AMBI@Home - The search for extra-frontal intelligence

Linux Audio Conference 2008, Kunsthochschule für Medien, Köln

How to build an Ambisonic listening rig at home using a Linux Audio workstation with JACK, COTS speakers/amps and digital room correction.

Jörn Nettingsmeier, freelance audio/event engineer
<nettings@stackingdwarves.net>

AMBI@Home Overview

1. *A short introduction to Ambisonics*
2. *System components*
3. *Basic setup*
4. *Applying digital room correction*
5. *Conclusion*

1. A short introduction to Ambisonics

1. ***What is Ambisonics? Why should I use it?***
2. ***How do we localize sounds?***
3. ***Virtual sound sources***
4. ***Near-field effect***
5. ***Speaker requirements***

1.1 What is Ambisonics?

Ambisonics is a technique for 3D surround sound.

It was developed in the 1970's at the BBC by Michael A. Gerzon and others.

Since all core patents have now expired, it is free to be implemented and used by anyone.

Powerful free software tools are available for production and playback.

1.1 What is Ambisonics?

Ambisonics is *homogeneous* – all directions are equal, no focus on a frontal soundstage.

Ambisonics *scales*: an Ambi signal can be played back in mono, stereo, horizontal or with-height surround.

**Ambisonic content is *independent of your speaker layout* – one format feeds many configurations.
(Think spouse acceptance factor!)**

1.1 What is Ambisonics?

The soundfield is sampled using *spherical harmonics* (~ microphone polar patterns).

First order uses one *omni* and three *figures-of-eight*.
(Think M-S Stereo in 3-D.)

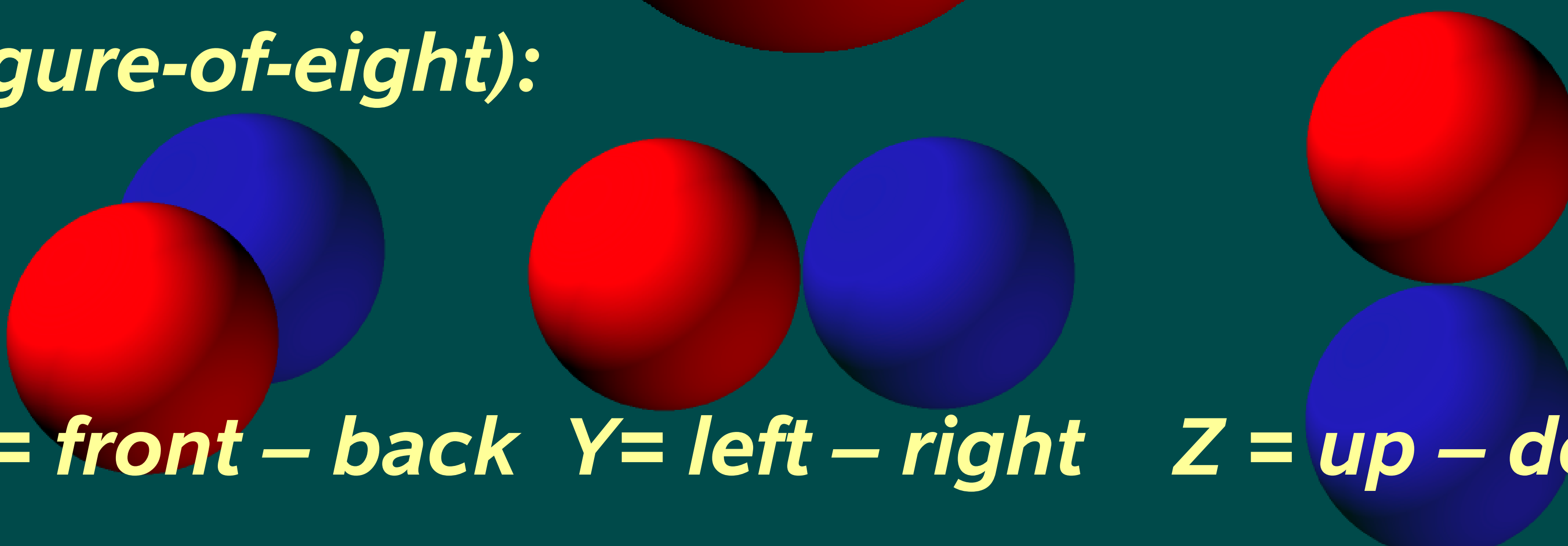
The resulting 4-channel signal is called *B-format* and contains *all information* about a soundfield *in one spot*.

1.1 What is Ambisonics?

Zero'th order (omni):

W = mono

First order (figure-of-eight):



X = front – back Y= left – right Z = up – down

1.1 What is Ambisonics?

B-format can be replayed on almost arbitrary speaker sets.

The trivial approach is to use a *virtual* cardioid microphone in the place of each speaker.
It “records” the subset of the entire B-format that will become the speaker feed.
This means you can optimise the decoder for your configuration.

1.1 What is Ambisonics?

For example, imagine a speaker directly in front (ϕ=0°), at ear level (ε=0°).

At that position, the speaker can give you neither left/right nor up/down cues.

Hence, its feed contains no Y and Z components! Instead, it will be something like

$$S = i * W + k * X,$$

where *k* and *i* depend on your speaker layout.

1.1 What is Ambisonics?

In practice, it's not quite that easy...

1.1 What is Ambisonics?

The set of all the weighing coefficients of X,Y,Z and W for all speakers is called the *decoding matrix*.
It represents the layout of your speakers, and can contain tweaks to the sound balance (for instance, a *forward dominance* might be used to emphasize a frontal sound stage over room effects).

1.2 How do we localize sounds?

At low frequencies (= long waves) we can only use phase differences between the ears as direction cues.

The head is no obstacle for long waves. So there are almost no shading or coloration effects to help us pinpoint a sound source.

1.2 How do we localize sounds?

As the wavelength approaches twice the ear distance, phase cues become confusing (we can't tell if just one or several periods “fit” between the ears).

So the brain ignores phase and uses loudness differences instead.

This works because at higher frequencies, our head creates an acoustic “shadow” at the side opposite of the sound source.

1.2 How do we localize sounds?

The "cross-over" frequency between those two basic mechanisms is around 700Hz.

Good ambisonic decoders (a.k.a. "classical") use two different decoding matrices that optimize for phase information (= *velocity vector, max r_V*) at low frequencies and loudness (= *energy vector, max r_E*) at higher frequencies.

1.2 How do we localize sounds?

Two-band decoding gives vastly better results in a home environment, but:

- **it requires phase-matched crossover filters, which are hard to get right**
- **the decoding matrix is now vastly more complicated than with the naïve “virtual mike” approach.**

1.3 Virtual sound sources

When testing an Ambi setup, it's important to know that

- ***all* speakers will contribute to any one virtual sound source,**
- **sources will generally have the same "sharpness" everywhere: better than 5.1 between speakers, not quite as good on speaker positions,**
- **ambisonic panning is a lot easier and smoother than pair-wise panning.**

1.3 Virtual sound sources

Don't get confused by meter readings. The speaker feed meters will give you no useful clue about the location of a source.

And even the B-format meters will only tell you if a signal is outside the corresponding plane, but not to which side.

1.4 Near-field effect

Basic Ambisonics assumes that all incoming sounds are plane waves.

It also assumes that speakers reproduce plane waves.

Of course, neither is true.

1.4 Near-field effect

Sounds are plane waves only if the distance is infinite.

When the distance is small compared to the frequency, the wave fronts are curved.

This curvature leads to a bass boost.

(Same thing as the proximity effect on directional microphones.)

1.4 Near-field effect

In first order, the effect is somewhat negligible at least for pop or chamber music.

But if you care about organs or good orchestra sound, your decoder must compensate for the near-field effect.

Unfortunately, this is also non-trivial, but there's software out there that gets it right.

1.5 Speaker requirements

All speaker chains must have matching levels and phase responses for good results.

That means you should not mix different brands or models.

It might be possible to compensate for that using DRC, but it's difficult.

1.5 Speaker requirements

If you want to use two-band operation (you do!), your speaker layout must be regular.

Make it a *regular polygon* (or polyhedron for full 3-D) where each speaker has a *direct opposite* (square, hexagon, but not pentagon; cube, octahedron, dodecahedron...).

If you can't make it regular, you have a problem:

1.5 Speaker requirements

Only Fons knows how to do optimization of irrgular layouts.

1.5 Speaker requirements

Fons likes exquisite red wines.

2. System Components

1. *Hardware*
   1. *Computer*
   2. *Audio Interface*
   3. *Speakers + Amps*
2. *Software*
   1. *JACK*
   2. *AmbDec*
   3. *DRC*
   4. *other useful stuff*

2.1.1 Computer

The decoding operation itself is trivial and does not create significant CPU load.

But digital room however works with convolution. The larger and more reverberant your room, the more cycles you will need.

However, the real-time correction of six speaker feeds never exceeds 20% of CPU time on an Athlon64 @ 2.4Ghz.

So a passively-cooled shoebox PC is fine.

2.1.2 Audio Interface

You will need as many channels of D/A out as you have speakers (there are ways to cheat, but not when you want to use DRC!).

I'm happy with a used RME 9652 from ebay and a Behringer ADA 8000. Don't take that one on tour, though.

2.1.3 Speakers + Amps

I don't know about cheap multichannel amps.

If you want to use more channels than a used 7.1 amplifier can drive, go for active studio monitors.

I'm using six Tannoy 5a. Generally, you get what you pay for.

I didn't pay much.

2.1.3 Speakers + Amps

Again: use the same model of speaker and the same type of amp for all channels.

Or be prepared to measure phase responses and to spent a looong time experimenting.

If you get good results with mixed and matched cheap gear, let me know!

2.1.3 Speakers + Amps

You will also need an instrumentation microphone. I'm using a Behringer ECM8000.

2.2.1 JACK

Of course, you need the JACK Audio Connection Kit.

Use QJackCtrl, there will be many virtual patch cables!

You can increase the buffer size to save CPU, since latency is not important for playback.

2.2.2 AmbDec

Fons Adriaensen's AmbDec is a state-of-the-art Ambisonic Decoder.

- **up to 24 speakers**
- **two-band operation**
- **first and second order**
- **near-field and distance compensation**
- **a nice GUI**

2.2.3 DRC

DRC (by Dennis Sbragion) is a toolkit to compute room correction filters.

You feed it an impulse response of the speaker (in the room), and it will try to

- figure out what's correctable at all,
- ignore the hopeless problems,
- create an inverse filter to undo room and speaker deficiencies.

2.2.4 other useful stuff

To obtain room IRs, try out Aliki.

For quick steady-state noise measurements, I use JAPA.

To actually apply the filter kernels created by DRC, a good choice is JConv.

3. Basic Setup

1. ***Define the listening position***
2. ***Determine possible speaker locations***
3. ***Fine-tune the speaker positions***
4. ***Wiring***
5. ***Enter the decoding matrix***
6. ***Testing***

3.1 Define the listening position

At first, assume that the sweet spot won't be much larger than your head.

Place a comfortable chair in the center of your planned setup. Avoid high backrests that could shade sounds from the rear.

If you can't make your setup perfectly regular, strive for perfect angles at the cost of slightly irregular distances.

3.2 Determine speaker locations

Usually, best practice suggests to keep the speakers away from the walls, so that the indirect sound will be heard as a (natural) early reflection, not as a (sound-imparing) comb filter effect.

With DRC, it's actually better to stick to the walls – earlier reflections can be corrected more easily and at less CPU load.

3.3 Fine-tune speaker positions

Put a mic stand or a camera tripod in your sweet spot.
Use a laser angle gauge and range finder, or a piece of string, adhesive tape and a self-printed cardboard angle gauge.

Check if the speaker positions are roughly equal. Move the sweetspot slightly if it helps.

3.3 Fine-tune speaker positions

Line the speakers up exactly (a few centimeters do matter!). Even if that's not possible, try to measure the actual location precisely (distance, azimuth, elevation).

3.4 Wiring

Well, plug in the cables, switch on the gear.

3.5 Enter the decoding matrix

AmbDec - Configuration

Load | Save | New | Cancel | Apply — myhexagon-drc.ambdec

Description: hexagon, max rV at LF, max rE at HF

Matrix scaling	Input scaling	Speaker distance	Near-field comp.	Crossover freq.
Normalised	Normalised	Delay comp.	None	600
Furse-Malham	Furse-Malham	Gain comp.	On inputs	
			On outputs	

Speakers				LF Decoder			HF Decoder		
				G(order)	0.3535	0.3535	G(order)	0.3535	0.2500
ID	Dist	Azim	Elev	W	X	Y	W	X	Y
F	1.59	0.0	0.0	1.0000	1.4142	0.0000	1.0000	1.4142	0.0000
RF	1.32	-60.0	0.0	1.0000	0.7071	-1.2247	1.0000	0.7071	-1.2247
RB	1.38	-120.0	0.0	1.0000	-0.7071	-1.2247	1.0000	-0.7071	-1.2247
B	1.55	180.0	0.0	1.0000	-1.4142	0.0000	1.0000	-1.4142	0.0000
LB	1.38	120.0	0.0	1.0000	-0.7071	1.2247	1.0000	-0.7071	1.2247
LF	1.62	55.0	0.0	1.0000	0.7071	1.2247	1.0000	0.7071	1.2247

Speakers | Decoder

3.5 Enter the decoding matrix

Speakers

ID	Dist	Azim	Elev
F	1.59	0.0	0.0
RF	1.32	-60.0	0.0
RB	1.38	-120.0	0.0
B	1.55	180.0	0.0
LB	1.38	120.0	0.0
LF	1.62	55.0	0.0

3.6 Testing

AmbDec has a built-in test signal – use it to verify your speakers are connected as you think they are. It's really hard to pinpoint even simple mistakes by ear in any non-trivial ambisonic system.

3.6 Testing

Use an ambisonic test signal with sounds from different directions as a quick plausibility check.

Then take a monophonic signal and move it around with an Ambi panner (either from the CMT or AMB plugin collections).

3.6 Testing

While testing, try to match speaker gains as closely as possible.

4. Applying Digital Room Correction

1. ***Measuring IRs***
2. ***DRC***
3. ***Convolution***
4. ***DRC optimization***

4.1 Measuring IRs

The easiest and most precise way to obtain good IRs *and* check for speaker distortion at the same time is Farina's *swept sine technique*, as implemented in the Aliki software.

See Fons' 2006 LAC paper.

4.1 Measuring IRs

Decide whether to measure the speakers *in the room* or in *free field*.

In-the-room allows you to correct for both speaker and room problems in one step.

A separate free field measurement would allow you to have more precise speaker correction, and then tackle the room in an extra step.

4.1 Measuring IRs

Place your instrumentation microphone in the sweet spot, at ear level.

No omni is perfect due to the shielding effect of the case.
Point your mike straight upward to get perfect response at least in the horizontal plane.

4.1 DRC

DRC computes correction filters from the IRs by applying the following steps:

- **trim (unfortunate - we lose information about the speaker distance, but AmbDec can correct that later)**
- **normalize (also unfortunate – we need to to manual gain matching later).**
- **remove nulls and deep troughs (uncorrectable)**
- **compute an inverse filter kernel**

4.1 DRC

In DRC, edit the desired frequency response to roll off where your speakers do the same.

Ported speakers should never be pushed below their port frequency – they will create distortion and a lot of heat for no gain.

Speakers with infinite baffle can go quite low, but don't overdo it.

4.2 Convolution

The filter kernels provided by DRC must be convolved onto your speaker feeds (i.e. the outputs of AmbDec) in real time.

I recommend Jconv, but any JACKable convolver will do.

4.3 DRC optimization

DRC has a number of tweakable parameters: various target responses, and several values dealing with the amount of correction.

Overcorrecting is evil.

The better (=more aggressively) you correct in the sweet spot, the worse it will sound everywhere else.

Wild overcompensation can damage your speakers.

5. Conclusion

At least for my setup, the benefit of using DRC was significant.

The frequency response was greatly improved, and the ambisonic localization was more focused, over a wider sweet spot.

DRC can help reduce room reverberation at low frequencies, making the bass a lot tighter.

5. Conclusion

No amount of DRC will un-distort speakers or turn garbage into gold.

The better the hardware, the better the results.

Still, it's nice to see what can be done with cheap gear.

5. Conclusion

I would like to try building larger Ambi rigs out of really cheap speakers (like gamer's 5.1 or 7.1 sets) and see how it turns out.

Can we do a sub-1000$ full 3-D rig for Joe User that doesn't suck?
If so, world domination would be imminent.

If you try, let me know.

Thanks

Thanks to all Ambisonic researchers whose papers are out there, thanks to the (unknown) LAC reviewers for valuable feedback, and also to Richard Lee, Eric Benjamin, Martin Leese and Fons Adriaensen.

Thanks

Thanks to you for staying until now.

Questions?