Crafting Sound in Space:
Working with Ambisonics using blue and Csound

Jan Jacob Hofmann
jjh@sonicarchitecture.de

Steven Yi
stevenyi@gmail.com

Abstract

This presentation/paper will briefly discuss Ambisonics, analyze requirements for working with Ambisonics for composition, and will discuss different strategies of working with blue and Csound to compose using Ambisonics for spatialisation of sound.

Keywords

ambisonics, csound, blue

1 Introduction

This presentation/paper will briefly discuss Ambisonics, analyze requirements for working with Ambisonics in context of composition, and will discuss different strategies of working with blue and Csound to compose using Ambisonics for spatialisation of sound.

2 General introduction into the method of storing spatial information using Ambisonics

2.1 Characteristics

Ambisonics is one of the very few methods of spatial audio which supports periphony (meaning the inclusion of the height component of sound). While Ambisonics was originally developed as a means of recording and reproduction of sound, the techniques of Ambisonics are also usable for those interested in composing and creating new sound works. Some of the attractive qualities of Ambisonics for the composer are:

- The image is fairly stable and precise, independent from the position of the virtual sound (coming from the speaker or from a point in-between the speakers). This means that the sound does not change, as it moves around; thus the sound gets liberated from the speakers.
- The position of the listener is not that important for getting a true localisation as it is in most other surround-sound techniques. Even listeners far off the sweet spot still get a fairly good impression of direction.
- It can be combined with the distance-clues of the virtual sound source, resulting in sounds that are perceived as being closer or more distant to the listener.
- Once the sound is spatially encoded it can be decoded for performance to any desired speaker-setup in a versatile way, as long as it is symmetric.
- Ambisonics is free and efficient.

2.2 Storage Format (Encoding/Decoding)

Ambisonics is a method of storing the position of the maximum of the sound energy in files which relate to different directions in space. Thus both, the energy and the velocity of a particle of air affected by a sound wave can be stored. A moving wavefront trough the listening space can be stored and reproduced. The encoded files then have to be decoded in a second step according to the actual positions of the speakers, which then recreate the wavefront together.

The higher the order, the higher is the accuracy with which the spatial information is stored. The higher the number of speakers, the higher is the accuracy with which the stored spatial information is reproduced.

- **First Order Ambisonic:** Encoding into **4** audio files (W, X, Y, Z)

  Minimum recommended number of speakers for a periphonic decode: **8** (theoretically: 5)

- **Second Order Ambisonic:** Encoding into **9** audio files (W, X, Y, Z, R, S, T, U, V)

  Minimum recommended number of speakers for a periphonic decode: **12** (theoretically: 10)

- **Third Order Ambisonic:** Encoding into **16** audio files (W, X, Y, Z, R, S, T, U, V, L, M, N, O, P, Q)

  Minimum recommended number of speakers for a periphonic decode: **20** (theoretically:17)

- **Fourth Order Ambisonic:** ...etc.

The different data streams for Ambisonics are generally stored either as separate files or as a single multi-channel file (WAV/AIFF/SDII/etc.). Each stream relates to determined directions in space.

Note: The Ambisonic format only yields the direction of the sound – the distance information has to be added by other means.

3 Coding Ambisonics

To work with Ambisonics, we will take a signal and specify its angle and elevation from the center location. Then we feed this into some equations for Ambisonics to encode that signal to create the x number of signals required for the order of Ambisonic we would like to use.

For the purpose of this article, we will use the Furse-Malham Set of 2nd order Ambisonic equations, which are related to the formal complex set (defined in terms of associated Legrendre polynomials) of spherical harmonics:

W = input signal * 0.707

X = input signal * cos(A) * cos(E)

Y = input signal * sin(A) * cos(E)

Z = input signal * sin(E)

R = input signal * 1.5sin(E) * sin(E) - 0.5

S = input signal * cos(A) * sin(2E)

T = input signal * sin(A) * sin(2E)

U = input signal * cos(2A) * cos(E) * cos(E)

V = input signal * sin(2A) * cos(E) * cos(E)

where A is equal to the angle of the signal's position and E is equal to the elevation of the signal's position. This 2nd order processing of a signal has been encapsulated in the Csound opcode **bformenc1**:

aw, ax, ay, az, ar, as, at, au, av **bformenc1** asig, kalpha, kbeta

where asig is the audio signal, kalpha is the angle of the sound''s position, kbeta is the elevation of the sound's position.

3.1 Assignment of a position to a sound in Cartesian or Polar coordinates

A (angle) and E (elevation) data has to be generated to feed the equation for the Ambisonic encoding. Also Cartesian coordinates may be used for special purposes and then be transformed to polar coordinate system, before fed into the equation or opcode.

4 Enhancing Ambisonics

Because the Furse Malham Set of equations works for practical reasons only to encode the direction of sound, adding further processing to help determine the distance of the sound as well can further enhance the spatialisation.

The most common distance clues used are:

1. Attenuation
2. Filtering
3. Local Reverb
4. Global Reverb
5. Early Reflections

For the distance-perception of the sound and depth of a sonic environment, several distance clues have to be added to the sound. These distance clues can be well combined with the Ambisonic method, so that a whole sonic environment with directional and distance information can be archived.

The sound is modified by attenuation, filtering and local reverb. The modified sound is then sent to the Ambisonic equation to store it along with its directional information.

Global reverb and early reflections can be distributed themselves independently of the position of the original sound source by the Ambisonic method. Thus the original source is sent to an independent device (instrument), which calculates the appropriate position of the reflections in respect to the position of original sound source.

Each early reflection and a number of sources for decorrelated reverberation are spread in space. The Ambisonic encoding of this signal takes place in this separate device/instrument, but can be mixed into the final encode of all sources and sounds.

The instrument generating spatial sound gets more complicated as more gadgets for spatialisation are combined with Ambisonics. However it is highly recommended to use methods of enabling distance perception (attenuation, reverberation, early reflections, filtering) to have a convincing and stable spatial perception. This effort may be very CPU intensive and therefore spatialisation of many sources goes far beyond performance in real time.

5 Composing Strategies for Ambisonics

There are a number of ways we can use Csound for composing with Ambisonics. The methods that are used will largely depend on the nature of the musical material we will be working with and how we want to specify their spatial qualities. At a high level, we will need to determine whether our material has a constant position or changing position.

5.1 Constant Position

In a constant position setting of sound, each sound is assigned to a fixed position that is kept during total duration of that sound. No temporal information is assigned. A real world example of this would be a guitar player that has been recorded in a single location. This resultant sound would be the source sound of the guitar player as well as the sound of the room the player was in; for example, the permanent distribution of twelve or more decorrelated reverberations of the same source at constant locations in space.

5.2 Changing Position (movement)

In a changing position setting of sound, a single or compound trajectory (many trajectories linked) has to be created. The information of the trajectory yields data about the position of the sound over time. The resultant sound would include the sound of the room as in the constant position situation, but the movement/speed of the sound also comes into play. The sound may be altered by a simulation of the Doppler effect.

6 Entering the Data of Position

In working with Csound, there are a number of ways in which one can specify positional data for an instrument:

1. Positional data is held within instrument code: sound location will be depend only on the defined constant values or code within the instrument itself. The spatial location of the instrument may be static for the duration of the piece or dynamic if the instrument is coded to generate control signals, though the dynamic shape will be the same for all instances of the instrument.
2. Positional data is passed in as p-field arguments to instrument: the sound location can be changed per note. The spatial location of the instrument may be static for the duration of the note or dynamic if the instrument is coded to generate control signals using the p-field arguments.
3. Positional information is generated as a control signal in one instrument and read in by another instrument. In this scenario, the control of spatial location then is a separate concern from the source signal generation.
4. Positional information is generated as a control signal external to Csound and read in by an instrument. The control signal may be communicated via MIDI, OSC, or through the Csound API directly from a host program to the Csound engine.

Using Csound alone, all of the first three methods above are possible. However, composing by text interface can be difficult as more sound sources are used, to the point of hindering the compositional process. (Personal experience has found that a text interface is sufficient for organizing approximately up to 20 sound sources where the spatial location is either constant or made up of simple trajectories.)

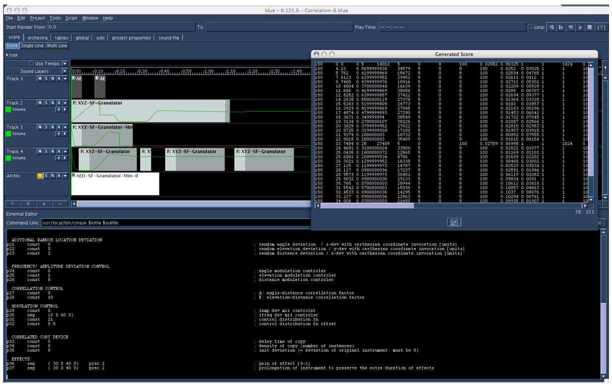
For organizing larger amounts of musical material, an external tool in conjunction with Csound can greatly improve the compositional experience. The software *blue* is a music composition environment built on top of Csound that provides a timeline, orchestra manager, and many other tools to aid in composing with Csound. It has the ability to accommodate the normal Csound text interface for when using that is most ideal, but also allows working with scripting internally and using external programs (i.e. CMask) to allow the user to develop routines to generate material (i.e. repeated patterns of music, algorithmically generated material, spatially-distributed stochastic distributions of sound).

Beyond these text-based possibilities, users can create their own GUI's to work with their scripts, as well as use other pre-made GUI objects (i.e. JMask, PianoRoll, Pattern Object) to generate their material. Also, to accommodate control of spatial location of an instrument separately from the instrument itself, control signals can be drawn in the blue interface using its parameter automation capabilities.

By using blue's feature on top of Csound, a number of convenient ways of working with Ambisonics has opened up, from simple working with material note-by-note all the way to the creation of spatial granular clouds.

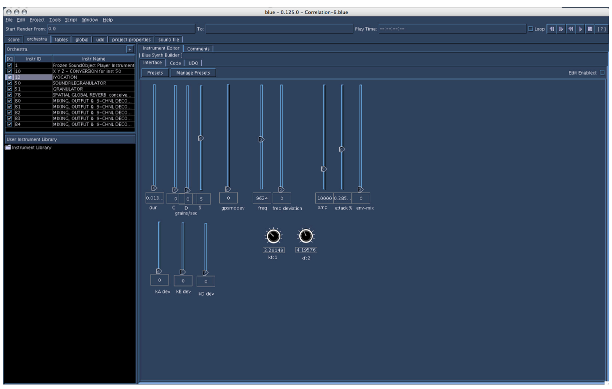
7 Examples

The following show examples of working with blue with material using Ambisonics.



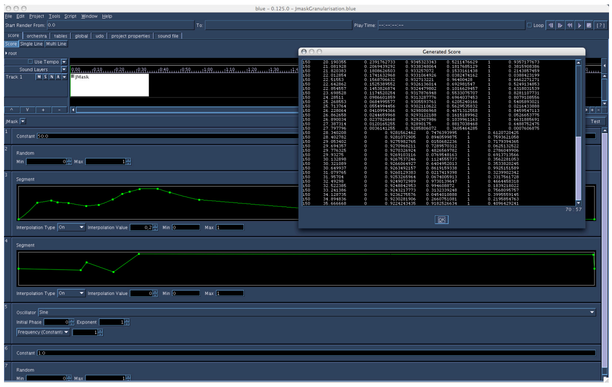
Example 1: CMask

Example 1 shows using Cmask within blue to create a spatialised granular cloud. Notes generated from CMask correspond to a single grain, and each grain has its own position information.



Example 2: UI Control of Position Data

Example 2 shows an instrument where position information is controlled by UI widgets. These widgets are themselves automatable on the score timeline as well as controllable in realtime while rendering with Csound for immediate audible feedback when enabling the use of the Csound API from blue. Use in realtime is dependent upon complexity of material due to CPU-intensive nature of Ambisonics with enhanced distance processing.



Example 3: JMask

Example 3 shows creating a spatialised granular cloud using the graphical JMask object. Like Example 1, notes generated from JMask correspond to a single grain, and each grain has its own position information.

8 Conclusion

Composing with Ambisonics is an effective means for locating the direction of sound in all directions. By enhancing Ambisonics with the use of distance clues, a high degree of control over material in space is possible. By using Csound and blue, we have a number of ways to then designate positional data for the musical material we work with.

The way of entering the information of the position of the sound has to be chosen appropriately to the task to be desired. The interface correlates with the compositional result intended. While for constant position or a limited number of sound-sources, text-based, graphical or even slider-based input of data is sufficient, these approaches fail as it comes to larger amounts of sound sources to be organized such as needed for spatial granular synthesis.

With the use of external stochastic organization of sound and position in blue, the limit of a few sources of sound no longer exists but an unlimited number of sound sources can be organized in space, creating the new experience of spatial granular synthesis. Now clouds and moving swarms of sound may be created conveniently.

By using spatial granular synthesis and correlating the parameters, lines or planes in space can be created. Also complex patterns of grains are possible. Unfortunately it is hard to perceive such organized distribution of grains in detail, as the spatial definition of human hearing perception is limited.

Still spatial granular synthesis may lead to impressions never heard before in electronic music if used with care and thought.

References

[1] Csound. http://www.csounds.com

[2] blue. http://www.csounds.com/stevenyi/blue

[3] Sonic Architecture site.
http://www.sonicarchitecture.de

[4] Spatial hearing mechanisms and sound reproduction by D.G. Malham, University of York, England 1998:
http://www.york.ac.uk/inst/mustech/
3d_audio/ambis2.htm

[5] First and Second Order Ambisonic Decoding Equations by Richard Furse 1999-2000
http://www.muse.demon.co.uk/
ref/speakers.html

[6] Spatial Sound Encoding Including Near Field Effect: Introducing Distance Coding Filters and a Viable, New Ambisonic Format
by Jérôme Daniel, AES 23rd International Conference, Copenhagen, Denmark, 2003 May 23-25
http://gyronymo.free.fr/index.html

[7] Higher order Ambisonic systems
(abstracted from "Space in Music - Music in Space", an Mphil thesis by Dave Malham, 2003)
http://www.york.ac.uk/inst/mustech/
3d_audio/higher_order_ambisonics.pdf