AMBISONICS EQUIVALENT PANNING

Martin Neukom, Jan C. Schacher
Zurich University of the Arts
Institute for Computer Music and Sound Technology
Baslerstrasse 30
8048 Zürich, Switzerland
{martin.neukom, jan.schacher}@zhdk.ch

ABSTRACT

In Ambisonics sound is encoded and stored in multi-channel sound files and is decoded for playback. In the en- and decoding process complicated functions are used. In this paper panning functions equivalent to the result of en- and decoding are presented which can be used for real-time panning in an arbitrary high ambisonic order. In the function equivalent to the so-called in-phase decoding the order of ambisonic resolution is just a variable that can be any positive number not restricted to integers and that can be changed during playback. The limitations and advantages of the technique are mentioned, real time applications are described and the computational costs are estimated. The described procedure offers a new intuitive understanding of Ambisonics.

1. INTRODUCTION

For a project called Interactive Swarm Orchestra, where hundreds of virtual moving sound sources have to be rendered in real-time, we have been looking for a simplification of the complicated and computational costly calculations of Ambisonics. The theoretical results were published in an AES paper in October 2007 [1]. The main results of this paper are recapitulated in a less technical stile in paragraph 4. Paragraphs 2 and 3 give a short introduction to the concepts of panning and Ambisonics. In paragraph 5 more practical issues such as implementations, limitations and advantages of the technique are addressed.

2. PANNING

Panning is the technique of the positioning of a single (monophonic) source within a stereophonic image. Vector Base Amplitude Panning (VBAP) was introduced by Ville Pulkki [2] for two dimensions. In VBAP loudspeaker arrays are treated as arrangements of subsequent stereo pairs or, when extended to three dimensions, as triples of loudspeakers. Panning normally uses only level differences and feeds only the loudspeakers nearest to the virtual sound source.
In amplitude panning the sound signal x(t) is fed to speaker i with the gain factor f_i

$$x_i(t) = f_i x(t), \quad i = 1, \ldots, N \tag{1}$$

In contrast to other panning techniques ambisonic panning functions normally produce signals for all speakers at the same time. The functions are defined on the whole horizontal circle or the whole sphere. The sum of all speaker gains equals 1.

3. AMBISONICS

Ambisonics is a surround-system for encoding and rendering a 3D sound field. In Ambisonics the room information of the recorded or synthesized sound is encoded together with the sound itself in a specific number of channels independent of the final speaker set-up. The encoding can be carried out in an arbitrary degree of accuracy. The accuracy is given by the so-called order of Ambisonics. The zeroth order corresponds to the mono signal and needs one channel. In first order Ambisonics the components of the sound field in the directions x, y and z are encoded in three more channels. The interpretation of higher orders is much more complicated than that of zeroth and first order.

3.1. Encoding

The formulas for ambisonic encoding are derived from the solution of the three-dimensional wave equation in the spherical coordinate system where a point P is described by radius r, azimuth θ and elevation δ. Assuming that the sound waves are plane and that the listener is located at the origin of the coordinate system the formulas can be simplified dramatically. In practice the arising infinite series is truncated and only a finite number of components are calculated and saved in the so-called ambisonic B-format. After all these simplifications a signal S is encoded by multiplying the signal with the first spherical harmonics in 3D and with the first harmonics in 2D [3][4].
The order of resolution m defines the accuracy of the encoding and the number of channels in the B-format, namely 2m+1 in 2D and $(m+1)^2$ in 3D.

3.2. Decoding

From a B-format file with n channels and a given set-up of at least n speakers the signals for the speakers can be calculated. They are a weighted sum of the B-format channels. The vector **S** of the speaker signals for symmetrical setups of n speakers can be calculated from the B-format and the matrix of the B-format of the speaker signals **C** as [3]

$$S = C^{-1}.B = \frac{1}{n} C^{T}.B \tag{2}$$

Since solutions for asymmetrical setups often are unusable (for 5.1 surround see [5]) normally this symmetric solution is used, even if the speaker set-up is not exactly symmetric.

3.3. Corrections

The truncation of the infinite series causes side effects such as signals on speakers far away from the original sound position and inverted phases (see figure 1). By windowing the decomposition i.e. weighting the ambisonic channels according to their order these side effects can be reduced at the cost of the precision of the directivity. Figure 2 shows two level functions for a speaker at position θ (sound at θ = 0, order m = 3) the first without correction (basic decoding) $f_{bas}(\theta)$, the second for so-called in-phase decoding $f_{inph}(\theta)$. The bars indicate the levels of 13 symmetrically positioned speakers. Thus for a sound source at position θ_s we get the gain for the speaker i at position θ_i as $f(\theta_s-\theta_i)$.

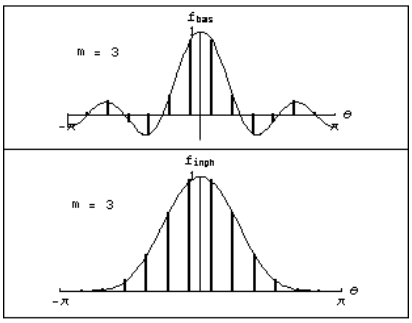

Figure 1. Level functions for basic and in-phase decoding.

Putting the correcting gains into equation (2) yields

$$S = \frac{1}{n} C^T Diag[...g_m...].B \tag{3}$$

For symmetric set-ups this equation can be simplified and we get the ambisonic panning functions [6]

$$f(\theta,m) = \frac{1}{n}(g_0 + 2\sum_{k=1}^{m} g_k \cos k\theta) \quad in\,2D \tag{4}$$

$$f(\theta,m) = \frac{1}{n}\sum_{k=1}^{m}(2m+1) g_k P_k(\cos\theta) \quad in\,3D \tag{5}$$

The correcting gains for in-phase decoding are

$$g_k = g_0 \frac{m!^2}{(m+k)!(m-k)!} \quad in\,2D \tag{6}$$

$$g_k = g_0 \frac{m!(m-1)!}{(m+k+1)!(m-k)!} \quad in\,3D \tag{7}$$

with normalizing factors g_0 [3].

4. AMBISONICS EQUIVALENT PANNING (AEP)

For basic and in-phase decoding the formulas (4) and (5) can be simplified.

4.1. Basic Decoding

For basic decoding (i.e. without correcting gains) the panning function (4) can be written in the simplified form

$$f(\theta,m) = \frac{\sin(\frac{2m+1}{2}\theta)}{n\sin(\frac{1}{2}\theta)} \tag{8}$$

This function is exactly equivalent to ambisonic en- and decoding in 2D. For higher orders the number of calculations is reduced dramatically with this technique. Figure 2 shows the panning function for order 15.

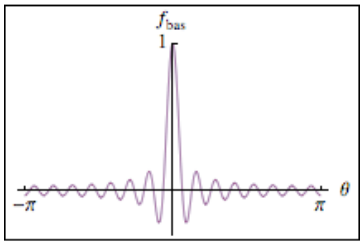

Figure 2. Ambisonic panning function for order 15.

The function depends only on the angle between speaker and sound source and so it can also be used in 3D as an approximation of the ambisonic panning function [1].

4.2. In-phase Decoding

In [1] we showed that the panning functions (4) and (5) with the gains (6) and (7) are equivalent to the simple function

$$f_{inph}(\theta,p) = (\tfrac{1}{2} + \tfrac{1}{2}\cos\theta)^p = (\cos\frac{\theta}{2})^{2p} \tag{9}$$

where θ is the angle between the speaker and the position of the sound source and p corresponds to the ambisonic order.
While ambisonic encoding is only possible with integer orders the exponent in the panning function $f_{inph}(\theta,p)$ can be an arbitrary positive number. Figure 3 shows the function for orders 1, 3 and 6.7.

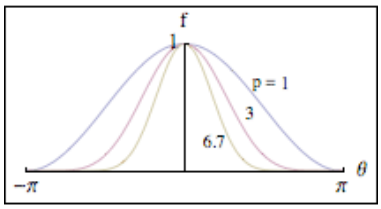

Figure 3. AEP-function for order 1, 3 and 6.7

For fractional orders the sum of the speaker signals does not exactly equal one but the deviation is very small, so that it is possible to change the exponent continuously without perceivable inaccuracies. Figure 4 shows the sum A of the speaker signals for 8 speakers as a function of order p and the angle θ of the sound source. It is nearly constant between p = 2 and p = n-1.

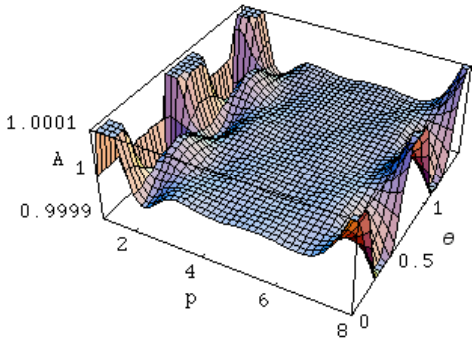

Figure 4. Sum A of 8 speakers signals as function of the order p and the angle θ of the sound source.

Since with increasing order p the function $f_{inph}(\theta,p)$ narrows more and more slowly, fewer speakers per order are necessary. Figure 5 shows that with as few as 20 speakers it is possible to use orders up to approx. p = 60.

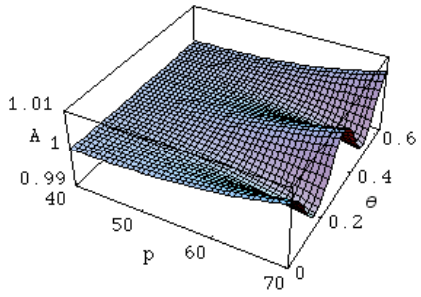

Figure 5. Sum A of 20 speakers as function of the order p and the angle θ of the sound source.

The same panning function can be used in 3D. ~~The~~ Only five symmetrical speaker setups correspond to the five platonic solids; for these setups it can be shown that the sum of the speaker signals is independent of the position of the sound source for small integer orders [1]. The sum can be normalized by the factor (p+1)/n

$$\frac{p+1}{n}\sum_{i=1}^{n} f_{inph}(\theta_i,p)=1 \qquad (10)$$

where θ_i is the angle between the sound source and the speaker i, p the order and n the number of speakers.

5. IMPLEMENTATION AND APPLICATIONS

5.1. Implementation

The implementation of the panning functions f_{inph} (9) is straightforward. In order to produce the signal for a certain speaker at position $P_s = (x_s, y_s, z_s)$ a sound at position P = (x, y, z) is multiplied by f(θ,p) where θ denotes the angle between the sound source and the speaker. The cosine of this angle is calculated as the scalar product $(x, y, z).(x_s, y_s, z_s)$. For a speaker setup on a sphere or a circle with radius 1 and a sound source at distance r we get in Cartesian coordinates

$$f_{inph}(P,P_s,p)=\left(\frac{xx_s+yy_s+r}{2r}\right)^p \quad in\,2D \qquad (11)$$

$$f_{inph}(P,P_s,p)=\left(\frac{xx_s+yy_s+zz_s+r}{2r}\right)^p \quad in\,3D$$

$$where\ r=\sqrt{x^2+y^2+z^2}$$

and in spherical coordinates

$$f_{inph}(P,P_s,p)=\left(\frac{1+\cos(\theta-\theta_s)}{2}\right)^p \quad in\,2D \qquad (12)$$

$$f=\left(\frac{1+\cos(\theta-\theta_s)\cos(\delta)\cos(\delta_s)+\sin(\delta)\sin(\delta_s)}{2}\right)^p \ in\,3D$$

5.2. Computational Costs without Look-up Tables

In [1] the complexity of ambisonic en- and decoding and AEP have been estimated and compared for implementations that do not use look-up tables. There are about 0, 3, 16, 45, 96, 177, 300, …(approx. $(m+.5)^3$) multiplications for the orders 0, 1, 2, … for the encoding of each signal. In the decoding process the matrix $\mathbf{C}^T$ is multiplied with the B-format. This takes an additional $n*(m+1)^2$ multiplications.
The panning function (11) takes only 3 or 4 multiplications and 1 function call for every sound and speaker.

5.3. Look-up tables

For applications in computer music where often a great number of independent sound sources are treated and a great number of speakers are used it is reasonable to use look-up tables instead of repeated calculations. In order to estimate the complexity of the implementation and the computational costs we have to take into account the dimensionality of the tables, the number of elements in the tables and the type of interpolation.

5.3.1. Two dimensional tables

Using tables with more then one dimension poses some problems. 1) The most common sound synthesis languages as Csound and Max (with the exception of extensions such as Jitter, FTM or language bindings like

Java, Python etc) do not support them. 2) Tables with length n per input variable need n^d entries for dimensionality d. For a resolution of for example 1024 points for two variables 1 MB of RAM is used. Thus RAM limits the table size and using interpolation becomes imperative.

5.3.2. Ambisonic en- and decoding

In 2D-Ambisonics the harmonics are just sine and cosine functions. They can be used as table look-up functions in the same way as in a standard oscillator.

In 3D ambisonic en- and decoding spherical harmonics are used. Since they are functions of two variables θ and δ we need 2 dimensional arrays. Tables are used whose dimensions correspond to the number of B-format channels $(m+1)^2$. Since the higher order spherical harmonics are complicated (see figure 6) a good resolution or high order interpolation is necessary.

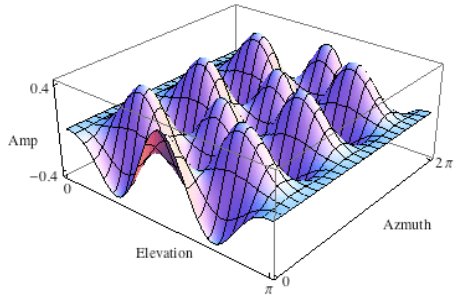

Figure 6. A spherical harmonic fifth order: Y^3_5

5.3.3. AEP-function of difference of angles

The input variable in the panning function (9) is the angle θ_s-θ_i between sound source and speaker i. If this angle is calculated and the order is constant a one-dimensional table is used. A two dimensional table is needed if the order is variable. Since the order normally changes slowly, and the changes of the function are accordingly small, the resolution of this parameter does not need to be very good and no interpolation is needed. Since the function is symmetric the size can be halved if we need abs(θ-θ_i) as input. Because the difference of the angles between sound source and speaker ~~are~~ is used, only one table is used and it is independent of the speaker set-up.

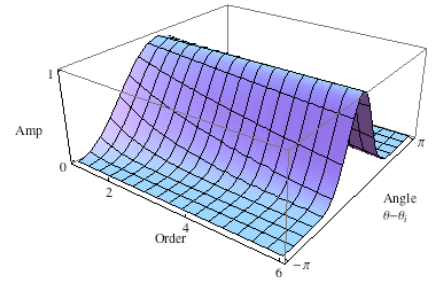

Figure 7. Panning as function of the order and the angle between sound source and speaker i.

5.3.4. AEP-function of angles of the sound source

In order to avoid the calculation of the angle between sound source and speaker we can produce a table for the amplitude of every speaker as function of azimuth θ and elevation δ of a sound source. Figure 8 shows a speaker set-up in a tetrahedron (left) and the panning function for the speakers at point (1,0,0) (centre) and at point (0,1,0) (right).

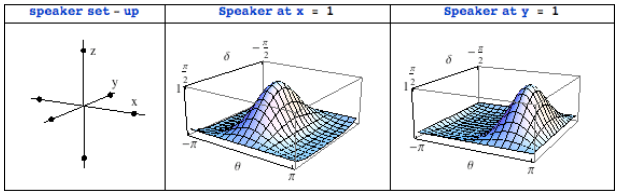

Figure 8. Functions for different speakers

Whilst the difference of the angle between a sound source and a speaker can not be calculated from the difference to an other speaker, the functions of the two variables θ and δ are similar but shifted in direction of the axes according to the angles of the speakers. Thus instead of calculating and storing as many functions as speakers only the function for one speaker can be stored and the values for the other speakers can be read out by adding the difference of the angles between the speakers to the input values of the function.

5.4. Csound

The following code sample shows a very simple implementation of AEP in Csound for a set-up with only three speakers. During the course of an event the sound source turns twice around the listener (kfi goes from 0 to 4π) and the order grows from 1 to 10.

```
nchnls = 3

instr 1
i2pi    =       2*3.141592
ifi1    =       0
ifi2    =       i2pi/3
ifi3    =       2*i2pi/3
kp      line    1,p3,10
kfi     line    0,p3,2*i2pi
kpan1   =       (.5+.5*cos(kfi-ifi1))^kp
kpan2   =       (.5+.5*cos(kfi-ifi2))^kp
kpan3   =       (.5+.5*cos(kfi-ifi3))^kp
a1      rand    30000
        outc    a1*kpan1,a1*kpan2,a1*kpan3
endin
```

Figure 9 shows the resulting amplitudes of the three channels over the duration of the event. It is obvious that in the second half of the time the order becomes too high for the small number of speakers resulting in the absence of a signal for part of the sweep.

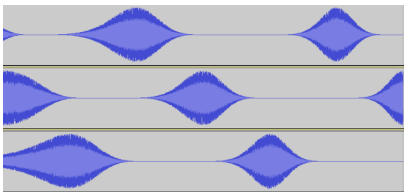

Figure 9. Resulting sound file of the Csound example.

With the following code the panning function of order 3.5 is saved in a sound file.

```
i2pi    =       6.2831853
kfi     line    0,p3,i2pi
a1      =       (.5+.5*cos(kfi))^3.5
        out     32767*a1
```

The sound file then is used as a table for the table look-up function that can be put in the above instrument.

```
kpan1   table   ipi2*(kfi-ifi1), 1, 1, 0, 1
```

More examples are available from [7].

5.5. MaxMSP

The implementation of AEP as a patcher in MaxMSP is fairly straightforward. The expression for the calculation of the signal amplitude for one speaker takes a source position, a speaker position and the order factor as shown in Fig. 10.

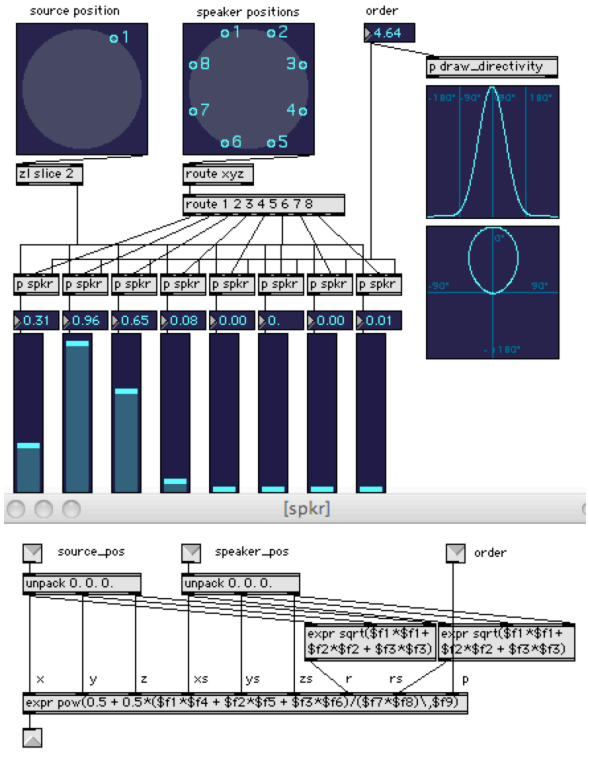

Figure 10. Patcher based MaxMSP implementation of the AEP ideally suited for didactic purposes.

It is necessary to use more optimized forms of the process for a large number of sources; this is made available as a MaxMSP external programmed in C [7]. The external is based on interpolated two-dimensional look-up tables and is optimized for static speakers and a multitude of sources. It implements an n by m signal matrix and includes position input in either Cartesian or polar coordinates, distance correction on both the source and speaker amplitudes and delay-time correction for the speaker feeds. The control syntax is the same as for the other ICST ambisonics tools in order to facilitate interchanging the processes [8].

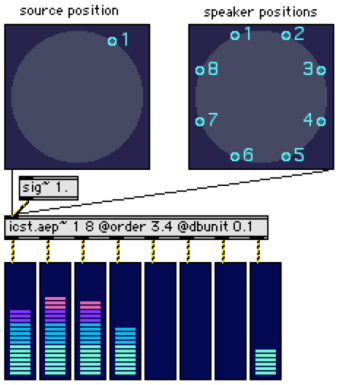

Figure 11. Implementation of the AEP as a MaxMSP external structured like an n by m matrix mixer.

5.6. Computational Costs with Look-up Tables

The effective computational costs of the different procedures depend on hardware and on the programming environment. In the following examples we therefore only list the number of multiplications, function calls and table look-ups for an order m and a number of speakers n. In general with interpolation the number of table look-ups is doubled in 2D and quadrupled in 3D and 2 additional multiplications occur in 2D or 8 additional multiplications occur in 3D. For the procedure described in paragraph 5.3.2 we only need the $(m+1)^2$ table look-ups per sound for the encoding; ~~(~~decoding is processed for all sounds together and thus computing costs does not increase with a higher number of sources~~)~~. For the procedure described in paragraph 5.3.3 we need 4 multiplications to calculate the angle between sound source and speaker and n table look-ups per sound. In the context of the I-S-O project (Interactive Swarm Orchestra [9])~~,~~ which was the starting point for our study, ambisonic en- and decoding ironically outplays AEP if no look-up tables are used and is of the same complexity and cost if one takes full usage of look-up tables. Since in this project many sound-sources are used the encoding process causes the biggest costs and so we only need $(m+1)^2$ table look-ups for ambisonic encoding and n table look-ups for AEP. For example with a speaker set-up in a dodecahedron (n = 20) up to m = 3 traditional ambisonic encoding still beats AEP.

5.7. Explaining Ambisonics

The mathematics used in ambisonic theory is beyond the skills of non-scientists or non-engineers. Since panning functions are familiar and easy to visualize they provide

a good didactical means for explaining Ambisonics to laymen and for deriving encoding formulas and gains for in-phase decoding. For example for order m = 3 we expand the powers of the panning function (…)

$$f_{inph}(\theta, p) = (\tfrac{1}{2} + \tfrac{1}{2}\cos\theta)^3 \\ = \tfrac{1}{8}(1 + 3\cos\theta + 3\cos^2\theta + \cos^3\theta) \quad (13)$$

and replace the powers of the cosine function by cosines of multiples of the angle to get the cosine part of the B-format together with the in-phase coefficients (6).

$$\tfrac{1}{32}(10 + 15\cos\theta + 6\cos 2\theta + \cos 3\theta) \quad (14)$$

6. CONCLUSIONS AND FURTHER INVESTIGATIONS

In order to store 3D sound independently of the speaker set-up either sound and position of the sound source can be stored separately or they can be stored together in the ambisonic B-format. If the sound need not be stored, ~~e.g~~. for example in testing environments, real time performances or playback of multi-channel sequencer-sessions AEP is easier to implement and takes less computing power than ambisonic en- and decoding. Only in applications where a great number of independent sounds occur, each with its own position or movement ~~occurs~~ and the spatial resolution need not to be very high ambisonic en- and decoding outplays AEP. In most other cases AEP performs better.

A further advantage of AEP is the possibility to use an arbitrary order of directivity for each individual sound source. It becomes possible to mix pre-recorded low order ambisonic B-format, medium order ambient sounds, high order precise localisable sound and sounds with changing localizability. How the individual sounds are perceived if different orders are used at the same time is an open question that can be answered only by experience. Can we simulate different sizes or directivities of the sound source or will we just hear unnatural~~,~~ incongruent sound scenes?

An open theoretical question is, whether there are simple formulas for ambisonic panning with other correcting gains of ambisonic decoding as for example basic decoding in 3D and so-called max r_E decoding [3].

A lot of work has still to be done to implement AED and especially 2D table look-up for different sound synthesis languages and to create plug-ins for commercial software.

7. REFERENCES

[1] Neukom, M. "Ambisonic Panning", AES 121st Convention, New York, USA, 2007.

[2] Pulkki, V. "Virtual sound source positioning using Vector Base Amplitude Panning", J. Audio Eng. Soc., 45, June 1997.

[3] Daniel, D. *Représentation de champs acoustiques, application à la transmission et à la reproduction de scènes sonores complexes dans un contexte multimédia*. Ph.D. Thesis, University of Paris VI, France, 2000, http://gyronymo.free.fr

[4] Sontacchi, A., Höldrich, R. "Konzepte zur Schallfeldsynthese und Schallfeldreproduktion", Jahrestagung der ÖPG FA-Akustik, 2000, http://iem.at/projekte/publications/paper

[5] Neukom, M. "Decoding Second Order Ambisonics to 5.1 Surround Systems", AES 121st Convention, San Francisco, CA, USA, 2006

[6] Daniel, D., Nicol, J. R., Moreau, S. "Further Investigations of Higher Order Ambisonics and Wavefield Synthesis for Holophonic Sound Imaging", AES 114th Convention, Amsterdam, The Netherlands, 2003

[7] http//:www.icst.net/downloads

[8] Schacher, J.C., Kocher, P. "Ambisonics Spatialization Tools for Max/MSP", Proceedings of the International Conference on Computer Music 2006 (ICMC'06) New Orleans, November 6-11, 2006

[9] http://www.i-s-o.ch