SOFA
Spatially Oriented Format for Acoustics

Piotr Majdak
Acoustics Research Institute, Austrian Academy of Sciences, Vienna ,
Austria
< piotr@majdak.com >

This document defines SOFA version 0.5 .

Further c ontributors:

- **Markus Noisternig** (UMR STMS IRCAM-CNRS-UPMC, Paris, France)
  markus.noisternig@ircam.fr
- **Hagen Wierstorf** (Telekom Innovation Laboratories, Technical University of Berlin, Berlin, Germany) hagen.wierstorf@telekom.de
- **Harald Ziegelwanger** (Acoustics Research Institute, Austrian Academy of Sciences, Vienna, Austria) h.ziegelwanger@me.com
- **Michael Mihocic** (Acoustics Research Institute, Austrian Academy of Sciences, Vienna, Austria)
  michael.mihocic@oeaw.ac.at
- **Thibaut Carpentier** (UMR STMS IRCAM-CNRS-UPMC, Paris, France)
  thibaut.carpentier@ircam.fr
- **Rozenn Nicol** (Orange Labs, France Telecom, Lannion, France)
  rozenn.nicol@orange.com
- **Matthieu Parmentier** (France Television, Paris, France)
  matthieu.parmentier@francetv.fr
- **Agnieszka Roginska** (Music Technology, New York University, NY, USA)
  roginska@nyu.edu

1. WHAT IS SOFA?

Head-related transfer functions (HRTFs) describe the spatial filtering of the incoming sound. So far available HRTFs are stored in various formats, making an exchange of HRTFs difficult because of incompatibilities between the formats. We propose a format for storing HRTFs with a focus on interchangeability and extendability. The **spatially oriented format for acoustics (SOFA)** aims at representing HRTFs in a general way, thus, allowing to store data such as directional room impulse responses (DRIRs) measured with a microphone-array excited by a loudspeaker array. SOFA specifications consider data compression, network transfer, a link to complex room geometries, and aim at simplifying the development of programming interfaces for Matlab, Octave, and C++. SOFA conventions for a consistent description of measurement setups are provided for future HRTF and DRIR databases.

1.1. SOFA version 0.1

We consider the specifications described in the proceedings of the AES convention in Rome, 2013, as SOFA, version 0.1

1.2. SOFA version 0.2

Compared to the version 0.1, the following changes happened:

1.3. SimpleFreeFieldHRTF renamed to SimpleFreeFieldHRIR because FIR is the DataType

1.4. No strings in variables supported yet, strike-through. Some of the variables moved to global attributes.

1.5. Global attributes added: APIVersion, APIName, RoomDescription, ReceiverDescription, SourceDescription, ListenerDescription, TransmitterDescription

1.6. Dimension 'I' added, it represents a singleton dimension (=scalar)

1.7. RoomCorner: unsupported dimension of 2, changed to two variables

1.8. TOAModel: unsupported dimension of 5, not supported yet

1.9. User-defined dimensions not supported yet.

1.10. Dimension number must not change. For addition of optional dimensions, 'I' must be used.

1.11. SOFA conventions SingleRoomDRIR added

1.12. Coordinate systems defined

1.13. SOFA version 0.3

Compared to the previous version, the following specifications have changes:

- Global attribute added: Version
- SOFAConvention renamed to SOFAConventions
- SOFAConventionVersion renamed to SOFAConventionsVersion

1.14. SOFA version 0.4

- Old stuff removed from this document
- Dimensional variables are optional
- Datatype: new added, previous updated
- Define of the types of variables added
- Minor details fixed
- AdditionalRotation and Additiona l Translation added
- ListenerRotation renamed to APV in order to remove the confusion with geometry issues

1.15. SOFA version 0.5

- APV removed, the geometry must be interpreted on loading

- AdditionalRotation and Additional Translation dropped
- Clarification in most of the tables (a dded extras columns)
- Reserved variable names and characters explicitly specified
- Unused parts of specs removed

2. INTRODUCTION

Head-related transfer functions (HRTFs) describe the spatial filtering of the incoming sound due to the listener's anatomy. HRTFs are crucially important for the binaural reproduction of virtual acoustics. HRTFs have been measured by a number of laboratories and are typically stored in each lab's native file format. While the different formats are of advantage for each lab, an exchange of such data is difficult due to incompatibilities between formats.

In this work , we propose specifications for an HRTF – data exchange format with a special focus on interchangeability and extendability. The spatially oriented format for acoustics (SOFA) aims at representing spatial data in a general way, allowing to store not only HRTFs but also more complex data, e.g., directional room impulse responses (DRIRs) measured with a multichannel microphone array excited by a loudspeaker array. In order to simplify the adaption of SOFA for various applications, examples of implementation of the format specifications are provided together with a collection of exemplary data sets converted to SOFA.

The AES-X212 HRTF file format standardization project is based on the SOFA format and was recently approved by the AES subcommittee SC-02 and assigned to the working group SC-02-08 on audio file interchange.

2.1. Typical measurement setups

One of the first publicly available HRTFs were those of a dummy-head microphone measured in an anechoic room [1] . Two microphones placed at the ear simulators were used for the recordings and one loudspeaker was used for the signal excitation. The loudspeaker was moved to the desired elevation and the mannequin was rotated to the desired azimuth. Taken together, HRTFs for 710 spatial positions were measured at elevations from -40° to +90° in steps of 10° and 360° azimuthal range in steps of 5° and a constant distance of 1.4 m. The HRTFs are provided as impulse responses (IRs) with the length of 512 samples at a sampling rate of 44.1 kHz.

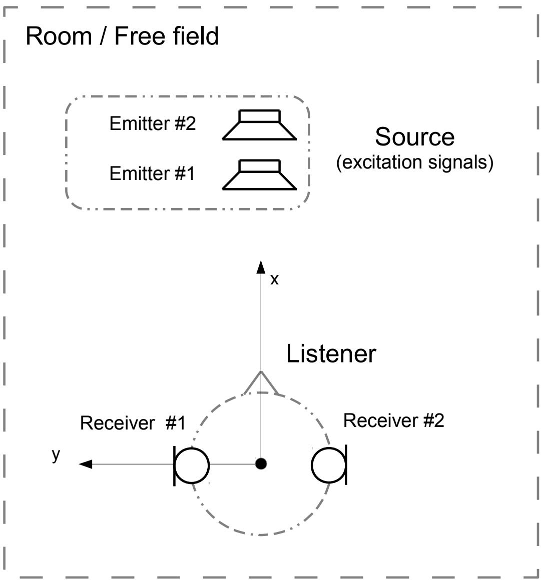

Figure 1: Typical HRTF/DRIR measurement setup.

One of the first publicly available HRTFs measured in *human listeners* was the CIPIC database [2] . The measurements were performed at a constant distance of 1 m for 1250 spatial directions around the listener. The HRTFs are available for 43 listeners as IRs of 200 samples at a sampling rate of 44.1 kHz. Since then many other HRTF/DRIR databases have been made publicly available [3–8] .

All those measurement setups have the following properties in common. In an anechoic chamber or in a room, excitation signals are generated and microphones are used to record the incoming signals (see Fig. 1). The measurement is repeated while varying the spatial position of the excitation source relative to the listener, which is done by varying the position of the listener, the sound source, or both in different dimensions.

Binaural HRTF measurement setups use only two microphones to record the left and right ear signals. However, HRTF/DRIRs measurements may also consider multiple microphones, e.g., three microphones per head side in hearing-assist devices [7] , tens of microphones arranged in an array structure at different directions and distances from the center [9] , a multichannel microphone array arranged around the listeners in a reciprocal HRTF measurement system [10], [11] , multichannel microphone arrays for measuring DRIRs [12] or various microphone positions in a room, e.g., for concert-hall acoustics measurements [13] . As a generalization, microphones and an object comprising those microphones can be identified. Thus, in this article, a microphone as the single receiver of the sound field is called the *receiver* , and the comprising all the receivers is called the *listener ,* see Fig. 1 .

The sound source used for the excitation signal is not necessarily a single point source. Loudspeaker arrays were used, either to control the sound field surrounding the listener, e.g., wave-field synthesis [11], [14], [15] , or higher-order Ambisonics [16], [17] or to control the radiation characteristics of the sound sources [18] . Similarly to the concept of listener and receivers, in this article, the particular sources creating the excitation signal are called *emitters* and the object comprising the emitters is called *source* . Note that a measurement setup with a source with multiple emitters and a listener with multiple receivers has already been considered [19] .

In typical HRTF measurements, only the direction of the incoming signal is varied. In more recent setups also different sound-ear distances have been considered [4], [11], [20] . However, sometimes the variation of other parameters is of interest. For example, HRTFs were measured as a function of the head orientation relative to the torso [21] , or the room IRs were measured as a function of the room temperature [22] . An HRTF file format should thus consider even such parameters.

2.2. Existing data formats

Until now, HRTFs are stored using different formats, all of them having advantages and disadvantages. The CIPIC database [2] provides a file per listener in either a plain text or Matlab (Mathworks, Inc.) file format. The directions are hard coded, i.e., the index of an HRTF corresponds to a predefined direction used in the measurements. While the representation of HRTFs from other directions is not allowed, anthropometric data have been stored within that format. The openDAFF package [1] , while similarly storing HRTFs only in a regular angular distance, uses a key-value system for the description of the metadata which seems to be very promising. Other databases such as LISTEN [3] and ARI [6] , consist of an HRTF matrix and additional matrices describing the direction of the corresponding HRTF, thus, allowing to represent HRTFs from any direction. In that formats, HRTFs from each listener are stored in a separate file. In the database storing the HRTFs as a function of distance [4] , the data are stored in a separate file for each distance. Combined with the necessity to store a separate file for each listener, those three latter formats would result in many files. The MARL-NYU database [23] harmonized the format of CIPIC, LISTEN, MIT, and others databases, and stores all those data in a single file. This concept seems to be promising when combined with a network interface and partial file access in the future. Most of those HRTFs are stored in Matlab formats, i.e., they use a Matlab file convention to store predefined matrices. In contrast, SDIF [24] , a general format for storing audio-related data, has been adapted to HRTFs, allowing to store HRTFs of a single listener in a mixed text-based and binary representation. The concert-hall data [25] , stored as compressed ".wav" files, are another example for a mixed-binary format, which further requires a description (separate text files) in order to being able to interpret the data. The HRTFs measured in rooms (e.g., [8]) are also Matlab files and the relationship between the data and the geometry of the measurement setup is provided in separate publications.

From this summary, the requirements on a file format storing HRTFs and spatially oriented descriptions of acoustic systems are derived:

- Description of a measurement setup with arbitrary geometry, i.e. , not limited to special cases like a regular grid, or a constant distance;
- Self-describing data with a consistent definition, i.e., all the required information about the measurement setup must be provided as metadata in the file;
- Flexibility to describe data of multiple conditions (listeners, distances, etc) in a single file;
- Partial file and network support;
- Available as binary file with data compression for efficient storage and transfer;
- Predefined description conventions for the most common measurement setups.

SOFA aims at fulfilling all those requirements. SOFA specifications are described in the following sections. A HRTF/DRIR measurement setup is described by various objects (Sec. 3.1) and their relations

[1] see http://www.opendaff.org/

(Sec. 3.2). The information is stored in a numeric container (Sec. 3.3) and structured by the *measurement* . Measurement is a discrete sampled observation done at a specific time and under a specific condition. A measurement consists of data, e.g., an IR (Sec. 3.4), and is described by its corresponding dimensions (Sec. 3.5) and metadata (Sec. 3.6). All measurements are stored in a single data structure, e.g., a matrix of IRs.

3. GENERAL SPECIFICATIONS

3.1. Objects

Receiver is any acoustic sensor like the ear or a microphone. The number of receivers in not limited in SOFA and defines the size of the data matrix.

Listener is the object incorporating all the receivers. For HRTFs, a listener can be a head or dummy-head microphone. For DRIRs, a listener represents the microphone-array structure such as a sphere or a frame. Incorporating the receivers in the listener as a single logical object is important because in measurements, usually the orientation and/or position of the listener vary without substantial changes in the head-microphone relation. For example, in measurements done for multiple positions in a room, the position of the head varies and the relation between the head and the microphones does not change. Note that only one listener is considered.

Emitter is any acoustic excitation used for the measurement. The number of emitters is not limited in SOFA. The contribution of the particular emitter is described by the metadata (see later).

Source is the object incorporating all emitters. In SOFA, source might be a multi-driver loudspeaker (with the particular drivers as emitters), or a speaker array (with the particular speakers as emitters), or a choir (with the particular human as emitter), etc. Note that only one source is considered but the source may incorporate an unlimited number of emitters.

Room is the volume enclosing the measurement setup. In the case of a free-field measurement, the room is not considered. An optional room description is considered for measurements performed in reverberant spaces, with a direct description of a simple shoebox, or with a link to a digital asset exchange file for a more complex description.

Optional *Object s* can be described by including user-defined metadata of a measurement. For example, this might be the information about a torso, as in the measurements in which the angle between the torso and the head is varied as an independent variable.

3.2. Relation between the objects

We use two coordinate systems. *Source* and *L istener* are defined in the coordinate system of the room, called global coordinate system. In free field, the global coordinate system is arbitrary.

Emitter s and *R eceiver s* have both their own coordinate system called local coordinate system. The local coordinate system of emitter and receiver are defined relatively to the coordinate system of the source and listener, respectively. With the source and listener in the origin and at default orientation, the local coordinate systems correspond to the global coordinate system.

Two vectors describe the basic orientation of the source/listener: the “view” vector defines the direction in which the source/listener looks; the “up” vector defines the top of the source/listener. “view” and “up” vectors use the same type of coordinate system as that used for the position. In spherical coordinates, the view vector describes the azimuth and elevation angles of the source/listener. The up vector describes the roll, which is usually not considered in HRTF measurements and is optional. If given, we suggest the up vector to be orthogonal to the view-vector. The default basic orientation for the source/listener is the view vector on x-axis and the up vector on z-axis.

In order to be flexible in the future, the way the position and orientations are defined is specified separately for the listener, source, all emitters, and all receivers. The default coordinate type for the position (and thus also view, and up) vectors is the Cartesian (x y z). When the spherical coordinate system is required, the default format is (azimuth, elevation, distance) with units (degree , degree , meter) .

Note the an application is supposed to interpret the geometry according to the application's needs.

3.3. Numeric container

SOFA stores the information in a single file by serializ ing the data in to a binary stream. The serialization is usually done by a numerical container, which defines the format of the binary representation. SOFA files have the extension “.sofa”.

In order to avoid custom development of a numerical container , SOFA relies on netCDF-4 (

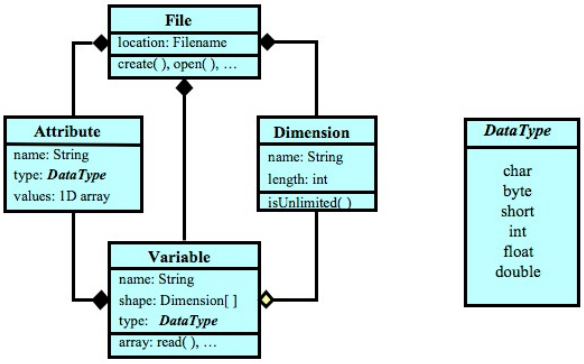

Figure 2: Classic netCDF data model.
Figure from Unidata (http://www.unidata.ucar.edu)

Unidata), which is a set of software libraries and data formats support ing the creation, access, and sharing of scientific data. [2] It is self-describing, n etwork-transparent , and machine-independent; it supports h uge file s , p artial access within a file, and allows for data c ompression. netCDF -4 is widely used in the field of climatology, meteorology, oceanography, and geographic information systems. It is based on the HDF5 (HDF5 Group) [3] , a more basic numerical container , further supported by many institutions worldwide . For SOFA, netCDF offers a structure d representation of multidimensional data and metadata . The open-access specifications are freely available and include a complete definition as well as examples of various implementation s . A pplication - programming interfaces are available as p re-compiled libraries for programming languages like C++, Octave, and JAVA. Note that netCDF is natively supported in Matlab .

netCDF considers *conventions* , a set of recommendations in a community on the naming of attributes, variables, and dimensions within a netCDF file. M any conventions exist , mostly in the field of climate and geographical research. [4] SOFA proposes conventions related to the HRTF/DRIR measurement. In particular, SOFA conventions are proposed for typical HRTF/DRIR measurement setups. According to the netCDF terminology, SOFA defines *dimensions* and stores data in *variables* and *attributes* .

SOFA uses the so-called enhanced data model from netCDF-4, which is based on the classic netCDF data model shown in Fig. 2 . Since the enhanced data model is more complex and not well spread in various computer systems yet, we mostly use the classic data model parts from the enhanced model. This way allows a simple data representation but still full flexibility in the future. More deep knowledge of netCDF format details is not required to read or write netCDF datasets. More interested readers are referred to the User's Manual. [5]

Note that in SOFA, variables of the type " string " are stored as character arrays with a single "string"-dimension, see Sec. 3.5 . Note that this does not correspond to attributes, which are always represented by the netCDF as a special data type .

3.4. Data

Data represent the numeric description of the acoustic systems and consist of a multidimensional matrix of an arbitrary size. Data stored in this format have the flexibility to be in the domain that best accommodates the measurement and measurement system. Data can be time domain finite IRs (data type FIR) or infinite IR filter coefficients (IIRBiquad), with or without sep arately stored broadband delays. The broadband delay (i.e., time-of-arrival, TOA) can be stored as discrete delays in a matrix or as parameters of continuous-directional TOA model [26] . Data contain fields (e.g., Data. F IR, Data.G) which are functions of the dimension N. The interpretation of N depends on the data type, e.g., for IRs , N represents the sampling interval (i.e., inverse of the sampling rate) or the number of FIR – filter taps . The interpretation is denoted in the attributes of the dimension variable N . The different data types and correspon d ing fields are shown in Tab. 1 .

[2] see http://www.unidata.ucar.edu/software/netcdf/

[3] see http://www.hdfgroup.org/HDF5

[4] see http://www.unidata.ucar.edu/software/netcdf- java/formats/UnidataObsConvention.html and http://cf-pcmdi.llnl.gov/

[5] http://www.unidata.ucar.edu/software/netcdf/docs/user_guide.html

FIR

Name	Default	Dimensions	Type	Comment
Data.IR	1	[m R n]	double	impulse responses (along a time axis)
Data.Delay	0	[I R], [M R]	double	broadband delay in the units of N (i.e. , the time axis of FIR) .
Data.SamplingRate	48000	[I], [M]	double	Sampling rate of the IRs and the delay
Data.SamplingRate:Units	hertz	irrelevant	attribute	Unit used for the sampling rate

TF

Name	Default	Dimensions	Type	Comment
Data.Real	1	[m R n]	double	real part of the complex spectrum
Data.Imag	0	[M R N]	double	imaginary part of the complex spectrum
N	0	[N]	double	frequency values
N:LongName	frequency	irrelevant	attribute	
N:Units	hertz	irrelevant	attribute	Unit used for N

Table 1: Data types considered . Dimensions noted in lower case define the corresponding dimension size within the SOFA file.

3.5. Dimensions

Each netCDF variable has fixed dimensions and its dimensions must be defined before creating the variable. Thus, in SOFA, netCDF dimensions are pre-defined, see 2 .

Data and metadata are described by using these dimensions. User-defined dimensions are currently not provided. Throughout this document, the variable sizes are denoted by [$A_1 A_2 \ldots A_I$] where A_i represents the length of the dimension *i* of the *I* -dimensional matrix. We use the Matlab/Octave notation, where the first, second, and third values represent the number of the rows, columns, and third dimension, respectively.

For example, assume a database consisting of one thousand measurements, i.e., M = 1000, obtained for 1000 different positions of the source, i.e., SourcePosition is [M C], using two microphones, i.e., two IR per measurement, and sampling rate of 48 kHz. Further, assume only a single measurement position, i.e. , a single ListenerPosition. This means that Data.IR, SourcePosition, and ListenerPosition will be of dimension [1000 2 3], [1000 3], and [1 3], respectively. Then, in the netCDF file, M = 1000, R = 2, and C = 3. Further, the netCDF variables Data.IR, SourcePosition, and ListenerPosition will have dimensions [M R N], [M C], and [I C], respectively.

In a SOFA file, each dimension size must be uniquely defined and all variables with the corresponding dimension must have that size. To this end, for each dimension, we chose one variable, which size defines the size of the corresponding dimension. In this document, these dimensions are noted as a lower-case letter, see e.g., [m R n] in Tab. 1 .

Variables can have different dimensions. For example, it is possible to provide the ListerPosition as a single entry, meaning that the single ListenerPosition is valid for all measurements. But it is also possible to provide a different ListenerPosition for each measurement. Note that there are restrictions on the variant dimensions:

- The dimensions must be the pre-defined dimensions, see Tab. 2 .

Dimension	Value	Description
I	1	Singleton dimension, defines a scalar value
M	unlimited	Number of measurements, must be integer larger than zero
R	unlimited	Number of receivers, must be integer larger than zero
E	unlimited	Number of emitters, must be integer larger than zero
N	unlimited	Number of data samples describing one measurement, must be integer larger than zero.
S	unlimited	Size of the largest string, must be integer larger than zero
C	3	Coordinate dimension, always three with the meaning depending on the coordinate type

Table 2: Dimensions defined in SOFA.

- The size of the dimensions may change, but the **number** of dimensions, i.e., dimensionality, must not change. In the above example, valid dimensions of the ListenerPositions are [I C] and [M C]. Invalid dimensions would be [C].

Strings are represented as character arrays along the dimension S. When more than one string array is considered in a SOFA file, S represents the size of the array with the longest string dimension.

3.6. M etadata

M etadata consist of variables and their attributes. Numerical variables are multidimensional matrices of the type “double” (i.e. 64 bits floating point data). String variables are saved as character arrays. Other types of variables are not allowed and can be derived from “double” or “string”. Each variable can have its attributes, which are netCDF-attributes. Further, the most important properties of the measurement are valid for the global measurement setup are described by global attributes (see Tab. 4). All metadata names must begin with a letter followed by letters or digits . Note that underscores (“_”) and the metadata names “API”, “GLOBAL”, and “PRIVATE” are not allowed because they are reserved for internal usage in the API . When saved as a variable, date and time uses the number of seconds from 1970-01-01 00:00:00 (Unix time). When saved as attributes, date and time uses a string in the ISO-8601 format “yyyy-mm-dd HH:MM:SS”. Units are lower case.

For the sake of simplicity, nested structures within the metadata are not allowed, but grouping by prefixes using the Pascal convention, e.g., ListenerPosition and Listener View is used.

3.6.1. Global attributes

General metadata are represented as global attributes in netCDF. Global attributes are always strings stored as special data types in a netCDF file. SOFA defines mandatory global attributes (see Tab. 4), further optional (user-defined or defined by a convention) global attributes are allowed. Note that some of the global attributes are read-only, i.e., the API has to provide the correct value s and user is not allowed to change it. Mandatory attributes must be always present .

3.6.2. Object metadata

The information about listener, receiver, source, and emitters is shown in Tabs. 3 to 7 . Some variables are mandatory and some others are proposed but optional.

Name	Type	Dimension(s)	Mandatory	Description	Default
ListenerDescription	attribute	irrelevant	yes	Description of the listener	
ListenerPosition	double	[I C], [M C]	yes	Position	[0 0 0]
ListenerPosition:Type	attribute	irrelevant	yes	Type of coordinate system used for the position	cartesian
ListenerPosition:Unit	attribute	irrelevant	yes	Unit of the coordinates	meter
ListenerView	double	[I C], [M C]		View vector for the orientation	[1 0 0]
ListenerUp	double	[I C], [M C]		Up vector for the orientation	[0 0 1]

Table 3 : Listener v ariables and their attributes.

Name	Default	Type	Read-Only	Comment
Conventions	SOFA	attribute	yes	Defines this file as a SOFA file
Version		attribute	yes	Version of the SOFA specifications. The version is in the form x.y with x the version major and y the version minor
SOFAConventions		attribute	yes	Name of the SOFA convention
SOFAConventionsVersion		attribute	yes	Version of this SOFA convention. The version is in the form x.y with x the version major and y the version minor.
APIName		attribute	yes	Name of the API that created/edited the file
APIVersion		attribute	yes	Version of the API that created/edited the file. The version is in the form x.y.z with x the version major, y the version minor, and z the version release.
ApplicationName		attribute		Name of the application that created/edited the file
ApplicationVersion		attribute		Version of the application that created/edited the file
AuthorContact		attribute		Contact information (e.g. email) of the author
Comment		attribute		Miscellaneous information about the data or methods used to produce it
DataType	FIR	attribute		Defines the data type
History		attribute		Audit trail for modifications to the original data
License	No license provided, ask the author for permission	attribute		Legal license under which the data are provided
Organization		attribute		Legal name of the organization of the author
References		attribute		Published or web-based references that describe the data or methods used to produce it
RoomType	free field	attribute		Defines the room type
Source		attribute		The method of production of the original data. If it was model-generated, source should name the model and its version, as specifically as could be useful. If it is observational, source should characterize it.
TimeCreated		attribute		Date and time of creation of the file, in ISO 8601 format. Will be updated when creating new file.
TimeModified		attribute		Date and time of last modification of the file, in ISO 8601 format. Will be updated each time when saving the file.
Title		attribute		A succinct description of what is in the file

Table 4: General metadata mandatory in SOFA, stored as global attributes in the netCDF file. " Read-only" means that the API has to provide the correct value and user is not allowed to change it.

Name	Type	Dimension(s)	Mandatory	Description	Default
ReceiverDescription	attribute	irrelevant	yes	Description of the receiver	
ReceiverPosition	double	[r C I], [r C M]	yes	Position	[0 0 0]
ReceiverPosition:Type	attribute	irrelevant	yes	Type of coordinate system used for the position	cartesian
ReceiverPosition:Unit	attribute	irrelevant	yes	Unit of the coordinates	meter
ReceiverView	double	[R C I], [R C M]		View vector for the orientation	[1 0 0]
ReceiverUp	double	[rRC I], [R C M]		Up vector for the orientation	[0 0 1]

Table 5 : Receiver v ariables and their attributes.

Name	Type	Dimension(s)	Mandatory	Description	Default
SourceDescription	attribute	irrelevant	yes	Description of the source	
SourcePosition	double	[I C], [M C]	yes	Position	[0 0 0]
SourcePosition:Type	attribute	irrelevant	yes	Type of coordinate system used for the position	cartesian
SourcePosition:Unit	attribute	irrelevant	yes	Unit of the coordinates	meter
SourceView	double	[I C], [M C]		View vector for the orientation	[1 0 0]
SourceUp	double	[I C], [M C]		Up vector for the orientation	[0 0 1]

Table 6 : Source v ariables and their attributes.

Name	Type	Dimension(s)	Mandatory	Description	Default
EmitterDescription	attribute	irrelevant	yes	Description of the emitter	
EmitterPosition	double	[e C I], [eC M]	yes	Position	[0 0 0]
EmitterPosition:Type	attribute	irrelevant	yes	Type of coordinate system used for the position	cartesian
EmitterPosition:Unit	attribute	irrelevant	yes	Unit of the coordinates	meter
EmitterView	double	[E C I], [E C M]		View vector for the orientation	[1 0 0]
EmitterUp	double	[E C I], [E C M]		Up vector for the orientation	[0 0 1]

Table 7 : Emitter v ariables and their attributes.

3.6.3. User-defined metadata

User can provide additional metadata in terms of variables and attributes. User-defined variables must have explicitly defined dimensions using one of the SOFA dimensions. User-defined attributes can be global or can accompany variables.

3.6.4. Room types

Room Type	Parameters	Size	Description	Default
free field	none	-	Data measured under assumed free field conditions	-
reverberant	RoomDescription	attribute	don't know, don't care, something reverberant	-
shoebox	RoomCornerA	[I C], [M C]	Coordinates [A B] of the shoe box	-
	RoomCornerB	[I C], [M C]		cartesian
	RoomCornerA:Type	attribute	Type of coordinate system used for the room	meter
	RoomCornerA:Units	attribute	Units of coordinates	-
	RoomCornerA:Description	attribute	Informal description of the room	

Table 8: Room types. “-”: empty string.

3.7. Coordinate systems

We describe the currently used coordinate systems, and also describe some additional systems, which are not used yet but have been proposed.

3.7.1. Cartesian

x, y, z as a basis

3.7.2. Spherical

Parameter	Range	Front, eye-level	Left, eye-level	Back, eye-level
Azimuth angle	0°...360°	0°	90°	180°
Elevation angle	-90°...90°	0°	0°	0°
Radius	>0	N/A	N/A	N/A

4. SOFA CONVENTIONS

In order to meet the different requirements coming from different application fields, SOFA conventions are specified, i.e., definitions of data and metadata consistently describing particular HRTF/DRIR measurement setups. I nstead of aiming at foreseeing the future, conventions should be developed only for known measurement setups. The known features should be consistently described while not limiting the development of future conventions.

The following SOFA conventions are being discussed. Measured data exist but their description must be fixed in order to create publicly available SOFA files and corresponding software interfaces.

- SimpleFreeFieldHRIR: aimed at storing HRTFs recorded in free field with omnidirectional emitter and source and stored as IRs for a single listener.
- SimpleFreeFieldTF: similar to SimpleFreeFieldHRIR, but uses TF as DataType covering special needs coming from HRTF simulations
- SingleRoomDRIR: Room impulse responses measured with an arbitrary number of receivers (such as a microphone array) and an omnidirectional source in a single room.

The conventions are more roughly explained at http://sofaconventions.org. Further, separate documents describe specifications of each conventions in more detail.

5. TECHNICAL ASPECTS

5.1. Application Programming Interface (API)

SOFA specifications also consider an application-programming interface (API) with similar calls for various programming languages (Matlab, Octave, C++) and computer platforms.

For Matlab and Octave, the API provides functionality to create, read, and write SOFA files, see the **API_MO directory** . The data and metadata are handled in structures considering consistency checks of all information. Numerical data and metadata can be efficiently accessed in whole or in part. Behind the user functions there are two different sets of low-level functions built on top of the netCDF – library support in Matlab and the netCDF Toolbox for Octave.

The SOFA C++ API is quite similar to the Matlab API; it is developed as a layer on top of the C-based netCDF library, see the **API_Cpp directory** .

Currently, SOFA API is in the development phase. The SOFA package with its current development status is accessible at SourceForge . [6] F or debugging and numeric representation of the binary SOFA files, HDF5Viewer is available at the HDF5-Group . [7]

5.2. Networking

A repository is available at http://www.sofacoustics.org/data. Currently, http requests for downloading full SOFA files are supported.

In principle, netCDF files can be also transferred via networks by using the Open Data Access Protocol (OpenDAP), which is a protocol for providing local data to remote locations regardless of local storage format. [8] SOFA, being technically speaking a netCDF convention, should be able to use OpenDAP. The OpeNDAP server will allow partial access of SOFA files via network.

[6] see http://sf.net/projects/sofacoustics

[7] see http://www.hdfgroup.org

[8] see http://opendap.org

6. ACKNOWLEDGMENTS AND REFERENCES

We thank Wolfgang Hrauda for valuable contributions during his internship. This study is supported by the French project Binaural Listening (BiLi, FUI-AAP14) and the Austria n Science Fund (FWF, P 24124-N13).

[1] W. G. Gardner and K. D. Martin, "HRTF measurements of a KEMAR," *J Acoust Soc Am*, vol. 97, pp. 3907–3908, 1995.

[2] V. R. Algazi, R. O. Duda, D. M. Thompson, and C. Avendano, "The CIPIC HRTF database," in *IEEE Workshop on Applications of Signal Processing to Audio and Acoustics* , 2001, pp. 99–102.

[3] O. Warusfel, "LISTEN HRTF Database," 2003. [Online]. Available: http://recherche.ircam.fr/equipes/salles/listen/.

[4] H. Wierstorf, M. Geier, A. Raake, and S. Spors, "A Free Database of Head-Related Impulse Response Measurements in the Horizontal Plane with Multiple Distances," in *130th Convention of the Audio Engineering Society (AES)*, 2011, p. eBrief 6.

[5] T. Nishino, S. Kajita, K. Takeda, and F. Itakura, "Interpolation of head related transfer functions of azimuth and elevation," *J Acoust Soc Jpn*, vol. 57, pp. 685–692, 2001.

[6] P. Majdak, M. J. Goupell, and B. Laback, "3-D localization of virtual sound sources: effects of visual environment, pointing method, and training.," *Attent Percept Psychophys*, vol. 72, no. 2, pp. 454–69, Feb. 2010.

[7] H. Kayser, S.D.Ewert, J.Anemüller, T. Rohdenburg, V. Hohmann, and B. Kollmeier, "Database of Multichannel In-Ear and Behind-the-Ear Head-Related and Binaural Room Impulse Responses," *EURASIP J Advances Sig Proc*, p. Article ID 298605, 10 pages, 2009.

[8] M. Jeub, M. Schäfer, and P. Vary, "A binaural room impulse response database for the evaluation of dereverberation algorithms," in *2009 16th International Conference on Digital Signal Processing*, 2009, pp. 1–5.

[9] I. Balmages and B. Rafaely, "Open-Sphere Designs for Spherical Microphone Arrays," *IEEE Trans Audio Speech Lang Proc*, vol. 15, no. 2, pp. 727–732, Feb. 2007.

[10] D. N. Zotkin, R. Duraiswami, E. Grassi, and N. A. Gumerov, "Fast head-related transfer function measurement via reciprocity," *J Acoust Soc Am*, vol. 120, no. 4, pp. 2202–2215, 2006.

[11] M. Pollow, M., Nguyen, K.-V., Warusfel, O., Carpentier, T., Müller-Trapet, M., Vorländer, M., and Noisternig, "Calculation of Head-Related Transfer Functions for Arbitrary Field Points Using Spherical Harmonics Decomposition," *Acta Acust United Ac*, vol. 89, pp. 72–82, 2012.

[12] B. Khaykin, D., and Rafaely, "Acoustic analysis by spherical microphone array processing of room impulse responses," *J Acoust Soc Am*, vol. 132, pp. 261–270, 2012.

[13] T. Pätynen, J., Tervo, S., and Lokki, "Analysis of concert hall acoustics via visualizations of time-frequency and spatiotemporal responses," *J Acoust Soc Am*, vol. 133, pp. 842–857, 2013.

[14] A. J. Berkhout, "Holographic Approach to Acoustic Sound Control," *J Audio Eng Soc*, vol. 36, pp. 977–995, 1988.

[15] S. Ahrens, J., and Spors, "Wave field synthesis of a sound field described by spherical harmonics expansion coefficients," *J Acoust Soc Am*, vol. 131, pp. 2190–2199, 2012.

[16] M. A. Gerzon, “Ambisonics. Part two: Studio Techniques,” *Studio Sound*, vol. 17, pp. 24–26, 1975.

[17] M. Zotter, F., Pomberger, H., and Noisternig, “Energy-preserving ambisonic decoding,” *Acta Acust United Ac*, vol. 98, pp. 37–47, 2012.

[18] B. Rafaely, “Spherical loudspeaker array for local active control of sound,” *J Acoust Soc Am*, vol. 125, pp. 3006–3017, 2009.

[19] S. Clapp, A. Guthrie, J. Braasch, and N. Xiang, “The use of multi-channel microphone and loudspeaker arrays to evaluate room acoustics,” in *Proceedings of the Acoustics 2012*, 2012, vol. 131, no. 4, p. 3208.

[20] S. Hosoe, K. I. Takanori Nishino, and K. Takeda, “Development of micro-dodecahedral loudspeaker for measuring head-related transfer functions in the proximal region,” in *Proceedings of the IEEE Conference on Audio, Speech and Signal Processing (ICASSP)*, 2006, pp. 329–332.

[21] M. Guldenschuh, A. Sontacchi, and F. Zotter, “HRTF modelling in due consideration variable torso reflections,” in *Proceedings of the Acoustics ’08*, 2008, pp. 99–104.

[22] G. W. Elko, E. Diethorn, and T. Gänsler, “Room impulse response variation due to thermal fluctuation and its impact on acoustic echo cancellation,” in *International Workshop on Acoustic Echo and Noise Control (IWAENC2003)*, 2003.

[23] A. Andreopoulou and A. Roginska, “Towards the Creation of a Standardized HRTF Repository,” in *131th Convention of the Audio Engineering Society (AES)*, 2011, p. Convention Paper 8571.

[24] D. Schwarz and M. Wright, “Extensions and Applications of the SDIF Sound Description Interchange Format,” in *Proceedings of the International Computer Music Conference*, 2000.

[25] J. Merimaa, T. Peltonen, and T. Lokki, “Concert Hall Impulse Responses - Pori, Finland,” 2005. [Online]. Available: http://www.acoustics.hut.fi/projects/poririrs/. [Accessed: 01-Feb-2013].

[26] H. Ziegelwanger and P. Majdak, “Continuous-direction model of the time-of-arrival in the head-related transfer functions,” *J Acoust Soc Am*, vol. submitted, no. submitted, p. submitted.

[27] M. Noisternig, F. Zotter, and B. F. Katz, “Reconstructing sound source directivity in virtual acoustic environments,” in *Principles and Applications of Spatial Hearing*, Y. Suzuki, D. S. Brungart, and H. Kato, Eds. Singapore: World Scientific Publishing, 2011, pp. 357–373.