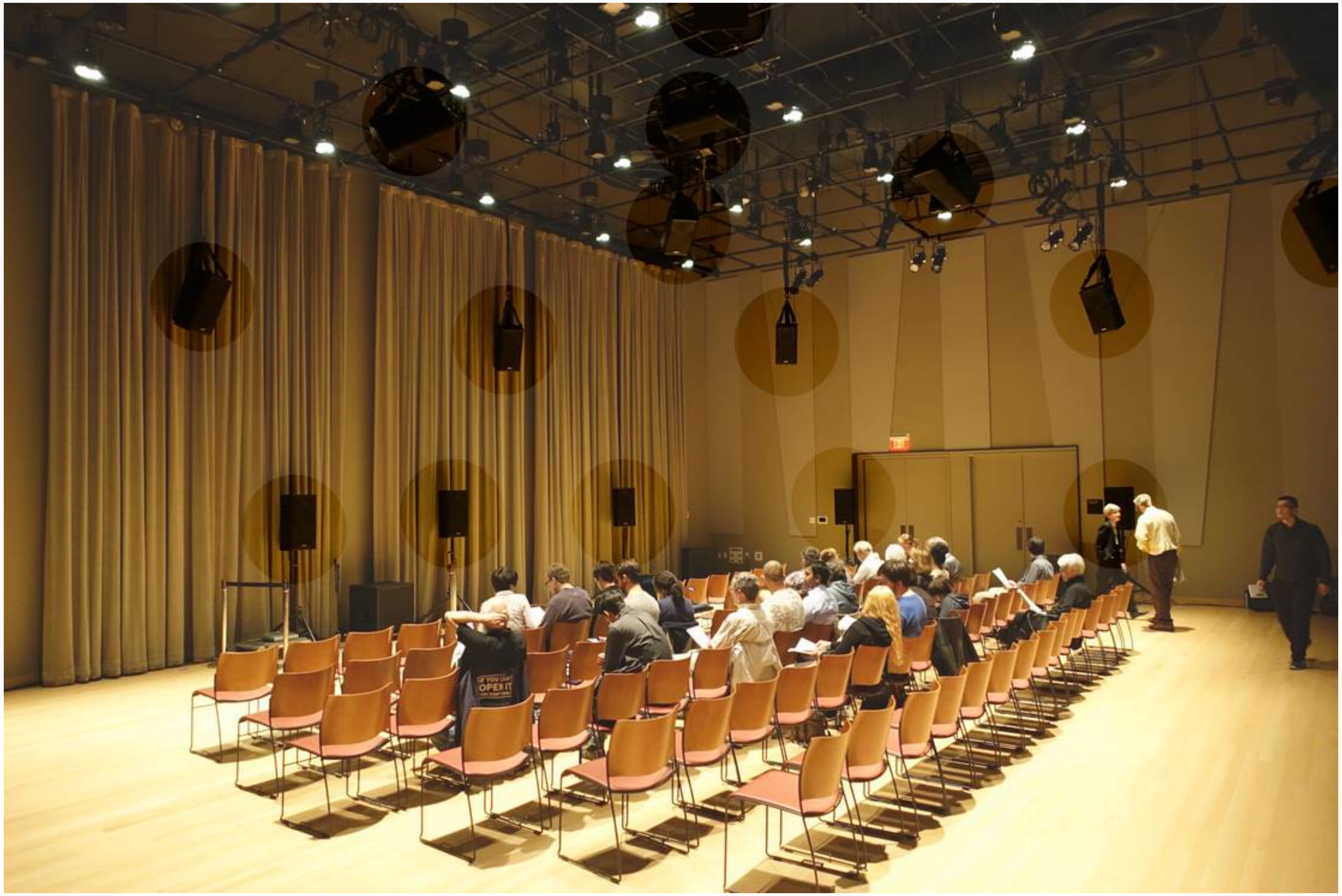
- 2015: Bing Studio (25.7 = 12+8+4+1)

concerts

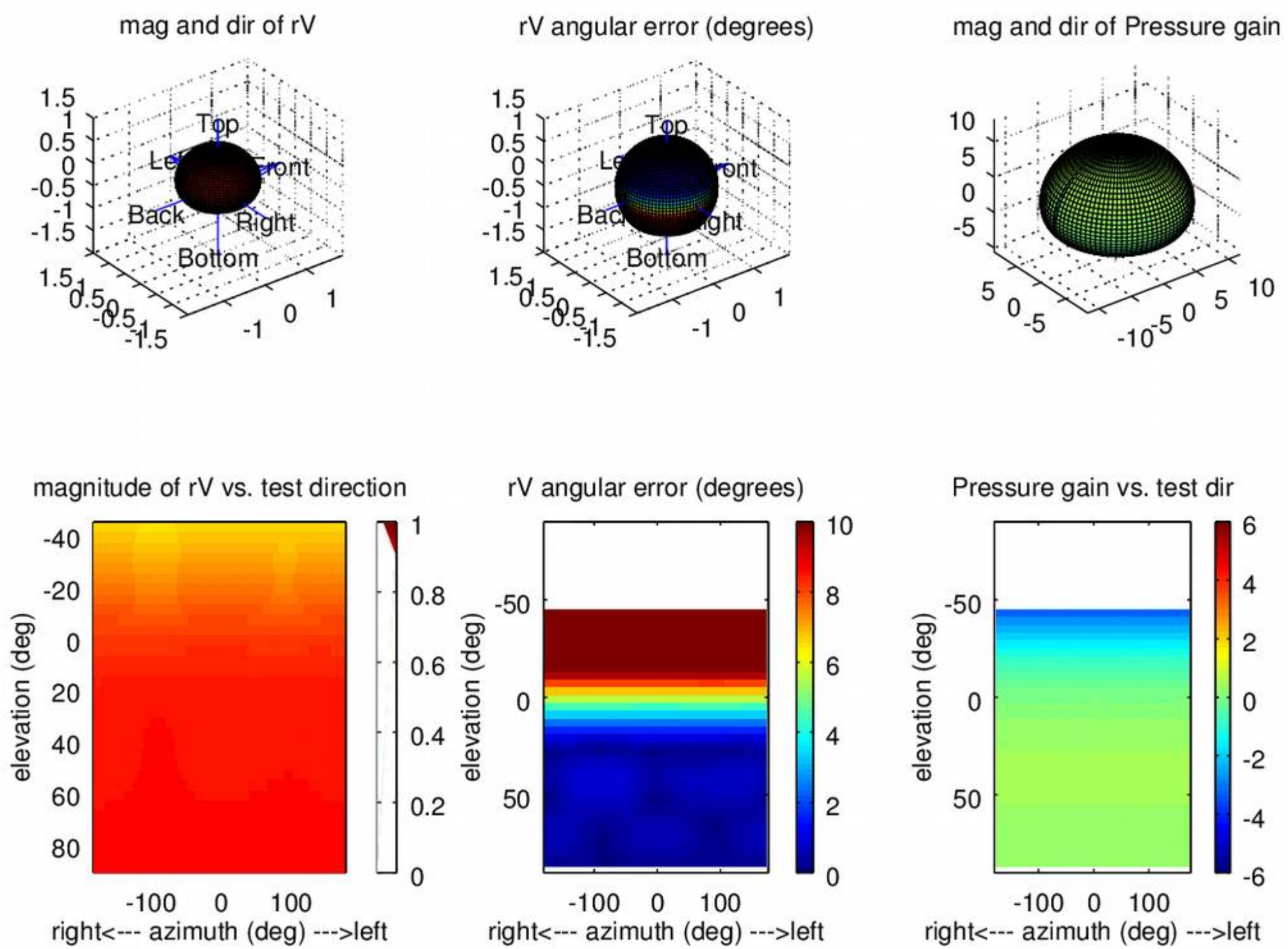
- 2015: Bing Studio (25.7)

concerts

- 2015: Bing Studio (25.7)

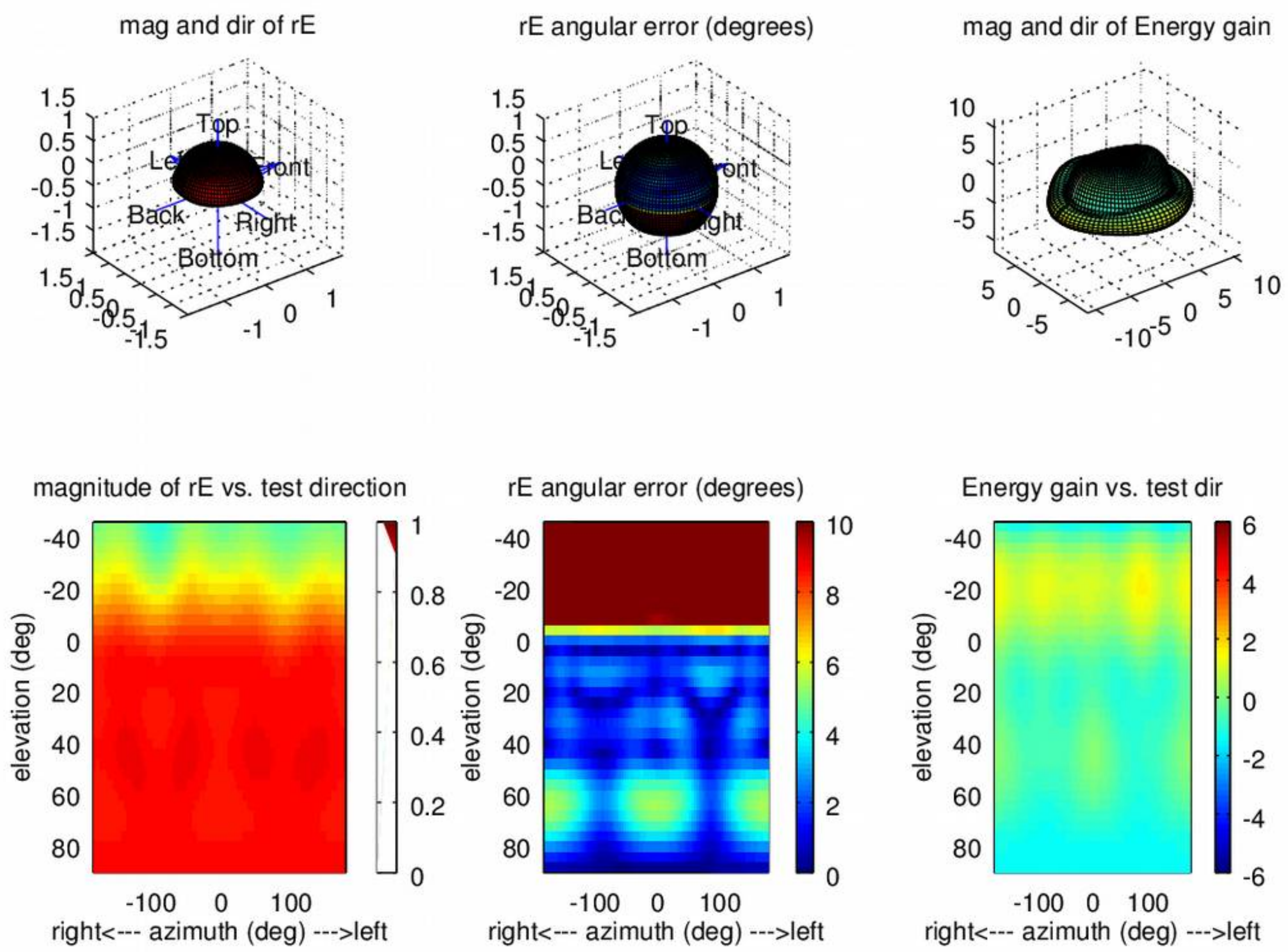


concerts



EMW2015

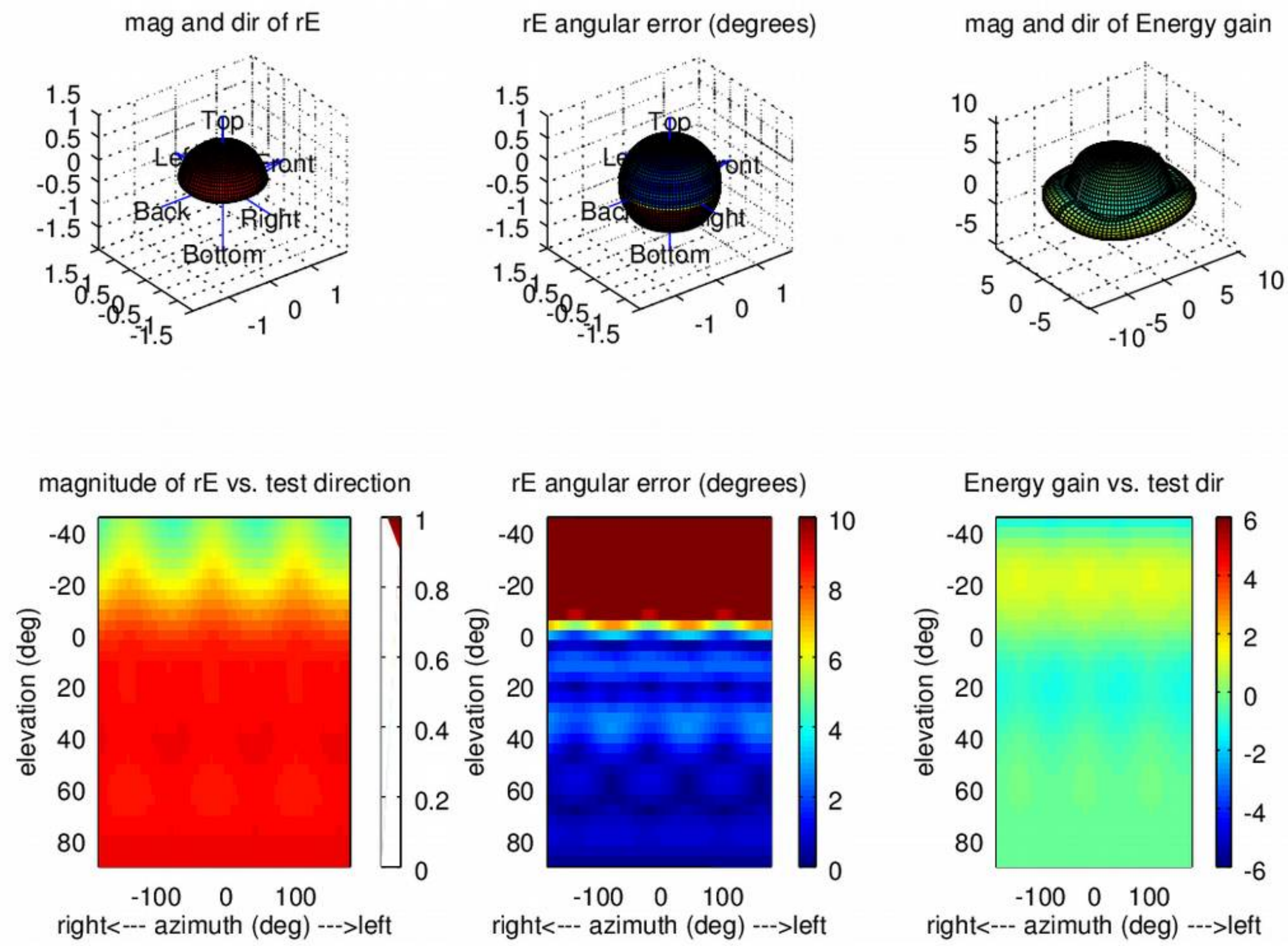
concerts



(C) Fernando Lopez-Lezcano 2015
EMW2015

concerts

changing the locations of the speakers to a more ideal array



concerts

- 2015: Transitions (24.8)
  - outdoors, end of Summer concert
  - two days of setup, calibration and rehearsals
  - two concerts, Wednesday and Thursday

concerts

- 2015: Transitions (24.8)
  - outdoors, end of Summer concert
  - two days of setup, calibration and rehearsals
  - two concerts, Wednesday and Thursday
  - it (almost) never rains during Summer

concerts

- 2015: Transitions (24.8) – Wednesday...

concerts

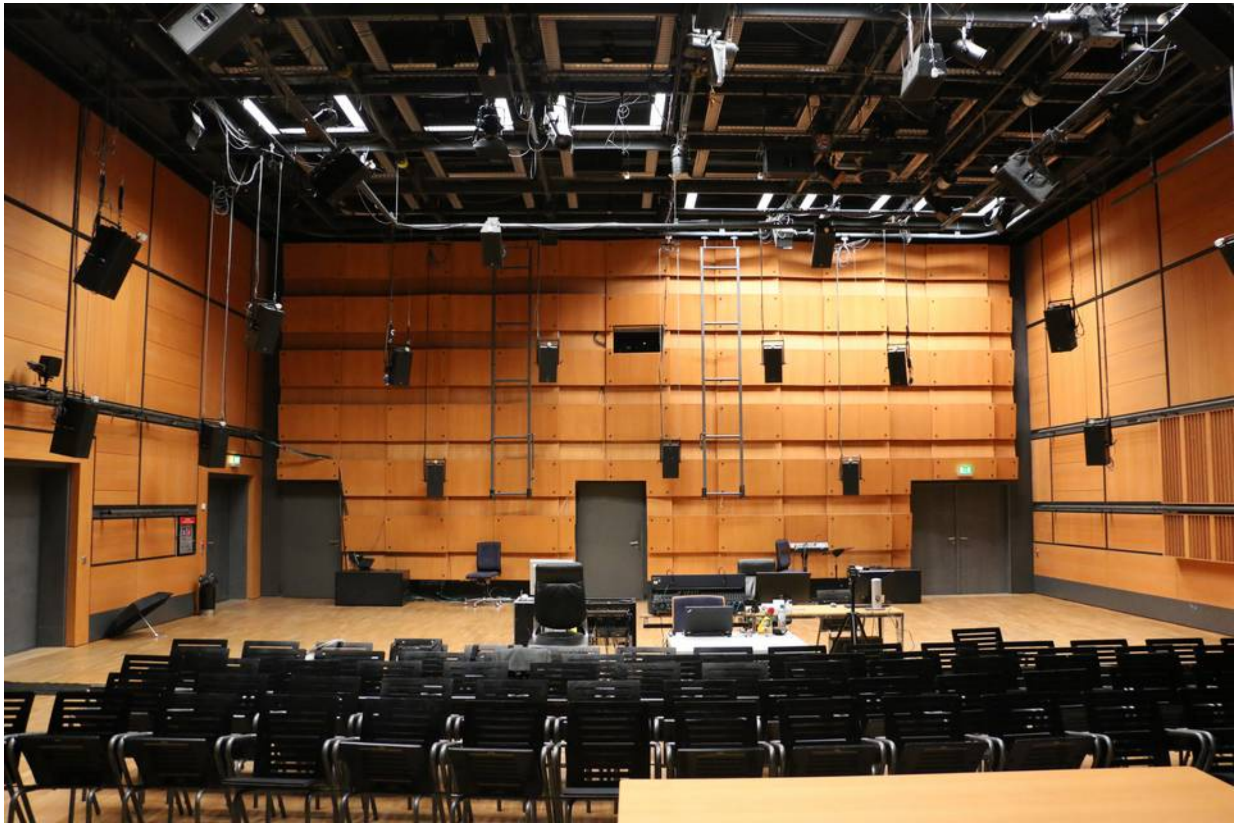
- 2015: Transitions -> Precipitations (16.8)

concerts

- 2015: Transitions -> Precipitations (16.8)
  - 16 speakers (8 + 6 + 2) instead of 24 (16 + 8)
  - ambisonics pieces: just play them with the new decoders
  - direct-to-speaker pieces, remap channels, make compromises, remix, all in just one day

pieces

- 2014: ZKM Kubus
  - 43 speakers in 5 layers



pieces

- 2014: ZKM Kubus -> Hagia Sophia

pieces

- 2014: ZKM Kubus -> Hagia Sophia
  - Space S[acred|ecular]
    - fixed media piece (CLM+Scheme)
    - 3rd order ambisonics (16 channels)
    - uses a new reverberation architecture (ambisonics in, ambisonics out, uses impulse responses from the Icons of Sound project for Hagia Sophia)

pieces

- 2014: ZKM Kubus -> Hagia Sophia
  - Space S[acred|ecular]
    - we have only two speakers
    - ardour session -> ambdec -> speakers (with a stereo decoder, Blumlein pair)

questions?

questions?

thanks!