Future-proof surround sound mixing using Ambisonics and Reaper

Education Material and a Practical Workflow for Ambisonic Production and Publishing

Dr Bruce Wiggins
b.j.wiggins@derby.ac.uk
http://www.derby.ac.uk/staff-search/dr-bruce-wiggins
http://www.BruceWiggins.co.uk
Topics for Today

• Ambisonics
  – Describing what it is.
  – Describing why you want it.
  • Decoding
    – Surround
      » Regular
      » Higher Order
      » Irregular
    – Stereo
    – Mono!

• Practical Considerations:
  – Mixing/Production
  • Reaper

• Demonstrations
Teaching and Using Ambisonics

• For a number of years, I’ve been using Ambisonics with:
  – Both BSc and BA students
    • Some interested in the technology/theory behind Ambisonics
    • Some interested in the results promised by such a technology.
    • Some with a good background in Maths/Electronics etc.
    • Some with a good background in Music/composition etc.
The Challenge

• Allowing students from multiple disciplines to use Ambisonics in their work
• Giving enough understanding of the system so correct setup & routing occurs and informed experimentation can take place.
  – Few books cover Ambisonics
  – Papers on the subject can be mathematically heavy going.
• Allow them to be able to distribute that work.
How to explain Ambisonics?

- Ambisonic encoding is based on Spherical Harmonic Decomposition of a sound field...

\[
\begin{align*}
\left( \Delta - \frac{1}{c^2} \frac{\delta^2}{\delta t^2} \right) \phi (\vec{r}, t) &= -q (\vec{r}, t) \\
(\alpha \phi) &= N_{mn} \cdot P_{mn} (\sin \phi) \begin{cases} 
\cos (n\theta) & \text{for } \sigma = 1 \\
\sin (n\theta) & \text{for } \sigma = -1 
\end{cases}
\end{align*}
\]

\[
p (\vec{r}) = \sum_{m=0}^{\infty} (2m+1) j^m j_m (kr) \sum_{0 \leq n \leq m, \sigma = \pm 1} B_{mn}^\sigma Y_{mn}^\sigma (\theta_r, \phi_r)
\]

\[
N_{mn} = \sqrt{\frac{(m-n)!}{(m+n)!}} \quad c_0 = 1
\]

\[
C = \begin{pmatrix}
Y_{00}^1 (\theta_1, \phi_1) & Y_{00}^1 (\theta_2, \phi_2) & \cdots & Y_{00}^1 (\theta_j, \phi_j) & \cdots \\
Y_{01}^1 (\theta_1, \phi_1) & Y_{01}^1 (\theta_2, \phi_2) & \cdots & Y_{01}^1 (\theta_j, \phi_j) & \cdots \\
Y_{10}^1 (\theta_1, \phi_1) & Y_{10}^1 (\theta_2, \phi_2) & \cdots & Y_{10}^1 (\theta_j, \phi_j) & \cdots \\
Y_{11}^1 (\theta_1, \phi_1) & Y_{11}^1 (\theta_2, \phi_2) & \cdots & Y_{11}^1 (\theta_j, \phi_j) & \cdots \\
\vdots & \vdots & \ddots & \vdots & \ddots \\
Y_{M0}^1 (\theta_1, \phi_1) & Y_{M0}^1 (\theta_2, \phi_2) & \cdots & Y_{M0}^1 (\theta_j, \phi_j) & \cdots \\
1 & Y_{M1}^1 (\theta_1, \phi_1) & \cdots & Y_{M1}^1 (\theta_j, \phi_j) & \cdots \\
\end{pmatrix}
\]
Combining Microphone Patterns

- It’s possible to create any microphone pattern between an omni-directional and a figure of 8, by x-fading between the two...

Omni
\[ r(\theta) = 1 \]

Figure of 8
\[ r(\theta) = \cos(\theta) \]

Mixing equation
\[ r(\theta) = 0.5 \times ((2-d) \times \text{omni} + (d \times \text{f8})) \]
Combining Figure of Eights

- Adding together two perpendicular figure of eight microphones will result in a figure of eight microphone positioned between them:

\[ r(\theta) = \cos(\theta) \quad \text{and} \quad r(\theta) = \sin(\theta) \]

\[ 0.707 \times (\cos(\theta) + \sin(\theta)) \]
The SoundField Microphone

The SoundField microphone uses the two principles discussed above:
– Omni + Figure of 8
– Adding Two figure of 8’s
in order to produce any first order polar response in any direction using just one microphone.

Picture of a SoundField ST250
The SoundField Microphone

• First the three figure of 8 microphones are summed to produce a figure of eight pointing in any direction.

• Then this is mixed with the omni-directional mic to produce a variable mic pattern in this direction.
How do we choose some mics?

• Given a mic system where we can generate mic signals that are:
  – Pointing in any direction we want
  – Using whatever polar pattern we wish…
• …how do we decide what we use for:
What is ‘Ambisonics’?

• What is Ambisonics?
  – A system where the spatial encoding of the audio is separate from the decoding of that audio to a set of speakers.
  – A set of rules and equations that help to optimise and quantify the performance of a multi-speaker audio presentation.
  – A system based on coincident recording principles.
  – An extendable system based on Spherical Harmonics (mic patterns!).
  – A future proof audio format.
Why should you want it?

• Can be successfully decoded to many different speaker arrays:
  – Mono
    • Mono compatibility is still important!
    • Just use W, omni feed!
  – Stereo
    • UHJ allows for stereo compatible downmix
    • Preserves level balance between sources
    • Hard panning (only amplitude) is avoided, making headphone listening more natural
Why should you want it?

- Decode to standard surround sound arrays (we want people to hear this stuff!):
  - 4.0
  - 5.1
  - 7.1
  - x.x
- Decode to larger, arbitrary speaker arrays
  - Better for large events (use more speakers, rather than bigger speakers!)
$W = 1/\sqrt{2}$

$X = \cos(\theta)$

$Y = \sin(\theta)$

$g_w = \sqrt{2}$

$g_x = \cos(\Phi)$

$g_y = \sin(\Phi)$

$S = 0.5 \times [(2 - d)g_wW + d(g_x X + g_y Y)]$

$\theta$ is desired source angle

$\Phi$ is speaker angle

$S$ is speakers output
Higher Order Ambisonics

- Uses more input signals…
- …which can result in better control of the speaker feeds and, hence, reproduced sound field.

- **0\textsuperscript{th} Order**
- **1\textsuperscript{st} Order**
- **2\textsuperscript{nd}**
Polar Pattern Choice
Visualisations

Web Versions

Ambisonic Signals
http://www.brucewiggins.co.uk/AmbiVis/Spherical

1st, 2nd & 3rd Order
Regular Decode
http://www.brucewiggins.co.uk/AmbiVis/Regular

Why not 3rd Order
All the time???
http://www.brucewiggins.co.uk/AmbiVis/Detent

3rd Order
Irregular Decode
http://www.brucewiggins.co.uk/AmbiVis/Irregular
Decoders for irregular speaker arrays

- Irregular decoders cannot be optimised so easily.
- For left/right symmetrical systems:
  - Amplitude
  - Polar pattern
  - Angular spread
- Must all be optimised, per speaker pair.
  - This means solving a set of non-linear simultaneous equations.
  - Heuristic methods can be used to solve this problem.
Higher Order Irregular Decoders

- Higher Order Components give two main benefits:
  - More focused polar patterns
    - Gives more ‘focused’ localisation
    - Needs more speakers
  - Allows asymmetrical decode patterns to be used
    - Ideal for irregular speaker arrays (ITU….)
Higher Order Decoders

2\textsuperscript{nd} Order Decoder 1

4\textsuperscript{th} Order Decoder 1

2\textsuperscript{nd} Order Decoder 2

4\textsuperscript{th} Order Decoder 2
Energy/Velocity is Not the Only Way!

- **LF Time Difference**: 0 degrees
  - **Source/Decoded Source Angle**
  - **Amplitude**

- **HF Amplitude Difference**
  - **Source/Decoded Source Angle**

- **4th Order Decoder 1**
  - **G Format**
  - **Real Source**

- **4th Order Decoder 2**
  - **G Format**
  - **Real Source**

Dr Bruce Wiggins
Visualisations

Web Versions

Ambisonic Signals

http://www.brucewiggins.co.uk/AmbiVis/Spherical

1\textsuperscript{st}, 2\textsuperscript{nd} & 3\textsuperscript{rd} Order
Regular Decode

http://www.brucewiggins.co.uk/AmbiVis/Regular

Why not 3\textsuperscript{rd} Order
All the time???

http://www.brucewiggins.co.uk/AmbiVis/Detent

3\textsuperscript{rd} Order
Irregular Decode

http://www.brucewiggins.co.uk/AmbiVis/Irregular
Practical Ambisonics

- Example Routing for 1st Order Ambisonics.

Diagram:
- Audio Track 1 and Audio Track 2 input into B-Format Encoders.
- Outputs from B-Format Encoders feed into 'Dry' 4-channel Bus.
- 'Dry' 4-channel Bus outputs to 'Wet' 4-channel Bus.
- 'Wet' 4-channel Bus feeds into 4 Channel Ambisonic Reverb.
- Outputs from the 4 Channel Ambisonic Reverb are split to W Bus, X Bus, Y Bus, and Z Bus.
- W Bus, X Bus, Y Bus, and Z Bus are connected to Ambisonic Decoder.
- Ambisonic Decoder outputs to Speaker 1, Speaker 2, Speaker 3, Speaker 4, Speaker 5, and Speaker 6.
Practical Ambisonics

• Things to note:
  – Channel Counts can get high:
    • 1\textsuperscript{st} order = 4
    • 2\textsuperscript{nd} order = 9
    • 3\textsuperscript{rd} order = 16
    • N\textsuperscript{th} order = (N+1)^2
  – The channels don’t represent speaker feeds
  – The number of channels per track can change throughout the project
  – We might not want to route all channels in one track to all the channels in another track:
• Most DAWs don’t like the above very much!
• Many Music Producers don’t like using modular hosts!
Reaper

• Reaper is great for Ambisonics:
  – It’s amazingly flexible
  – Up to 64 channels per track
  – Tracks aren’t setup by speaker arrangement
  – Routing anything to anywhere is possible…
  – …with or without hardware outputs
  – Multi-channel plug-ins are supported…on any track of any channel count!
  – It’s really good value for money

  • “REAPER requires no dongle, has no copy protection, and can be evaluated with full functionality.”
Reaper
Built in Plug-in Engine (Jesuasonic)
Supports OSC & Python/Perl Scripts

- OSC can be used to automate any action via a network.
- Reascript allows for scripting of any reaper commands or APIs via Python and Perl.
  - Macros
  - Complete Automation

Picture of TouchOSC from [http://hexler.net/](http://hexler.net/)
Multi-Channel File Support

• It will read and write multi-channel:
  – Wave Files
  – FLAC
  – WavePac (supports more channels than FLAC)
  – Ogg Vorbis
  – And others….

• Any tracks input or output can be rendered to a multi-channel file.
• Hierarchical (parent/child) Tracks make routing straight forward
  – Make a multi-channel, B-Format Bus, the parent
  – All multi-channel child tracks are, summed to that bus.
WigWare Plug-ins

- 1\textsuperscript{st}, 2\textsuperscript{nd} & 3\textsuperscript{rd} order regular decoders
  - Include NFC and speaker distance compensation
- 1\textsuperscript{st}, 2\textsuperscript{nd}, 3\textsuperscript{rd} & 4\textsuperscript{th} order irregular ITU 5.0 decoders
  - Multiple solutions per order
  - Includes NFC and speaker distance compensation.
- 1\textsuperscript{st}, 2\textsuperscript{nd}, 3\textsuperscript{rd} & 4\textsuperscript{th} order panners
  - Polar or XY interface
  - W panning, or NFC and Distance Filtering
- 1\textsuperscript{st} Order 3D Reverb

- Planned
  - UHJ Encoder/Decoder
  - B-Format Manipulations
    - Rotate/Tumble/Dominance etc.
  - 3D Delay
Example Configuration
Demonstrations