PROPOSING SPATDIF - THE SPATIAL SOUND DESCRIPTION INTERCHANGE FORMAT

Nils Peters
McGill University, Montréal, CA
Centre for Interdisciplinary Research in Music Media and Technology (CIRMMT)
nils.peters@mcgill.ca

ABSTRACT

This paper outlines the requirements for an interchange format that can describe and share spatial parameters across 3D audio applications, and proposes SpatDIF for its implementation.

1. WHY USING A SCENE DESCRIPTION FORMAT FOR COMPOSING?

Formats as a structuring concept are integral to musical practice. For example, in the form of scores, a written symbolic representation of music, compositions can be stored, exchanged, studied, performed but also revised and adapted after their initial creation. MusicXML shows how the score concept is digitally maintained.
Although spatialization can be considered as a core element of electroacoustic music, there is no general consensus in how to describe and notate spatialization. Nowadays the spatial aspects are mostly created and automatized on a low-level within diverse digital audio environments, such as Max/MSP or ProTools. Because these environments have different syntaxes, units, and storage solutions, the control messages (e.g. a trajectory to move a sound in space) are only valid within this specific audio environment. Therefore the interchangeability of these descriptors is ineffectual and usually spatial aspects are directly rendered into multichannel sound files. Now that processing power is usually sufficient to render multiple virtual sound sources in real-time, a separation of "raw" sound material from the spatial descriptors within an open data format would increase the portability across different 3D audio applications, loudspeaker configurations and concert venues. Furthermore, spatial rendering algorithms could be compared and combined without having to change the spatial-sound syntax. This, of course, relies on the standardization of descriptors.

2. REQUIREMENTS

We need to develop an interchange format for spatial audio scenes—a "lingua franca" of spatial audio. The success of this effort will strongly depend on its acceptance within the communities. The challenge is to design a format that supports consumer needs and is also attractive for manufacturers and commercial developers of 3D-Audio tools, such as the following:

Easy connectivity with editors, interfaces and controllers to create spatialization in multiple ways.
Multiple layer of interaction to control and explore spatial features from different higher level viewpoints.
Human-readable syntax to prevent misunderstandings when exchanging stored data.
Real-time control of spatialization is mostly desired to explore the possibilities and interactions within the virtual space through receiving immediate audible feedback. But **also non-real-time** applications should be be supportable, such as OpenMusic [1].
Free and open source to increase the acceptance and widespread usage of the new format.
Extensibility is important as long as the format is under development, but moreover, to adapt to new developments in audio technology and compositional styles.
Platform independence permits audio scenes to be exchanged. Any 3D audio rendering algorithm on any computer platform should technically be able to interpret this format.
Artistic flexibility is paramount to allow creative diversity. Limitations would cause users to reject it.

2.1. The SpatDIF OSC structure

Figure 1 shows how SpatDIF can be integrated into real-time and non-real-time implementations. For controlling spatialization in real-time SpatDIF uses the Open Sound Control protocol (OSC) [7]. OSC is integrated into most audio applications because it is the *de facto* standard for communication across controllers and audio processors. However, the current OSC 1.0 specification lacks in the ability to change properties of the addressed node. Therefore SpatDIF uses an augmented OSC namespace, proposed in [3] and prototyped in Jamoma [1] .
The *SpatDIF Interpreter* moderates between incoming OSC-messages and the executing spatial sound renderer. With respect to initial settings (e.g. source position interpolation) the OSC-messages are translated into low-level commands, tailored to the connected renderer.
If applicable, we propose the use of existing data stor-

[1] `http://www.jamoma.org`

age solutions to focus on conceptual rather than on technical details. The Sound Description Interchange Format (SDIF) and it's XML-extension SDIF-SRL [6], developed for storing sound descriptors, could be used. The spatial properties of each sound could be stored into an SDIF-stream, whereby initial settings as well as other descriptive meta-information (e.g. preferred spatial sound renderer) are attached as annotations. Alternatively, pure XML-based description formats, such as ASDF [2] might also be suitable. Recorded data would be reconverted into OSC streams in order to get processed by the SpatDIF Interpreter.

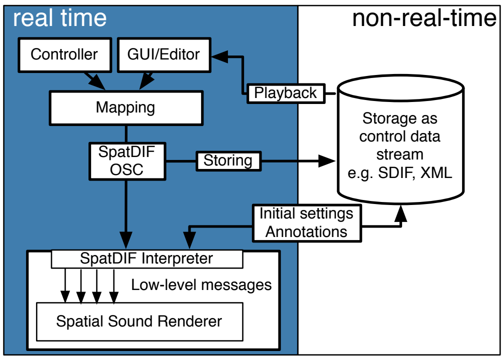

Figure 1. Real-time controlling and storing in SpatDIF

2.2. Prototype

The following examples show the principles of SpatDIF. A change in position of virtual sound sources is the most common controllable parameter in spatial sound rendering. However, there are multiple aspects to consider:

1. Coordinate system: two or three dimensional, cartesian or spherical, left-handed or right-handed;
2. Origin, the zero-reference point of the coordinate system;
3. Absolute or relative (normalized) parameter values;
4. Units, if absolute parameter values are defined;

With the OSC-messages `(1)` a global cartesian coordinate system with x-axis to the right, y-axis to the front, z-axis to the top is created. With message `(2)` the units of this system are defined - here "meter" for each dimension. These are also the default settings. Message `(4)` defines the position of source Nr.1 in cartesian coordinates according to the previous settings. It is left to the user whether the cartesian system and/or polar coordinates are preferred. The lines `(5)+(6)` define a clockwise polar coordinate system (azimuth elevation distance) to set the second source at the same position as the first source (line `(4)`). Often, a change in only one dimension is required: in line `(8)` the azimuth component is changed to an absolute value. Line `(9)` shows an advantage of the OSC structure: all sound sources are incrementally elevated along the z-axis. Conversion factors can be set to enable "scene scaling" and the transformation into normalized positions.

```
(1) /SpatDIF/*/xyz :/def right front top
(2) /SpatDIF/*/xyz :/units meter meter meter
(4) /SpatDIF/source.1/xyz -0.5 0.5 0.0
(5) /SpatDIF/*/aed :/def clockwise
(6) /SpatDIF/*/aed :/units deg deg meter
(7) /SpatDIF/source.2/aed -45.0 0.0 0.707
(8) /SpatDIF/source.1/a 180.0
(9) /SpatDIF/source.*/z:/value/inc
```

Assuming the ICST-Ambisonics renderer [5] is used, the SpatDIF Interpreter would send:

```
(1)  aed 1 -45.0 0 0.707
(7)  aed 2 -45.0 0 0.707
(8)  aed 1 180.0 0 0.707
(9)  aed 1 180.0 54.7 1.22
     aed 2 -45.0 54.7 1.22
```

The low-level messages vary appropriately if another renderer is applied, e.g. VBAP[4].

3. OUTLOOK

For now, SpatDIF handles elementary parameters, such as position and gain values. It should also include other spatial aspects (scene scaling, source type, doppler effect) as well as higher-level parameters, such as reverberance, source width, and distance coding. SpatDIF is under development in two projects related to live electronics: Jamoma, and an NSERC/CCA New Media Initiative project at CIRMMT. Researchers from McGill, the University of Oslo and the Bergen Center for Electronic Arts are involved—others are welcome to join. A webpage `http://www.spatdif.org` is under development.

References

[1] J. Bresson, C. Agon, and G. Assayag. OpenMusic 5: A cross-platform release of the computer-assisted composition environment. In *Proc. 10th Brazilian Symposium on Computer Music*, Belo Horizonte, Brazil, 2005.

[2] M. Geier, J. Ahrens, and S. Spors. The SoundScape Renderer: A Unified Spatial Audio Reproduction Framework for Arbitrary Rendering Methods. In *124th AES Convention*, Amsterdam, The Netherlands, May 2008.

[3] T. Place, T. Lossius, A. R. Jensenius, N. Peters, and P. Baltazar. Addressing Classes by Differentiating Values and Properties in OSC. In *Proceedings of the 2008 conference on New Interfaces for Musical Expression*, Geneva, Switzerland, June 2008.

[4] V. Pulkki. Generic panning tools for MAX/MSP. *Proceedings of International Computer Music Conference*, pages 304–307, 2000.

[5] J. C. Schacher and P. Kocher. Ambisonics Spatialization Tools for Max/MSP. In *Proceedings of the 2006 International Computer Music Conference*, pages 274 – 277, New Orleans, US, 2006.

[6] M. Wright, A. Chaudhary, A. Freed, S. Khoury, A. Momeni, D. Schwarz, and D. Wessel. An XML-based SDIF Stream Relationships Language. *Proc. ICMC*, 2000.

[7] M. Wright, A. Freed, and A. Momeni. Open Sound Control: state of the art 2003. In *Proceedings of the 2003 conference on New Interfaces for Musical Expression*, pages 153–160, Singapore, 2003.