|  |  |
| --- | --- |
| **Title** | **Annex C of ISO/IEC 23090-34, Immersive audio reference software**  **MPEG-I Immersive Audio Augmented Reality Listener Space Description Format, Version 3** |

# Introduction

This Annex specifies the MPEG-I 6DoF Listener Space Description Format (LSDF). Its purpose is to describe the listening space for the MPEG-I 6DoF audio AR renderer. In AR, where content is augmented on top of real-world objects and spaces, knowledge of the geometry of the listening space is important for realistic reproduction of virtual content within the listening space acoustics. LSDF provides a mechanism to provide the listening space environment information directly to the renderer for MPEG-I 6DoF audio AR evaluation. LSDF additionally provides loudspeaker setup properties parameters that can be used to control loudspeaker rendering. Moreover, the integration of locally captured audio streams into the renderer can be defined.

Figure 1, below, describes the overall concept of the LSDF. The scene information contained in the Encoder Input Format (EIF) [1] is available to the renderer as bitstream. However, for the rendering of AR scenes, information about the listening space, specified in the LSDF, is provided directly to the renderer.



Figure 1. The LSDF is used to provide the information about the listening space to the renderer.

The LSDF includes a subset of elements of the MPEG-I 6DoF Audio Encoder Input Format (EIF) [1] with restrictions. The elements are used to describe the walls, ceiling and floor of the listening space. Furthermore, the LSDF describes anchors for aligning elements in the scene EIF to their positions in the listening space. The elements used in the LSDF are listed in Section 2.

When using the MPEG-I Audio Evaluation Platform (AEP) [2], an LSDF file is placed in a specific folder inside the AEP folder structure. The AEP provides the path to the Max External plug-in containing the renderer. See the AEP documentation for more information [2].

# LSDF elements

Table 1 summarizes the elements that are used in the LSDF for describing the listening space. The elements are explained in more detail below. Since the LSDF does not go through an encoder, but rather directly to the renderer, some restrictions have been added to the <Mesh> element to keep the listening space geometry simple.

|  |  |
| --- | --- |
| **Element** | **Purpose** |
| <AudioScene> | Defines the listening space audio scene |
| <Mesh>, <Face>, <Vertex>, <AcousticMaterial> | Defines the walls, floor and ceiling |
| <AcousticEnvironment>, <AcousticParameters>, <Frequency> | Defines the acoustic characteristic of the listening space with parameters (RT60, rdr, …) or a room impulse response (RIR) |
| <ARAnchor> | Defines the alignment of EIF elements to real-world objects |
| <LoudspeakerSetup> | Defines a loudspeaker setup for loudspeaker rendering |
| <AudioStream> | Defines a locally captured signal |
| <ObjectSource> | Defines a locally captured audio source in the scene |

Table 1. Summary of elements in LSDF

## **Audio scene**

The <AudioScene> element declares the AR listening space. It is similar to the <AudioScene> described in the EIF [1], but with changes to the list of allowed child node entities and their counts. A single <Mesh> and a single <AcousticEnvironment> are allowed. A new child node, <ARAnchor>, has been added.

Like in the EIF ([1, §4.6]), by default, an element’s position and orientation are expressed in its parent entities’ coordinate space. The global cartesian coordinate space is defined by the root entity, that is the audio scene itself. Any other entity with a position and an orientation defines its own local coordinate space.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| <AudioScene> | | | | |
| Declares an audio scene. Each scene description must have exactly one <AudioScene> node. This node marks the root node of the static description of the audio scene. It contains all entities of the scene as children. | | | | |
| Child node | Count | Description | | |
| <Mesh> | 0.. 1 | Mesh (see below) | | |
| <AcousticMaterial> | >=0 | Acoustic material (see below) | | |
| <AcousticEnvironment> | 1 | Acoustic environment (see below) | | |
| <ARAnchor> | >=0 | AR anchor (see below) | | |
| <LoudspeakerSetup> | 0..1 | Loudspeaker setup (see below) | | |
| <AudioStream> | >=0 | AudioStream for locally captured audio (see below) | | |
| <ObjectSource> | >=0 | ObjectSource for locally captured audio (see below) | | |
| Attribute | Type | Flags | Default | Description |
| id | ID | R |  | Identifier |

## **Mesh**

The <Mesh> element is used to describe the walls, ceiling and floor of the listening space.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| <Mesh> | | | | |
| Declares a triangle mesh. A mesh consists of a list of vertices (3D coordinates) and a number of triangular faces (i.e. the indices of three vertices). Meshes can be used to describe an arbitrary geometry. Any <Mesh> node has to have one or more <Vertex> or <Face> child nodes, defining points and triangles. The mesh is required to be a manifold mesh. | | | | |
| Child node | Count | Description | | |
| <Vertex> | >=1, <= 36 | Vertex (see EIF [1]) | | |
| <Face> | >=1, <= 48 | Face (see EIF [1]) | | |
| Attribute | Type | Flags | Default | Description |
| id | ID | R |  | Identifier |
| position | Position | O | (0, 0, 0) | Position (origin of the mesh) |

Vertex and Face are as specified in the EIF [1].

A manifold mesh can be checked for by checking that all vertices are manifold by verifying that ([2]):

* every edge is shared by exactly two faces
* every vertex has a single, complete loop of triangles around it.

**Acoustic Environment**

<AcousticEnvironment> and <AcousticParameters> are similar to the corresponding nodes in the EIF [1] with certain differences required for describing real (physical) room acoustics.

The <AcousticEnvironment> in the LSDF must contain either <AcousticParameters> or <RoomImpulseResponse> but not both.

Since <AcousticEnvironment> does not contain a position, its origin corresponds to the origin of the LSDF root entity <AudioScene>.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| <AcousticEnvironment> | | | | |
| Declares an acoustic environment. The acoustic characteristics of the environment are provided by either <AcousticParameters> or <RoomImpulseResponse>, optionally at several points in space. Either <AcousticParameters> or <RoomImpulseResponse> need to be provided but not both. Additionally, a bounding volume can be specified. | | | | |
| Child node | Count | Description | | |
| <AcousticParameters> | >=0 | Acoustic parameters. See the description below. | | |
| <RoomImpulseResponse> | >=0 | Room impulse response (see below) | | |
| Attribute | Type | Flags | Default | Description |
| id | ID | R |  | Identifier |
| region | Geometry ID | R | none | Region in which properties are specified. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| <AcousticParameters> | | | | |
| Inside of an <AcousticEnvironment> node, this declares the environment’s acoustic behavior at a specific point in space. | | | | |
| Child node | Count | Description | | |
| <Frequency> | >=1 | Frequency data specification (see below) | | |
| Attribute | Type | Flags | Default | Description |
| position | Position | O |  | Position (Required if multiple <AcousticParameters> are specified within one <AcousticEnvironment>) |
| diffuseOnset | Float | O | none | A time offset, in seconds, after which the impulse response is considered to have become diffuse, that is, when it can be considered sufficient to model it statistically. If diffuseOnset is not provided, a default value is derived based on the dimensions of region. |
| complexityLevel | Integer | O | 3 | ComplexityLevel of reverb in the <AcousticEnvironment> on a five-point scale 1, 2, …, 5. The value of 3 causes the default reverb settings to be used providing a good quality/complexity compromise for most situations. Values 1 and 2 correspond to computationally lighter configurations having also less echoes and density in reverb. Correspondingly, values 4 and 5 create an increasingly dense but also more complex reverb suitable for cases where reverb quality is very important |

Note: the definition of diffuseOnset differs from the EIF definition of predelay given in [3] which is in terms of the longest room dimension. The diffuseOnset as defined here is shorter than what is obtained based on the definition in [3]. For monophonic impulse responses, the term mixing time is often used of the time where the RIR becomes diffuse.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| <Frequency> | | | | |
| Inside of a <AcousticParameters> node, which is inside an <AcousticEnvironment> node, this declares the environment’s acoustic behavior at a specific frequency. | | | | |
| Attribute | Type | Flags | Default | Description |
| f | Float | R |  | Frequency in Hertz |
| rt60 | Float | R |  | Reverberation time (RT60) in seconds |
| rdr | Float | O | none | Reverberant-to-Direct ratio measured from diffuseOnset onwards. Note: rdr can only be specified, if diffuseOnset is specified. |

Note: The specification of the <Frequency> node, within <AcousticParameters>, differs from the EIF [1] in that it specifies a Reverberant-to-Direct ratio (RDR) instead of a DSR attribute.

The details of the RDR are described in [3].

A recording of a spatial or mono room impulse response can be provided instead of acoustic parameters to describe the listening space acoustics. The data of the measurement is carried in <RoomImpulseResponse>.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| <RoomImpulseResponse> | | | | |
| Inside of an <AcousticEnvironment> node, this declares a room impulse response (RIR). The RIR can be monophonic or spatial (SRIR). | | | | |
| Attribute | Type | Flags | Default | Description |
| id | ID | R |  | Identifier |
| file | String | R |  | Path of the RIR audio file in the Microsoft WAV format. |
| format | String | R | “ACN-SN3D” | Format of the RIR data in the file. One of “mono”, “ACN-SN3D” or “ACN-N3D”. The Ambisonic channel ordering must be ACN and the normalization must be either SN3D or N3D as indicated by the format string. Ambisonic orders from first to fourth are supported. |
| micPosition | Position | O | None | Position of the the RIR measurement microphone as Cartesian coordinates |
| micOrientation | Rotation | O | (0°, 0°, 0°) | Orientation of the RIR measurement microphone |
| sourcePosition | Position | O | None | Position of the source (loudspeaker or other sound emitter) used for the RIR measurement in Cartesian coordinates |
| sourceOrientation | Rotation | O | None | Orientation of the source (loudspeaker or other sound emitter) for the RIR measurement |
| directivity | Integer | O | 1 | An integer, 0 or 1. 0 indicates an omnidirectional measurement source, 1 indicates a typical loudspeaker directivity with forward emphasis with increasing frequency. |
| complexityLevel | Integer | O | 0 | The complexity level which scales the number of delay lines (output channels) used in the reverberator as well as the number of directions analyzed for directional gain, in the case that the RIR is spatial. Integer values from 1 to 5 correspond to delay line counts of 3, 7, 15, 31, 63, respectively. A value of 0 (the default) specifies a 15-channel design. |

Note: the coordinate system is the OpenGL Cartesian coordinate system as in the EIF [1, §A.1]. If the micPosition and sourcePosition are not provided, the Reverberant-to-Direct Ratio (RDR) cannot be measured from the RIR and is instead calculated using a theoretical model informed by the bounding region of the <AcousticEnvironment>. If the if the micOrientation is not provided, it is assumed to be oriented forward, in the negative Z direction.

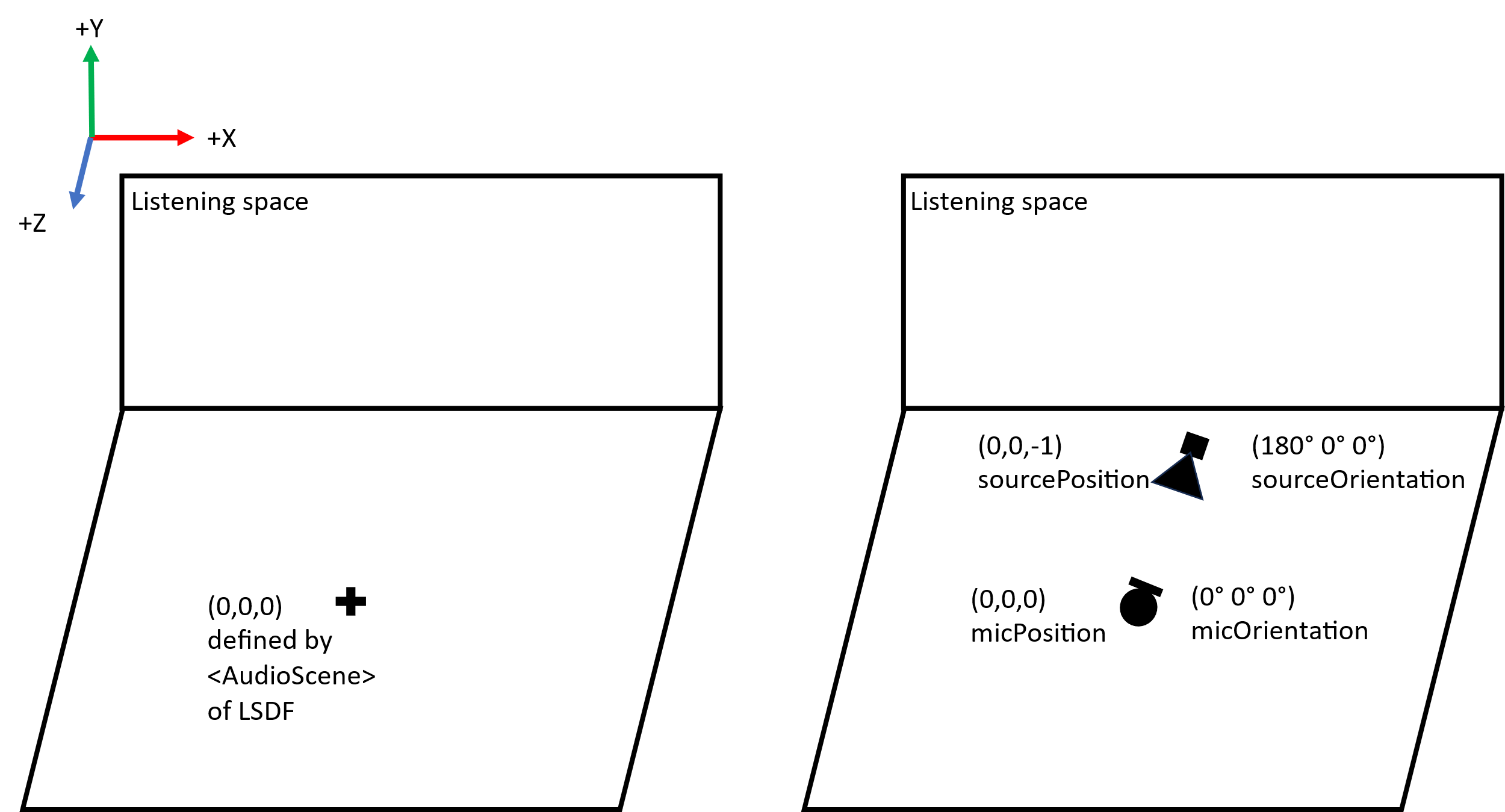


Figure 2. Illustration of the coordinate system of the LSDF (left), and the default position and orientation of the source and microphone in the <RoomImpulseResponse> (right).

Note: when providing a <RoomImpulseResponse> it is encouraged to specify a region, defined as a <Box> primitive, passed to the <AcousticEnvironment>. This <Box> can be interpreted as an approximation of the dimensions of the space in which the (S)RIR is recorded, offering useful inference of acoustical features to the renderer, such as the proximity of the source and microphone to room boundaries (according to the sourcePosition and micPosition).

**Anchors**

An <ARAnchor> is used to indicate a real-world position for the EIF to reference.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| <ARAnchor> | | | | |
| Indicates the position and orientation of a real-world anchor. <AnchorObject> in EIF may refer to this object for correct placement of scene elements. | | | | |
| Attribute | Type | Flags | Default | Description |
| id | ID | R |  | Identifier |
| position | Position | R |  | Position of the anchor object w.r.t the origin |
| orientation | Rotation | R |  | Orientation of the anchor object w.r.t to the origin |

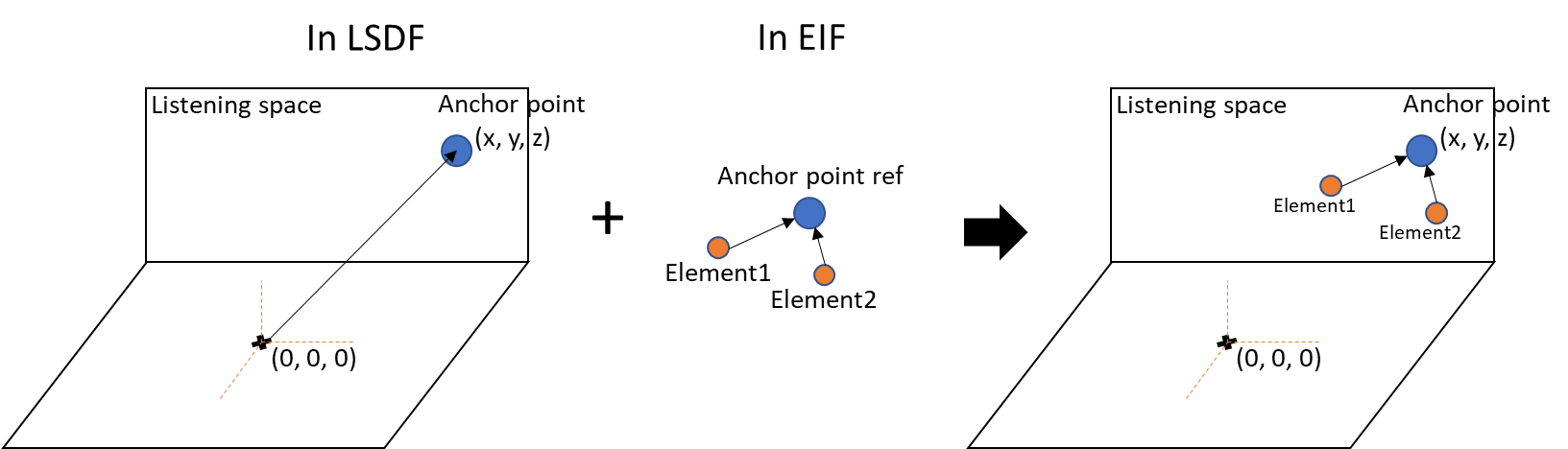


Figure 3. Anchor mechanism. <ARAnchor>s in the LSDF describe points in the listening space which may be referred to in the EIF using <AnchorObject>s [1].

For anchors referring to a position on a wall, ceiling or floor of the listening space, a convention shall be used where the anchor is oriented such that it is pointing outside of the room (perpendicular to the wall, ceiling or floor).

**Loudspeaker Setup**

The <LoudspeakerSetup> is used to specify the loudspeaker setup in the listening space that is used for loudspeaker rendering.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| <LoudspeakerSetup> | | | | |
| Declares a loudspeaker setup. The <LoudspeakerSetup> has a position/orientation in space. | | | | |
| Child node | Count | Description | | |
| <ReferencePoint> | 0..1 | Reference point (see below) | | |
| <Loudspeaker> | >=1 | Loudspeaker (see below) | | |
| <DirectivityCompensation> | 0..1 | Loudspeaker directivity compensation (see below) | | |
| <LevelCompensation> | 0..1 | Loudspeaker level compensation (see below) | | |
| Attribute | Type | Flags | Default | Description |
| position | Position | R |  | Position |
| orientation | Rotation | O | (0° 0° 0°) | Orientation |
| inputLayout | CICP layout | R |  | Speaker layout of the original audio signal |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| <ReferencePoint> | | | | |
| Inside a <LoudspeakerSetup> node, this declares its reference point, sometimes also called “sweet spot”. The position and orientation are specified relative to the position/orientation of the parent <LoudspeakerSetup>. This also holds information whether the loudspeaker setup is calibrated. If no <ReferencePoint> element is present, the reference point is at the position of the <LoudspeakerSetup>. | | | | |
| Attribute | Type | Flags | Default | Description |
| position | Position | R |  | Position |
| calibrated | Boolean | O | False | Is the loudspeaker system calibrated? |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| <Loudspeaker> | | | | |
| Inside a <LoudspeakerSetup> node, it declares a single loudspeaker. The position and orientation are specified relative to the position/orientation of the parent <LoudspeakerSetup>. Loudspeakers are not modifiable at runtime. | | | | |
| Attribute | Type | Flags | Default | Description |
| position | Position | R |  | Position |
| orientation | Rotation | O | Towards reference point | Orientation |
| channel | Integer | R |  | Channel index |

When no orientation is given, the default rotation must be computed given the loudspeaker position ls\_pos and the reference point ref\_pos according to the following algorithm:

minus\_z = ref\_pos – ls\_pos

If the length of minus\_z is less than 0.000001, an error is raised because the reference point is not allowed to be that close to a loudspeaker.

minus\_z /= norm(minus\_z)

up = [0.0, 1.0, 0.0]

If the dot product of minus\_z and up is greater than 0.999999, the loudspeaker rotation is set to yaw/pitch/roll values of [0°, 90°, 0°].

If the dot product of minus\_z and up is less than -0.999999, the loudspeaker rotation is set to yaw/pitch/roll values of [0°, -90°, 0°].

x = cross(minus\_z, up)

x /= norm(x)

y = cross(x, minus\_z)

The columns of the rotation matrix of the loudspeaker are set to [x, y, -minus\_z].

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| <DirectivityCompensation> | | | | |
| Inside a <LoudspeakerSetup> node, it declares a directivity compensation filter to be used for each loudspeaker. If no <DirectivityCompensation> element is present, a filter with the default values is used. The filter can be deactivated by setting eqFreq to 0. | | | | |
| Attribute | Type | Flags | Default | Description |
| eqFreq | Float | O | 9000 | High-shelving cut-off frequency (Hz) |
| eqGainDb | Float | O | 6 | High-shelving gain (dB) |
| eqAngleStart | Float | O | 20 | Above this off-axis angle (in degrees) the filter starts having an effect |
| eqAngleRange | Float | O | 30 | Within this range (in degrees) the filter gain changes from 0 to the given eqGainDb. For off-axis angles greater than eqAngleStart + eqAngleRange the filter gain stays at eqGainDb. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| <LevelCompensation> | | | | |
| Inside a <LoudspeakerSetup> node, it controls the level compensation to be used for each loudspeaker. If no <LevelCompensation> element is present, default values are used. | | | | |
| Attribute | Type | Flags | Default | Description |
| slope1 | Float | O | 8.0 | Nearfield sound decay per distance doubling (dB) |
| slope2 | Float | O | 0.0 | Farfield sound decay per distance doubling (dB) |
| criticalDistance | Float | O | 2.7 | Critical distance (m) |
| beta | Float | O | 5.0 | Nearfield-farfield transition |

**Locally Captured Audio Interface**

Two specific elements support the integration of locally captured elements into a scene to be rendered. These elements allow for the definition of signals and sources to be used to accommodate real-time captured audio. The elements have a syntax that fits their counterparts in the EIF definition.

This interface is implementation dependent. The method that is used to define it can be adapted by a device manufacturer to best fit a given implementation or appliance.

**AudioStream**

This element represents a single audio input that is local to the user. There can be as many AudioStream elements as the user requires.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| <AudioStream> | | | | |
| Declares an audio stream provided by an input local to the user. LSDF defined <ObjectSource>s can use it as signal source. Optionally, the input can be routed to a bitstream defined source if that is explicitly allowed in the authoring of the scene by the isLocallyCaptured flag. This element only supports single channel inputs. | | | | |
| Attribute | Type | Flags | Default | Description |
| id | ID | R |  | Identifier |
| localInputChannel | Integer | R |  | Index of the input channel where the local audio is provided to the renderer. |
| overrides | ID | O |  | ID of the Bitstream defined stream to be overridden by this element input. The referenced <AudioStream> must be flagged as isLocallyCaptured. |

**ObjectSource**

This element defines a point source to specify the position of local streams in the scene. To enable a best possible listening experience for the listener voice input, individually adjustable extra parameters are available to adapt the rendering to the properties of the local setup and to the listener preference: listenerVoice, directGainDb, erGainDB.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| <ObjectSource> | | | | |
| Declares an <ObjectSource> which emits sound into the virtual scene. The <ObjectSource> has a position/orientation in space. The source radiates omnidirectionally. The signal component of the <ObjectSource> must be also defined in the LSDF. | | | | |
| Attribute | Type | Flags | Default | Description |
| id | ID | R |  | Identifier |
| position | Position | R, M |  | Position. If listenerVoice=True default to average position of human voice relative to the userCoordinateSystem (0.00 -0.10 0.00). |
| orientation | Rotation | O, M | (0° 0° 0°) | Orientation |
| cspace | Coordinate space | O | relative | Spatial frame of reference. If listenerVoice=True this parameter is ignored and cspace is set to listener relative space. |
| active | Boolean | O, M | true | If true, rendering of this source is enabled |
| gainDb | Gain | O, M | 0 | Gain (dB) |
| refDistance | Float > 0 | O | 1 | Reference distance (m) (detailed in EIF [1] section 3.2.4) |
| signal | AudioStream ID | R, M |  | ID for the LSDF defined Audio stream that provides the signal for this source. |
| directiveness | Value | O, M | 1 | Directiveness (as defined in EIF [1] section 3.6.1) |
| aparams | Authoring parameters | O | none | Authoring parameters (as defined in EIF [1] section 4.13) |
| vdlMethod | String | O | “spline” | Defines the Variable Delay Line method to be used (“spline”, “linear”, “spline\_erLinear”) |
| reverbGainDb | Gain | O, M | -6 | Additional gain (dB) to be used for reverb rendering in all acoustic environments.  It shall range between -119.0 to +12.0. |
| directGainDb | Gain | O, M | -17 | Only used if listenerVoice=true. Additional gain (dB) to be used for direct sound level tuning. It shall range between -119.0 to +12.0. |
| erGainDB | Gain | O, M | -13 | Only used if listenerVoice=true. Additional gain (dB) to be used for early reflections level tuning.  Range between -119.0 to +12.0. |
| listenerVoice | Boolean | O | False | Tags the source as the Listener’s Voice. |

# LSDF creation

An LSDF is defined for each listening space. Creation of the LSDF starts by defining an origin in the listening space, usually a spot on the floor close to the center of the listening space. The room geometry is then measured w.r.t the origin and this information is inserted to the single mesh in the LSDF file. All anchors required by the different AR scenes to be rendered at a given listening environment are then defined in the LSDF file, again w.r.t the chosen origin. Description of the required anchors are to be provided by the scene authors along with the scenes. Further information (RT60, material coefficients, etc.) of the listening space may be added using the elements described in Section 2.

# References

[1] Annex B of ISO/IEC 23090-34, Immersive audio reference software (MPEG-I Encoder Input Format)

[2] ‘MPEG-I Immersive Audio Documentation for the Audio Evaluation Platform’

[3] ISO/IEC JTC1/SC29/WG6 N0247, ‘CD of ISOIEC 23090-4, MPEG-I Immersive Audio’, MPEG 146