Improving Csound's Ambisonics decoders

Pablo Zinemanas[1], Martín Rocamora[1] and Luis Jure[2] ⋆

[1] Facultad de Ingeniería
[2] Escuela Universitaria de Música
Universidad de la República
lj@eumus.edu.uy

Abstract. This paper describes the efforts we devoted to improve Ambisonics decoders in Csound. Current version of the existing opcode, namely `bformdec1`, has some limitations that should be surpassed in order that the decoders better fulfill the Ambisonics criteria. In particular, the implemented decoders have no near–field compensation and do not use a different decoding matrix for low and high frequencies. These issues are addresses in a new implementation of the opcode, namely `bformdec2`, that also adds some features, such as additional loudspeaker array configurations (rectangle, hexagon) and a binaural output for headphones.

Keywords: Ambisonics decoder, HOA, Csound

1 Introduction

Ambisonics is a spatial sound metatheory (a theory of theories) for audio recording, coding and reproduction, developed by Michael Gerzon in the 1970s [4]. It provides a method of codifying physical properties of a sound field (the pressure and velocity components), that captures directional information of the sound sources, and enables its accurate reconstruction in a point in space.

Unlike traditional multichannel audio for spatialization in which each channel corresponds to a given loudspeaker, the Ambisonics sound format—called B-format—contains a speaker-independent representation of a sound field. By means of an appropriate decoder, i.e. matched to the geometry of the loudspeaker array, this sound file can be played back in different speaker layouts [2].

The channels of the B-format file can be regarded, from a theoretical perspective, as the coefficients of a series expansion of the sound field around the origin, in terms of spherical harmonics [2,9]. The spherical harmonics are a complete set of orthogonal functions on the sphere, and thus can be used to represent functions defined on the surface of a sphere.

The order of the series expansion determines the number of channels involved. Thus, zeroth order Ambisonics represents the omni–directional component of the sound field and corresponds to the the sound pressure, which consist of one single

⋆ This research was partially funded by Comisión Sectorial de Investigación Científica, Universidad de la República, Uruguay. We thank Aaron Heller for his advise.

channel (the W channel). First order Ambisonics adds three directional components (channels X, Y and Z), corresponding to the pressure gradient and representing the acoustic velocity. Higher order Ambisonics (HOA) add additional coefficients to the series expansion, corresponding to higher order derivatives of the sound field [9]. Increasing the order of the series expansion provides better approximation of the sound field, which leads to increased spatial resolution.

2 Ambisonics decoding

The decoder has to provide suitable linear combinations of the B-format signals for each loudspeaker in the array, so that the pressure and particle velocity is reproduced correctly at the listening position, i.e. the centre of the array. The set of coefficients needed to produce that linear combination is called the decoding matrix. The number of loudspeakers must be at least the number of the B-format signals [8]. There are essentially two different approaches that can be adopted: the basic (or physical)[3] decoding and the energy (or psychoacoustic) decoding [9]. It turns out that the basic decoding achieves accurate perception of spatial localization only at low frequencies, where as the energy decoding provides optimal localization only for high frequencies. For this reason, a better approach consist in using different solutions for low and high frequencies.

2.1 Physical decoding

The basic decoding seeks the reconstruction of the sound field, up to a given Ambisonics order, from the superposition of the sound waves emitted by the loudspeakers, assuming phase coherence among the signals [9]. In essence, the solution of the decoding equations corresponds to the projection of the spherical harmonics to each of the directions of the speakers. In most cases the number of speakers is greater than the number of Ambsionics channels, which yields an under-determined system of equations whose solution can be obtained with algebraic methods (pseudo-inverse). For regular speaker arrays the problem is well–conditioned and the method will result in a correct solution. For irregular arrays the solution could be still obtained with algebraic methods but the problem is often ill–conditioned, making the obtained solution inappropriate.[4] The basic decoding succeeds at reproducing the impression of sound source locations only at low frequencies (approximately below 500 Hz), and close to the center of the loudspeaker array [9,2]. For higher frequencies or a large listening area it is better to use a psychoacoustic decoder.

2.2 Psychoacoustic decoding

The psychoacoustic decoding aims at reproducing the original energy and acoustic intensity of the sound field, assuming an incoherent sum of the speakers

[3] Other names are used to refer to this decoding solution, such as exact or velocity.

[4] Although there are several proposals to deal with irregular arrays, this still remains as an area of open research [9,5].

signals [9,2]. By incoherently summing the signals of several loudspeakers it is physically impossible to exactly reconstruct the acoustic intensity, so the decoder will instead try to maximize a statistical estimator of the signal energy. In the case of regular (or semi-regular) speaker arrays it is possible to obtain the energy decoder by altering the coefficients of the basic decoder matrix.

The *in-phase* decoders additionally impose the restriction that no loudspeaker emits in opposite phase [9]. This provides more robust localization for listeners who are far from the center of the loudspeakers array.

2.3 Dual–band decoding

Given that no decoding approach is adequate for both high and low frequencies, many Ambisonics decoders split the B-format signals into (at least) two bands and use independent solutions for low and high frequencies [2]. Then the output of each band is recombined to produce the audio signals for the loudspeakers. It is important to note that the band–splitting filters must be carefully designed to preserve the magnitude and phase response of the signal [8]. Besides, when combining the output of each band, different criteria can be used to deal with the signal's level difference between the low and high frequencies, such as preserving the amplitude, the root–mean–square RMS level or the total energy [8].

2.4 Near–field compensation

Another important aspect of an Ambisonics decoder is to provide near–field compensation [3]. The recreation of the sound field at the central position holds under the hypothesis that the wavefronts are planar. Given the finite distance to the loudspeakers, the sound wavefronts at the listening position present instead a curvature, which produces a bass–boosting effect that has to be compensated. The compensation is essentially a high–pass filter, which depends on the order of reproduction and on the distance of the loudspeakers to the center of the array.

2.5 Criteria for correct Ambisonics decoding

In summary, as suggested in [8], apart from having a decoding matrix matched to the geometry of the loudspeaker array, we focus on the following key aspects for correct Ambisonics decoding:

- dual–band decoding (high and low frequencies) using phase–matched filters
- near–field compensation, implemented as a high–pass filter.

These features are not provided in the Ambisonics decoders currently available in Csound, so they are addressed in a new opcode implementation.

3 Current Ambisonics decoders implementation

Ambisonics decoders in Csound are implemented in the `bformdec1` opcode.[5] There are five loudspeaker layouts available: stereo, quad (2D square), 5.0, octagon and cube. Given the constrain on the number of loudspeakers for a given Ambisonics order,[6] all decoders are first–order, except for the octagon layout, which provides first–, second– and third–order decoders, and the 5.0 which has first and second order. It is important to note that the decoders are of the in–phase type—the same decoding matrix used in low and high frequencies—and without near field compensation.

4 New Ambisonics decoders implementation

The new implementation of the Ambisonics decoders, namely `bformdec2`, focus on providing dual–band decoding and near–field compensation. It is designed with backward compatibility in mind, so it offers the same loudspeaker layouts available in `bformdec1`. Besides, some additional loudspeaker array configurations are provided, including a binaural output for headphones.

4.1 Dual–band decoding

The decoders implemented are dual–band, providing a different decoding matrix for low and high frequencies. The band splitting filters are designed to be phase–matched, as described in [8] and explained below.

Phase–matched dual–band splitting filters The dual–band splitting is obtained by combining two second order–filters, a low–pass and a high–pass filter, that are phased matched. The phase match is achieved by reversing the phase response of the high-pass filter in order to match that of the low-pass filter. The two filters acting together give a first–order all–pass filter [8,6].

The filters are implemented as infinite–impulse response (IIR) filters, as,

$$H(z) = \frac{b_0 + b_1 z^{-1} + b_2 z^{-2}}{a_0 + a_1 z^{-1} + a_2 z^{-2}}$$

Coefficients a_i are calculated by:

$$a_1 = \frac{2(k^2 - 1)}{k^2 + 2k + 1}, \quad a_2 = \frac{k^2 - 2k + 1}{k^2 + 2k + 1}$$

for both filters, whereas b_i coefficients are:

$$b_0 = \frac{k^2}{k^2 + 2k + 1}, \quad b_1 = 2b_0, \quad b_2 = b_0,$$

[5] There is a deprecated opcode, namely `bformdec`, which was tested in [8].

[6] For an Ambsiconic order l, $(l+1)^2$ loudspeakers are needed for full–sphere systems, and $2l+1$ for horizontal–only reproduction.

for the low-pass filter and

$$b_0 = \frac{1}{k^2 + 2k + 1}, \quad b_1 = -2b_0, \quad b_2 = b_0,$$

for the high-pass filter; and $k = \tan\left(\pi \frac{f_c}{fs}\right)$, where f_c is the splitting frequency and fs is the sampling rate [8].

The filters of the `bformdec2` opcode are implemented by the Direct-Form II, using a similar code to that of the `filter2` Csound's opcode algorithm. The splitting frequency can be selected by the user, and its default value is 400 Hz.

Low– and high–frequency balance Starting from the basic decoding matrix, which is used for the low frequencies, the decoding matrix for the high frequencies is obtained by applying a set of coefficients, as explained in [7]. Different criteria can be used to balance the gain between low and high frequencies. By default, `bformdec` uses conservation of total energy, but via an optional parameter, any of two additional methods can be selected: preservation of the amplitude, and preservation of the root–mean–square (RMS) level. The coefficients are computed using the Ambisonics Decoder Toolbox[7] (ADT) [7,5].

4.2 Near–field compensation

The near–field compensation is achieved through the high–pass filters proposed by [3], following the implementation described in [1] and the code provided by ADT [5,6]. The order of the compensation filters corresponds to the Ambisonics order. The current implementation of `bformdec2` allows for near–field compensation of decoders up to order five. The user can set the distance to the speakers as an input parameter, and can also disable the near–field compensation.

4.3 Loudspeaker layouts and binaural output

The decoders implemented in `bformdec1` are hard–coded, that is, each decoder's data is fully embedded into the source code. In contrast, the implementation of `bformdec2` was designed to be modular, so that, from a decoding matrix for the basic solution and some input parameters, a set of functions compute the decoding solution. This offers some flexibility for the addition of loudspeaker layouts, even from a decoding matrix supplied by the user.

Current implementation of `bformdec2` provides the same layouts available in `bformdec1`, for backward compatibility. The additional loudspeaker arrays implemented so far are horizontal–only, namely hexagon and rectangle (for different length–to–width ratios). A binaural output for headphones is also provided.

The binaural output, up to third–order Ambisonics, is obtained through a two–step process. First, the input signal is decoded to a virtual loudspeakers

[7] `https://bitbucket.org/ambidecodertoolbox/adt/`

array (octagon for horizontal–only, and dodecahedron for full–sphere). Then, we compute the convolution of the signal of the loudspeakers and head–related transfer functions (HRTFs) corresponding to the direction of the virtual loudspeakers. The sum of the obtained signals yields the binaural output for headphones. The implementation of the HRTF convolution is based on the code of the `hrtfstat` opcode, and the set of HRTFs used is already available in Csound.

5 Discussion and conclusions

The implementation of a new opcode for Ambisonics decoding is released[8] that aims to remedy the lack of dual–band decoding and near–field compensation of the previous implementation. The opcode offers some backward compatibility and the option to try out the best decoder for one's particular needs, through parameters set by the user (e.g. band splitting frequency, disable near–field compensation). Ultimately, the most appropriate decoder may depend on the sound source material and the intended use, for instance, the size of the loudspeaker array and the listening area. In the future, the opcode will include more loudspeaker layouts and the option to use a decoding matrix specified by the user.

References

1. Fons Adriaensen. Near Field filters for Higher Order Ambisonics. online, accessed 29-Oct-2018, `https://kokkinizita.linuxaudio.org/papers/hoafilt.pdf`.
2. Jérôme Daniel. *Représentation de champs acoustiques, application à la transmission et à la reproduction de scènes sonores complexes dans un contexte multimédia.* PhD thesis, Université Paris 6, 2001.
3. Jérôme Daniel. Spatial sound encoding including near field effect: Introducing distance coding filters and a viable, new Ambisonic format. In *Proceedings of the Audio Engineering Society (AES) 23rd International Conference: Signal Processing in Audio Recording and Reproduction*, May. 2003.
4. Michael A. Gerzon. General Metatheory of Auditory Localisation. In *Proceedings of the Audio Engineering Society (AES) 92th Convention*, Mar. 1992.
5. Aaron Heller and Eric Benjamin. The Ambisonic Decoder Toolbox: Extensions for partial-coverage loudspeaker arrays. In *Linux Audio Conference (LAC)*, May. 2014.
6. Aaron Heller and Eric Benjamin. Design and implementation of filters for Ambisonic decoders. In *Proceedings of the 1st International Faust Conference (IFC)*, Jul. 2018.
7. Aaron Heller, Eric Benjamin, and Richard Lee. A toolkit for the design of Ambisonic decoders. In *Linux Audio Conference (LAC)*, Apr. 2012.
8. Aaron Heller, Richard Lee, and Eric Benjamin. Is my decoder Ambisonic? In *Proceedings of the Audio Engineering Society (AES) 125th Convention*, Oct. 2008.
9. Davide Scaini and Daniel Arteaga. Decoding of higher order ambisonics to irregular periphonic loudspeaker arrays. In *Proceedings of the Audio Engineering Society (AES)55th International Conference: Spatial Audio*, Aug. 2014.

[8] Available at `https://github.com/pzinemanas/bformdec2`.