

Paper title:

Sound spatialisation using Ambisonic

Topics involved:

- Physical modeling and sound diffusion
- Digital audio processing

Paper category:

- Research paper

Author:

Oscar Pablo Di Liscia
Universidad Nacional de Quilmes
Email: odiliscia@unq.edu.ar

Sound spatialisation using Ambisonic

Oscar Pablo Di Liscia
odiliscia@unq.edu.ar
Carrera de Composición con Medios Electroacústicos
Universidad Nacional de Quilmes
Buenos Aires
Argentina

Abstract. The main resource to simulate location of a sound source is to try to fool our auditory system by creating phantom sources. Those are sounds that do not come from where the virtual source seems to be, but gives to our auditory system the impression that the sound source is located at some -real- point in the space.

The procedure more commonly involved in the simulation of phantom sources is referred to as intensity panning. Though widely used, intensity panning is at present being strongly criticized because of perceptual limitations.

Some British recording engineers -mainly Michael Gerzon- developed a technique called Ambisonic (which is a registered Trademark of Nimbus Communications International), widely used at present. Ambisonic attempts to overcome the limitations above mentioned by encoding the signal on the same way that a special microphone would record it. The decoding procedure attempts to recreate the wave front that the microphone “had listened” for a given array of loudspeakers.

The computer program WDSPA uses this technique -among others- to spatialise digital sound.

This paper discusses both the Ambisonic technique in comparison with intensity panning, and its implementation on the program WDSPA.

1. General Background

Both spatial listening, and the DSP techniques that entertain its simulation in electronic music, constitute huge subjects. To deal with them completely would go far beyond the aims of this paper. At present, fortunately, a big amount of high quality documented research on spatial listening can be found -among other- in (Blauert 1983; Haas 1951; Wallach 1973; Chowning 1992; Gardner 1969; and Hartmann 1983), as well as developments on traditional DSP methods for sound spatialisation and their programming implementation under several environments (Moore 1983, 1989, 1990; Chowning 1971; Lopez Lezcano 1992; Dodge & Jerse 1985; Bossi 1990; Kendall 1989; Moorer 1979; and Karpen 1998). The reader may consult -among others- the above mentioned works, in order to obtain the basic concepts needed for a sound comprehension of the following discussion.

2. Ambisonic versus Intensity Panning:

One of the most easy controllable method for the angular location of sound over loudspeakers, consists in the scaling of the energy emitted by them. The procedure most usually involved is widely known and referred to as *intensity panning* (See Moore 1990; Bossi 1990; Chowning 1971; and Dodge & Jerse 1985).

It is possible to calculate the amount of energy delivered by a sound source for a given spatial location, listener location, number of channels, location of the loudspeakers and directional characteristics of the source, to scale the gain of a signal at each output channel of a reproduction system.

To simulate a sound source located at any angle using Intensity Panning, the energy delivered by the sound source must be distributed between pairs of loudspeakers, and the sum of the signals emitted by both of them must always be the same for any angle value. To accomplish this, the following trigonometric identity may be used:

$$\text{Cos}(A)^2 + \text{Sin}(A)^2 = 1$$

Where A is any angle in radians.

We need know next how to use the above equation on a practical situation (i.e., which gain values we will obtain from it). Since energy equals to the square of amplitude, using the sine and cosine functions will produce the correct result in terms of amplitude.

As a simple example, assuming that we have two loudspeakers at angles of 0 and $\pi/2$ radians, then the gain for each channel will be:

$$\text{Ch1} = \text{Cos}(A)$$

$$\text{Ch2} = \text{Sin}(A)$$

Where A is any angle between 0 and $\pi/2$ radians.

Another procedure may be to calculate the energy and then obtain from it the correspondent amplitude for each channel taking its square root (Bossi 1990).

The above equation does not take in account neither the distance from the source to the listener, nor the directional characteristics of the source. Both the *distance scaling* and the *directional characteristics scaling* factors are, however, not exclusive of intensity panning, and may be used in other cases as the one I will deal with next (Ambisonic).

To take in account of distance, the equation must be restated as:

$$\text{Ch1} = \text{Cos}(A) / (\text{distance} + \textit{offset})$$

$$\text{Ch2} = \text{Sin}(A) / (\text{distance} + \textit{offset})$$

(the *offset* term may be necessary in order to prevent the cases in which the distance is less than the unit -specially when it becomes zero...-). Psychoacoustic research had shown that distance cues may be more perceptually effective using a different scaling. This being the case, a *scaling exponent* may be used to raise the distance factor at a power greater than one, thus producing a more pronounced ("exaggerated") gain curve as the source moves towards the listener:

$$\text{distance factor} = (\text{distance}^{\text{scaling exponent}} + \textit{offset})$$

Finally, the signal at both channels may be scaled by a *directional factor* derived from the directional characteristics of the sound source and its orientation (if this is not done, the source is considered omni directional...). We can find a useful approach to this in (Moore 1989). Moore proposes to model *radiation vectors*(with their magnitude determined by their

angle), and shows how a *supercardioid* shape is determined in two dimensions by the formula:

$$r(A) = (1 + (((back - 1) | R - A |) / \pi))^2$$

where R is the angle (in radians) of the radiation vector and *back* specifies the relative amount of radiation in the direction opposite to R . It can be seen that, setting *back* to zero will produce the cardioid shape, while setting it to one will produce an omni directional shape. A similar conception (though in a three dimension space) is found on the program *Vspace* (Furse 1999).

Another approach to the simulation of the directional characteristics of the sound sources can be found on the DirectSound System (which is a registered trademark of Microsoft). DirectSound attempts modeling sound sources in a three dimension space as if they were "sound cones" (Bargen & Donnelly 1998).

Though widely used, intensity panning is at present being strongly criticized because:

1-It is effective only for a small group of listeners located at the center of the listening room. This is a consequence of the so called "*Haas effect*" or "*precedence effect*" (see Haas 1951; Wallach 1973) by one hand, as well as of the "pulling effect" of the louder signal emitted by the nearby loudspeaker.

2-The signals coming from all the loudspeakers will reach both ears of the listener (on a very different way than the "real" sound source being emulated will do...), meanwhile rendering confusing information. The latter is often referred to as the *crossstalk* of the loudspeakers. An explanation and graphics of this effect can be found on (Ellen 1973).

3-Once a mix is done for a particular array of loudspeakers, it is not possible neither to reproduce it, nor to modify it for a different one (at least it *should* not be possible...).

Some British recording engineers -mainly Michael Gerzon- created a technique called *Ambisonic* (which is a registered Trademark of Nimbus Communications International), widely used at present. Ambisonic attempts to overcome the limitations above mentioned by encoding the signal on the same way that a special microphone would record it (as a matter of fact such microphones exist, and one of them is the *Calrec Soundfield* microphone). This encoding keeps the information of the energy delivered by a sound source located on a three dimension field using four signals (there are other encoding formats using more than four signals, but we will not deal with these now) as if it were recorded by an array of three figure of eight microphones (each one pointing to the three axes), plus an Omni-directional microphone. The decoding procedure attempts to recreate the wave front that the microphone "had listened" for a given array of loudspeakers (Ambisonic's specialist refers to that array as the *rig*). In Ambisonic, all the loudspeakers works together "pulling and pushing", and that is why it is not advisable to mix signals not encoded using Ambisonic with an encoded one. One of the main advantages of Ambisonic, however, is that if we have a signal properly encoded we may further decode it for the rig of loudspeakers we are to use in a particular situation.

3. The Ambisonic B Format

This format stores the directional information of the wave front encoded in four signals:

X= front / rear
Y= left / right

Z = above / below
 W = Omni directional information.

The encoding process may be accomplished, either recording sound with an Ambisonic microphone, or performing DSP operations on a signal stored on a sound file. The following encoding equations are used (Malham 1998):

$X = \cos(A) * \cos(B) * \text{input signal}$
 $Y = \sin(A) * \cos(B) * \text{input signal}$
 $Z = \sin(B) * \text{input signal}$
 $W = 0.707 * \text{input signal}$

Where: A is the horizontal (azimuth) angle, counterclockwise measured.
 B is the elevation angle

If the sound source is located at any point within the unit sphere, and therefore $(x^2 + y^2 + z^2)$ is always minor or equal than 1, then the encoding equations are simpler:

$X = x * \text{input signal}$
 $Y = y * \text{input signal}$
 $Z = z * \text{input signal}$
 $W = 0.707 * \text{input signal}$

Where x , y and z are rectangular coordinates indicating sound source position on a three dimension space. Malham (Malham 1998) points out that, if placement inside of the unit sphere is required, the levels of the signals X , Y and Z will reduce the total intensity of sound instead of increasing it, as expected. Therefore, he propose to make W to change according to:

$$W = 1 - 0.293(x^2 + y^2 + z^2)$$

Are we to use a computer program to encode a signal, the equations must take in account also of distance between source and microphone. For this purpose, Richard Furse state the following equations (Furse 1995):

$ds = x*x + y*y + z*z$
 $dist = \text{sqrt}(ds)$
 $X = x / (ds + \text{offset})$
 $Y = y / (ds + \text{offset})$
 $Z = z / (ds + \text{offset})$
 $W = .707 / (dist + \text{offset})$

Where *offset* is a quantity added to simulate the *core sphere radius* of the microphone (thus, used to avoid infinite gain values if distance equals 0). Furse (Furse 1995), suggest $\text{offset}=0.3$

It would be possible also to scale the W, X, Y and Z signals by a directional factor such as the one that was already described in the anterior section.

The decoding process is quite straightforward and elegant. As shown in (Bamford 1995), having a B-format signal encoded, the feed for a First Order N speaker Ambisonic System would be:

$$P_n = 1 / N (\mathbf{W} + 2\mathbf{X} \cos(\theta_n) + 2\mathbf{Y} \sin(\theta_n))$$

Where θ_n is the angle, in radians, of the N-th speaker of the system.

Here follows a C language function which calculate the gain values for encoding and decoding a signal for First Order B Format. This function expects two floats (x, y) indicating the rectangular coordinates of the source, and one (input) being the input signal. It calculates (i.e., encode the input) the X, Y and W signals, and writes the decoded result to an array of four floats, each one being a channel of a four-loudspeakers square rig. For simplicity reasons the z (height) dimension was disregarded in this example. Most used decoding matrixs can be found in (Furse 1999).

```
void gain_ambisonics(float x, float y, float z, float *ch[], float input)
{
    register float ds, dist;
    register float amb_x, amb_y, amb_w;
    /*AMBISONIC B ENCODING (Z signal ommited here...) */
    ds    = x*x + y*y;
    dist  = sqrt(ds);
    amb_y = input * (x / ds);
    amb_x = input * (y / ds);
    amb_w = input * (.707 / dist);
    /*AMBISONIC B DECODING (Z signal ommited here...) */
    ch[0]= (amb_w + amb_x + amb_y);
    ch[1]= (amb_w - amb_x + amb_y);
    ch[2]= (amb_w - amb_x - amb_y);
    ch[3]= (amb_w + amb_x - amb_y);
    return;
}
```

4. WDSPA: an implementation for Windows (Microsoft) OS

The main goal of the WDSPA (by Oscar Pablo Di Liscia) is to join the most useful techniques for sound spatialisation on a single stand-alone computer program, with an effective and easy-to-use Graphic Interface to be run on a widely used OS.

WDSPA is a computer program written in the C programming language for Windows OS (Microsoft). At present was tested successfully under Windows95, Windows98 and Windows ME. WDSPA is based on a program already developed (DSPA, 1997-1999, O. P. Di Liscia, See Di Liscia 1993, 1999) with versions both for Linux and Windows OS. However, WDSPA has its own GUI (Graphic Interface) while the formers are command-line versions.

WDSPA uses the following cues to simulate room, location of source and movement in a two dimension space:

- 1-Amplitude scaling: using either Intensity Panning or Ambisonic.
- 2-Frequency shift: according source and listener relative speed (Doppler shift)
- 3-Early echoes: according room geometry.
- 4-Global reverberation: with control of diffusion, t_{60} , gain and direct / reverberant ratio.
- 5-Filtering due to absorption of gasses in the air.

The data of general configuration (room size, wall absorption, T_{60} , etc.) as well as of movement path is stored on a binary file with proprietary format.

The program assumes that the listener is located at the origin of a Cartesian plane and surrounded by four loudspeakers(having angles of 45, 135, 225 and 315 degrees), at equal distance of the origin(unity by default).

Since the program was conceived for sound reproduction under loudspeakers, some powerful cues (H.R.T.F., see Kendall et al 1989) were disregarded.

WDSPA reads an input audio signal stored in an input file (RIFF WAV standard format) and writes the result of the action in one (stereo) or two (stereo) output files (RIFF WAV standard format also). The parameters of the output signal (Sampling Rate and Bytes per sample)will be taken from the input file.

5. Specific features of WDSPA

At present WDSPA only handle First Order Ambisonic, B Format, other formats will be further added. Since WDSPA allows both to write the encoded and decoded signal, it is possible to store an entire electronic composition on B format and to decode it for any rig available on any particular playback situation. It is possible, however, to spatialise any signal using intensity panning as well, and this is most useful for comparison of perceptual results.

The spatialisation is performed using two groups of data which can be generated using WDSPA and stored on disk in files with proprietary format (detailed data of this format is available under request):

Parameters:

Those are general data, as following:

- General gain.
- Direct gain.
- Early Echoes gain.
- Lowpass Cutoff Frequency for the Early Echoes.
- Reverb in gain.
- Reverb out gain.
- Reverb t60.
- Reverb Diffusion.
- Local Reverb gain.
- Room Size.
- Output type (Stereo, Quad, Encoded Ambisonic, Decoded Ambisonic)

Path:

In order to simulate movement and/or location of a virtual sound source, WDSPA needs a "spatial path". This path is stored in N segments, each one being limited by two points (or nodes). A simple example of path may be:

x=1 ;first node x rectangular coordinate

y=1 ;first node y rectangular coordinate

t=100 ;time in % it will take to go from first node to second

;100 will mean the entire sound file

a=0 ;zero will produce constant speed. Nonzero positive and negative numbers will produce an exponential speed curve (increasing to the middle of the segment, decreasing till the end of the segment)

x=-1 ;second node x rectangular coordinate

y=1 ;second node y rectangular coordinate

The path can be created and modified by:

1-Left-clicking the mouse on a location where there is NOT a node, will create a new node.

2-Holding the left button on a node will allow to drag it to any location.

3-Right-clicking on a node will open a menu which offers:

-undo

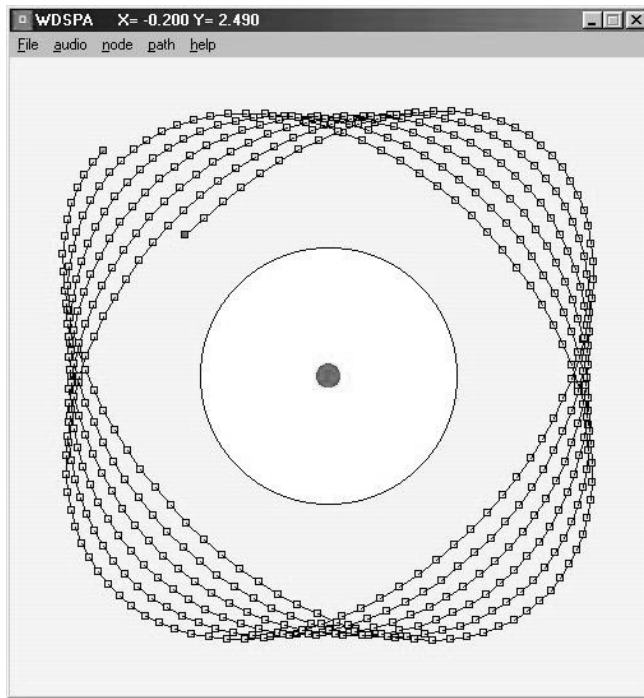
-delete the node

-interpolate new node/s: at present, only linear interpolation between the two nodes involved is performed. The number of nodes to interpolate can be set at */node/interpolation settings*.

Accessing to */node/edit segment data*, you can specify x,y,t and a arguments typing numbers.

The *path* menu allows transforming the entire path on some useful way. Some transformations are very easy to guess while other are not so. As an example, the *set equal time steps* option will set the same time for all the segments (the number of segments is computed and the "t" for each one is set to 100 / nsegments), while the *scale time steps* option will make the time steps proportional to the distance between points.

Knowing the format of the *spa* files, some programs using special routines to design spatial trajectories may be also written, since a constant experience has proved that to draw with mouse strokes is somewhat tedious and useless when complicated -but accurate- movements are required. The solution will be further included within WDSPA as special routines. See, for example, the plot of one output path of the program *phasor.exe* (by O. P. Di Liscia), which consists on a moving phasor with control of frequency, direction, X, and Y scaling and offset (translation):



(Plot of a spatial trajectory stored on a SPA file obtained with the program *phasor.exe* by O. P. Di Liscia. Each little square constitutes a “node”)

6. Conclusions

The technique of Ambisonic was tested using a four-loudspeaker square rig, on several concerts and play-back situations, and was judged to be more perceptually effective than intensity panning.

The Program is being used by several composers and has proved to be very robust, easy to handle and perceptually effective.

Future improvements of this research will address to include more encoding-decoding formats.

7. Acknowledgements

The author is most grateful to:

Juan Pampin, and Fernando Lopez Lezcano (C.C.R.M.A. at Stanford University) for so many useful suggestions and assessment.

C.A.R.T.A.H. (Center for Advanced Research Technologies in Arts and Humanities of Washington University, Seattle, USA) for support and hardware supply in research and testing (Ambisonic Microphones).

Richard Furse and David Malham for their valuable developments and documentation on Ambisonic technology.

To Jezar, Technology Consultant at *Dreampoint Design and Engineering* (UK), for providing the data for high quality reverberation using arrays of comb and allpass filters (Freeverb).

8. References

- Blauert, J. (1983). *Spatial Hearing*. MIT Press, Cambridge, Massachusetts.
- Bamford, J. S. (1995). *An Analysis of Ambisonic Sound Systems of First and Second Order*. Master Thesis. Waterloo University, Ontario, Canada.
- Bargen, B. & Donnelly, P. (1998). *DirectX a fondo*. MacGraw-Hill Interamericana de España, Madrid.
- Bossi, Marina (1990). *An interactive Real-time System for the control of sound localization*. Computer Music Journal, Vol. 14, No.4, MIT.
- Chowning, John (1992). *Music from machines: Perceptual Fusion. Auditory Perspective*. C.C.R.M.A., Stanford University.
- Chowning, John (1971). *The simulation of Moving Sound Sources*. Journal of the Audio Engineering Society, 19.
- Di Liscia, O. and Bonnier, E. (1996). *DSPA: software tools for spatial treatment of sound*. Anais do III Simpósio Brasileiro de Computacao e Música, Recife, Brazil.
- Di Liscia, O. P. (1999). *RTSPA1: a program and some remarks on real time spatialisation of sound*. Anais do VI Simpósio Brasileiro de Computacao e Música, Rio de Janeiro, Brazil.
- Dodge, Ch. & Jerse, T. (1985). *Computer Music Synthesis, composition and performance*. Schirmer Books, New York.
- Ellen Richard (1991). *Whatever happened to Ambisonics?* (<http://www.audiomedia.com>)
- Furse, Richard (1995). *Spatialisation Tools*. (<http://www.muse.demon.co.uk/csound.html>)
- Furse, Richard (1999). *Vspace*. (<http://www.muse.demon.co.uk/vspace/vspace.html>)
- Gardner, M. B. (1969). *Distance Estimation of 0° or Apparent 0°-oriented Speech Signals in Anechoic Space*. Journal of the Acoustical Society of America, 45.
- Haas, H. (1951). *Über den einfluss eines einfahechos auf die hörsamkeit von sprache*. Acustica, Vol.1.
- Hartmann, W. M. (1983). *Localization of sound in Rooms*. Journal of the Acoustical Society of America, 74.
- Karpen, Richard (1998): *Space and Locsig Ugs*. In *The Csound Manual*, (<http://hem.passagen.se/rasmuse/Csound.htm>), MIT.

- Kendall et al (1989): *Spatial reverberation. discussion and demonstration*. In: M. Matthews and J. Pierce: *Current Directions in Computer Music Research*, MIT Press, Cambridge, Mass.
- Lezcano, F. (1992). *Quad sound playback hardware and software*. C.C.R.M.A., Stanford University.
- Malham, David G. (1998). *Spatial hearing mechanisms and sound reproduction*. University of York, England. (<http://www.york.ac.uk/inst/mustech/mtg.htm>)
- Moore, F.R. (1990). *Elements of Computer Music*. Prentice Hall., New Jersey.
- Moore, F.R. (1983). *A General Model for Spatial Processing of Sounds*. *Computer Music Journal*, Vol. 7, No.3.
- Moore, F.R. (1989). *Spatialisation of sounds over loudspeakers*. In: M. Matthews and J. Pierce: *Current Directions in Computer Music Research*, MIT Press, Cambridge, Mass.
- Moorer, A. (1979): *About this reverberation business*. *Computer Music Journal*, Vol.3, N°2.
- Wallach, H. et al (1973). *The precedence effect in sound localization*. *Journal of the Audio Engineering Society*, 21.