



**International
Standard**

ISO/IEC 23090-31

**Information technology — Coded
representation of immersive media —**

**Part 31:
Haptics coding**

*Technologies de l'information — Représentation codée de média
immersifs —*

Partie 31: Codage haptique

**First edition
2025-01**

Reference number
ISO/IEC 23090-31:2025(en)

Full document details: [ISO_IEC 23090-31_2025](#)


© ISO/IEC 2025

Documents
Document Sharing Hub



COPYRIGHT PROTECTED DOCUMENT

© ISO/IEC 2025

All rights reserved. Unless otherwise specified, or required in the context of its implementation, no part of this publication may be reproduced or utilized otherwise in any form or by any means, electronic or mechanical, including photocopying, or posting on the internet or an intranet, without prior written permission. Permission can be requested from either ISO at the address below or ISO's member body in the country of the requester.

ISO copyright office
CP 401 • Ch. de Blandonnet 8
CH-1214 Vernier, Geneva
Phone: +41 22 749 01 11
Email: copyright@iso.org
Website: www.iso.org

Published in Switzerland

Contents

Page

Foreword	v
Introduction	vi
1 Scope	1
2 Normative references	1
3 Terms, definitions and abbreviated terms	1
3.1 Terms and definitions	1
3.2 Abbreviated terms	3
3.3 Mnemonics	4
4 Overview and architecture	4
4.1 Overview	4
4.2 Codec architecture	4
5 Data model	5
5.1 Data structure overview	5
5.2 Haptic experience	7
5.3 Haptic avatar	8
5.4 Haptic perception	9
5.5 Haptic device	11
5.6 Haptic channel	12
5.6.1 General	12
5.6.2 Custom mesh avatar	13
5.6.3 Body part mask	14
5.6.4 Semantic body part and actuator mapping	15
5.7 Haptic band	19
5.8 Haptic effect	23
5.9 Haptic keyframe	26
6 Interchange file format	27
6.1 Overview	27
6.2 HJIF Specifications	28
6.2.1 MPEG_haptics	28
6.2.2 MPEG_haptics.avatar	29
6.2.3 MPEG_haptics.perception	30
6.2.4 MPEG_haptics.sync	32
6.2.5 MPEG_haptics.reference_device	32
6.2.6 MPEG_haptics.channel	33
6.2.7 MPEG_haptics.vector	34
6.2.8 MPEG_haptics.band	35
6.2.9 MPEG_haptics.effect	36
6.2.10 MPEG_haptics.keyframe	37
7 MPEG-I haptic stream (MIHS) format	37
7.1 Overview	37
7.1.1 General	37
7.1.2 Initialization units	39
7.1.3 Temporal and spatial units	39
7.1.4 MIHS packets	41
7.2 Syntax and semantics	41
7.2.1 mpegIHapticStream()	41
7.2.2 mpegIHapticUnit()	42
7.2.3 mpegIHapticPacket()	43
7.2.4 MIHSPacketPayload()	44
7.2.5 readMetadataInitializationTiming()	45
7.2.6 readMetadataTiming()	46
7.2.7 readMetadataExperience()	47

7.2.8	readAvatar()	48
7.2.9	readMetadataPerception()	48
7.2.10	readReferenceDevice()	50
7.2.11	readMetadataChannel()	53
7.2.12	readMetadataBand()	55
7.2.13	readLibrary()	56
7.2.14	readLibraryEffect()	57
7.2.15	readData()	58
7.2.16	readEffect()	59
7.2.17	readEffectBasis()	63
7.2.18	readKeyframe()	64
7.2.19	readTransientKeyframe()	64
7.2.20	readCurveKeyframe()	64
7.2.21	readVectorialKeyFrame()	65
7.2.22	readWaveletEffect()	65
7.2.23	readCRC()	67
7.3	Description of MIHSPacketType	68
7.3.1	InitializationTiming	68
7.3.2	Timing	68
7.3.3	MetadataHaptics	68
7.3.4	MetadataPerception	69
7.3.5	MetadataChannel	69
7.3.6	MetadataBand	70
7.3.7	LibraryEffect	70
7.3.8	Data	71
7.3.9	CRC16 and CRC32	72
7.3.10	GlobalCRC16 and GlobalCRC32	72
7.4	Application examples	73
7.4.1	Initialization units	73
7.4.2	Temporal and spatial units	73
7.4.3	Silent units	73
7.4.4	CRC packets	74
7.5	Random access support with MIHS (informative)	74
8	Processing model	75
8.1	Overview	75
8.2	Encoder (informative)	75
8.2.1	Encoder architecture	75
8.2.2	OHM metadata input file	76
8.2.3	Descriptive input files	76
8.2.4	PCM input file	76
8.2.5	Transcoding descriptive content	77
8.2.6	Frequency band decomposition	79
8.2.7	Keyframe extraction for low frequencies processing	80
8.2.8	Wavelet encoding	81
8.3	Decoder	94
8.3.1	Decoder Architecture	94
8.3.2	Metadata and parametric data decoding	94
8.3.3	Wavelet Decoding	94
8.3.4	Random access decode	100
8.4	Synthesizer (informative)	100
Annex A (normative) JSON schema reference		103
Annex B (informative) OHM metadata input file format		104
Annex C (informative) Semantic body part and actuator mapping		106
Annex D (normative) Profiles and Levels		110
Bibliography		112

Foreword

ISO (the International Organization for Standardization) and IEC (the International Electrotechnical Commission) form the specialized system for worldwide standardization. National bodies that are members of ISO or IEC participate in the development of International Standards through technical committees established by the respective organization to deal with particular fields of technical activity. ISO and IEC technical committees collaborate in fields of mutual interest. Other international organizations, governmental and non-governmental, in liaison with ISO and IEC, also take part in the work.

The procedures used to develop this document and those intended for its further maintenance are described in the ISO/IEC Directives, Part 1. In particular, the different approval criteria needed for the different types of document should be noted. This document was drafted in accordance with the editorial rules of the ISO/IEC Directives, Part 2 (see www.iso.org/directives or www.iec.ch/members_experts/refdocs).

ISO and IEC draw attention to the possibility that the implementation of this document may involve the use of (a) patent(s). ISO and IEC take no position concerning the evidence, validity or applicability of any claimed patent rights in respect thereof. As of the date of publication of this document, ISO and IEC had not received notice of (a) patent(s) which may be required to implement this document. However, implementers are cautioned that this may not represent the latest information, which may be obtained from the patent database available at www.iso.org/patents and <https://patents.iec.ch>. ISO and IEC shall not be held responsible for identifying any or all such patent rights.

Any trade name used in this document is information given for the convenience of users and does not constitute an endorsement.

For an explanation of the voluntary nature of standards, the meaning of ISO specific terms and expressions related to conformity assessment, as well as information about ISO's adherence to the World Trade Organization (WTO) principles in the Technical Barriers to Trade (TBT) see www.iso.org/iso/foreword.html. In the IEC, see www.iec.ch/understanding-standards.

This document was prepared by Joint Technical Committee ISO/IEC JTC 1, *Information technology*, Subcommittee SC 29, *Coding of audio, picture, multimedia and hypermedia information*.

A list of all parts in the ISO/IEC 23090 series can be found on the ISO and IEC websites.

Any feedback or questions on this document should be directed to the user's national standards body. A complete listing of these bodies can be found at www.iso.org/members.html and www.iec.ch/national-committees.

Introduction

Haptics provide an additional layer of entertainment and sensory immersion to the user. Therefore, the user experience and enjoyment of media content, from a ISOBMFF files, broadcast channel, streaming games or mobile advertisements can be significantly enhanced by the judicious addition of haptics to the audio/video content. To that end, haptics has been proposed as a first-order media type, akin to audio and video, in ISOBMFF. Further, haptics has also been proposed as an addition to ISO/IEC 23009-1 (MPEG-DASH) to signal the presence of haptics in MP4 segments to DASH streaming clients. Lastly, the ISO/IEC 23090 (MPEG-I) use cases have been augmented with haptics

Haptics digital encoding is the storing of tactile data in a digital format. As with audio and video, digital encoding is of fundamental importance to allow digital haptic devices to function. Haptics encoding gained relevance with the increased market importance of wideband haptics in consumer peripherals such as smartphones with haptic engines and game consoles with haptic enabled controllers. The prior generation of haptics peripherals was based on less expressive haptic actuators usually based on state machine control processes.

In the field of haptics, the signal encoding usually takes one of two approaches:

- Quantized: This representation is generally made from measured data. The samples from the original phenomenon are stored at a specific acquisition frequency inside the file to represent this signal. One example of a quantized haptic signal is proposed through WAV files, originally developed for audio. WAV file formalism allows the capture of real-world data and the representation of complex wideband haptic feedback. This type of haptics encoding has the disadvantage of being difficult to modify once encoded due to the inability to access the primitives used to create the signal.
- Descriptive: This representation is used to encode haptic signals as a combination of functions to be synthesized. Examples of such vectorized formats include AHAP and IVS. These formats have the advantage of being created with a composition of primitives. They are easily modifiable at runtime by an application and by dedicated editing tools. Currently, these solutions support only vibrotactile perception, but can easily be extended for other forms of haptics such as kinaesthetic, temperature and textures. They also tend to be memory inefficient with increasing signal complexity and cannot encode non-periodic phenomena

This document describes the coded representation allowing to encode both descriptive and quantized data in a human readable JSON format (.hjif) used as an exchange format. This format can be compressed and packetized into a binary file format for distribution and streaming purposes (.hmpg).

Information technology — Coded representation of immersive media —

Part 31: Haptics coding

1 Scope

This document specifies technology that supports the efficient transmission and rendering of haptic signals for the playback of immersive experiences in a wide variety of scenarios. The document describes in detail a robust coded representation of haptic media covering the two most popular haptic perceptions leveraged by devices today: vibrotactile and kinaesthetic. Support for other haptic modalities has also been integrated.

The coded representation allows to encode both descriptive and quantized data in a human readable JSON format used for exchange purposes, and a compressed bitstream version, optimised for memory usage and distribution purposes. This approach also allows to meet the expectations for compatibility with both descriptive and quantized formats, as required by the market, as well as interoperability between devices for 3D immersive experiences, mobile applications and other distribution purposes.

Information provided in this document related to the decoder is normative, while information related to the encoder and renderer is informative.

2 Normative references

The following documents are referred to in the text in such a way that some or all of their content constitutes requirements of this document. For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments) applies.

ISO 8601-1, *Date and time — Representations for information interchange — Part 1: Basic rules*

ISO 8601-2, *Date and time — Representations for information interchange — Part 2: Extensions*

ISO/IEC 21778:2017, *Information technology — The JSON data interchange syntax*

IETF RFC 3986, *Uniform Resource Identifier (URI): Generic Syntax*, available at: <https://www.rfc-editor.org/info/rfc3986>

IETF RFC 8259, *The JavaScript Object Notation (JSON) Data Interchange Format*, available at: <https://www.rfc-editor.org/info/rfc8259>

IETF RFC 4648, *The Base16, Base32, and Base64 Data Encodings*, available at: <https://www.rfc-editor.org/info/rfc4648>

3 Terms, definitions and abbreviated terms

3.1 Terms and definitions

For the purposes of this document, the following terms and definitions apply.

ISO and IEC maintain terminology databases for use in standardization at the following addresses:

— ISO Online browsing platform: available at <https://www.iso.org/obp>

— IEC Electropedia: available at <https://www.electropedia.org/>

3.1.1

actuator

component of a device for rendering haptic sensations

3.1.2

avatar

body (or part of body) representation

3.1.3

band

component in a channel for containing effects for a specific range of frequencies

3.1.4

channel

component in a perception containing one or more bands rendered on a device at a specific body location

3.1.5

device

physical system having one or more actuators configured to render a haptic sensation corresponding with a given signal

3.1.6

effect

component of a band for defining a signal, consisting of a haptic waveform or one or more haptic keyframes

3.1.7

experience

top level haptic component containing perceptions and metadata

3.1.8

haptics

tactile sensations

3.1.9

keyframe

component of an effect mapping a position in time or space to an effect parameter such as amplitude or frequency

3.1.10

metadata

global information about an experience, perception, channel or band

3.1.11

MIHS format

self-contained stream for transporting MPEG-I haptic data

3.1.12

MIHS initialization unit

MIHS unit containing metadata necessary to reset and initialize a haptic decoder

3.1.13

MIHS packet

MIHS data packet which includes metadata or binary effect data

3.1.14

MIHS unit

MIHS data unit covering a duration of time

3.1.15

modality

type of haptics, such as vibration, force, pressure, position, velocity or temperature

3.1.16

perception

haptic perception containing one or more channels of a specific modality

3.1.17

signal

representation of the haptics associated with a specific modality to be rendered on a device

3.1.18

transient

short momentary effect defined by an amplitude and, directly or indirectly, a frequency

3.1.19

vector

direction in space which can be used for haptic signals spatialization

3.2 Abbreviated terms

AC	Arithmetic coding/coder
AHAP	Apple Haptic and Audio Pattern - JSON-like file format that specifies a haptic pattern
ATSC	Advanced Television Systems Committee
CDF9/7	Cohen–Daubechies–Feauveau 9/7
CRC	Cyclic redundancy check
DASH	Dynamic Adaptive Streaming over HTTP (specified in ISO/IEC 23009)
FFT	Fast Fourier Transform
HJIF	Haptics JSON Interchange Format
ID	Identifier
ISOBMFF	ISO Base media file format (specified in ISO/IEC 14496-12)
IVS	XML-based file format for representing haptic effects
JSON	JavaScript Object Notation
LOD	Level of Detail
MIHS	MPEG-I Haptic Stream
MPEG	Moving Picture Experts Group
MPEG-I	MPEG Immersive media (specified in this and other parts of ISO/IEC 23090)
OHM	Object Haptic Metadata - Text file format for haptics metadata
PCM	Pulse-code modulation
SPIHT	Set Partitioning In Hierarchical Trees

3.3 Mnemonics

The following data types are used for data specifications:

bslbf	Bit string with left bit first, where “left” is the order in which bit strings are written. Bit strings are written as a string of 1s and 0s within single quote marks, for example ‘10000001’.
vlcs8	Variable length character string. Contains string data stored as a character array encoded in UTF-8.
uimsbf	Unsigned integer with most significant bit first.
imsbf	Integer with most significant bit first.
vlclbf	Variable length code with left bit first, where “left” refers to the order in which the variable length codes are written.
duimsbf	Decimal stored as unsigned integer with most significant bit first in a given range. Default range is [-1,1]
islif	Integer stream with left integer first, where “left” is the order in which integer strings are written. Only for buffering.
vlislf	Variable length integer stream with left integer first, where “left” refers to the order in which the integers are written.

4 Overview and architecture

4.1 Overview

This document describes a coded representation of haptics based on a data model that enables the encoding of both descriptive and quantized haptic data. Two complementary formats based on this shared data model ([Clause 5](#)) are detailed in the specifications: an interchange format (.hjif) detailed in [Clause 6](#), and a packetized compressed binary format for streaming that can also be stored in a binary file (.hmpg) detailed in [Clause 7](#). In addition, a normative decoder and an informative encoder and synthesizer are described in detail.

The .hjif format is a human-readable format based on JSON and is not optimized for memory usage. It can easily be parsed and manually edited, which makes it an ideal interchange format, especially when designing or creating content. For distribution purposes, the .hjif data can be compressed and packetized into a more memory efficient binary .hmpg bitstream. This compression is lossy, with different parameters impacting the encoding depth of amplitudes and frequencies composing the bitstream. The compressed and packetized data can be directly distributed as a MPEG-I haptic stream (MIHS).

4.2 Codec architecture

The generic codec architecture can process both waveform PCM signals (WAV) and descriptive haptic files such as AHAP, IVS or HJIF, the proposed MPEG format. Metadata information is provided to the codec through OHM input files (described in [Annex B](#)). An overview of the codec architecture is depicted in [Figure 1](#), and a more detailed description of the input files and encoder architecture is provided in [Clause 8](#).

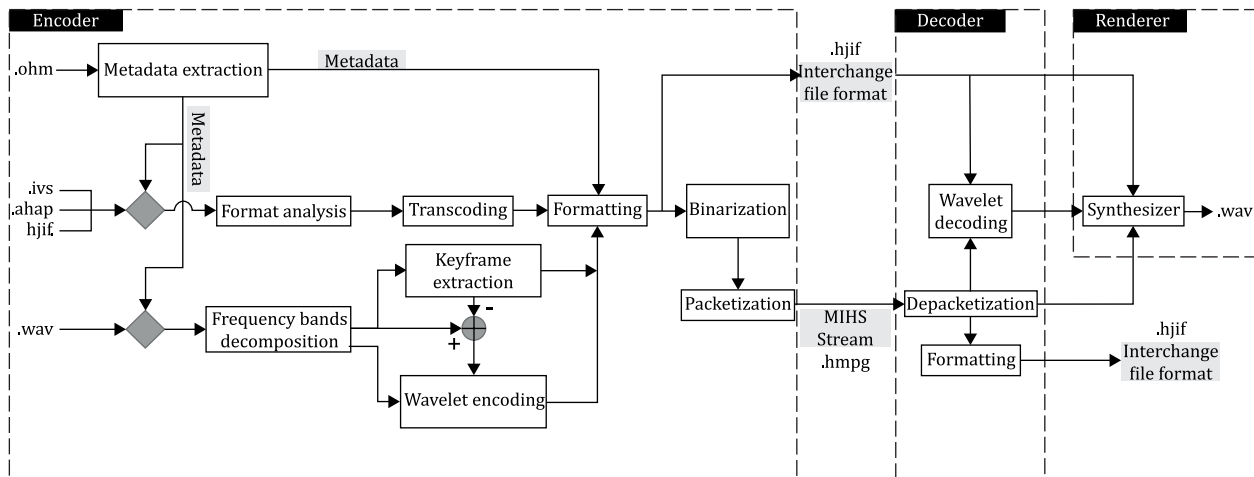


Figure 1 — Overview of the codec architecture

The encoder processes the two types of input files differently. For descriptive content, the input is analysed semantically to transcode (if necessary) the data into the proposed coded representation. For PCM content, the signal analysis process is split into two sub-processes. After performing a frequency band decomposition on the signal, the different bands (one or more) can be encoded as a set of keyframes. In [Figure 1](#), an efficient example is given where the low frequencies are encoded using a keyframe extraction process. The low frequency band(s) is(are) then reconstructed, and the error between this signal and the original low frequency signal is computed. This residual signal is then added to the original high frequency band(s), before encoding using wavelet transforms. This multi-bands hybrid encoding scheme is detailed in [Clause 8](#).

The encoder can output two types of formats: an interchange file format (.hjif) encoded in JSON, and a binary encoded streaming format (MIHS). that can also be stored as a binary file (.hmpg). The two formats have complementary purposes, and a lossy one-to-one conversion can be operated between them. The encoder is informative and is detailed in [subclause 8.2](#).

The decoder takes as input a binary .hmpg file (the MIHS bitstream) or a .hjif file and either outputs a .hjif file or directly synthesizes the haptic data. Haptic data contained in a .hjif file can be rendered directly on haptic devices or using an intermediate synthesizer generating PCM data. The decoder is normative and is detailed in [subclause 8.3](#).

The synthesizer allows to render haptic data from a .hjif input file or MIHS stream into a PCM output file. The synthesizer is informative and is detailed in [subclause 8.3.4](#).

5 Data model

5.1 Data structure overview

This Clause focuses on the data model of the coded representation of haptics. It specifies the information required by a synthesizer to render the haptic data. The following subclauses provide detailed definitions for every property of each element of the data model depicted in [Figure 2](#). The data structure introduced in this Clause is shared by the interchange format and the streaming format defined respectively in [Clauses 6](#) and [7](#).

The data structure of the two formats follows the hierarchical organization illustrated in [Figure 2](#).

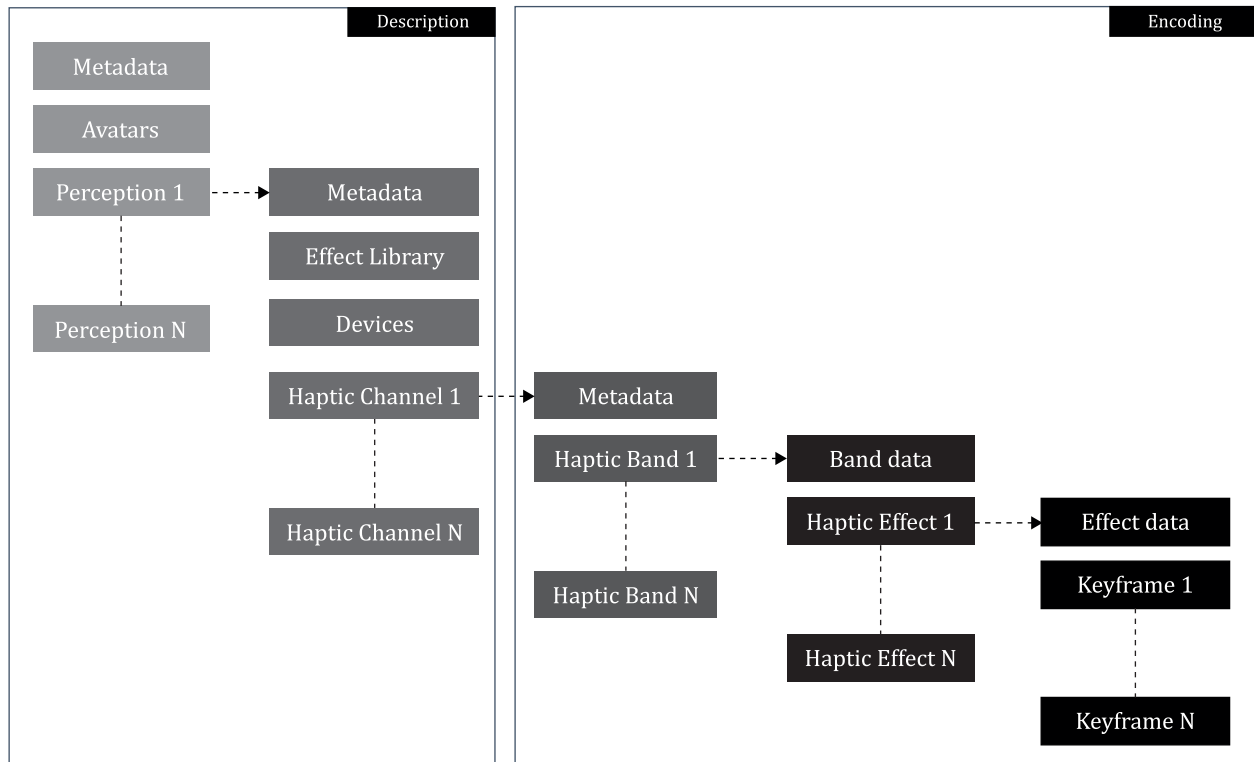
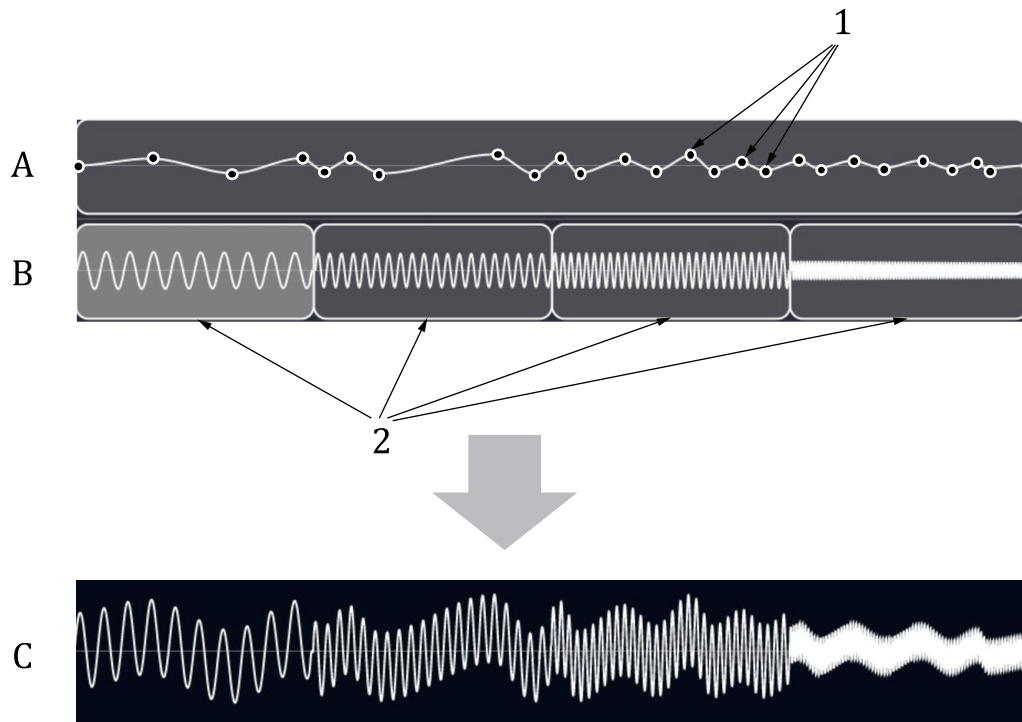


Figure 2 — General data model

The highest level of the structure describes the entire haptic experience defined in the file or stream. It contains some high-level metadata information. It also provides a list of avatars (i.e., body representation) referenced later in the file to specify the desired location of haptic stimuli on the body. Finally, the haptics are described through a list of perceptions. These perceptions correspond to haptic signals associated with specific perception modalities (e.g., vibration, force, position, velocity, temperature).

In addition to specific metadata, a perception contains a list of channels. The data in each channel are decomposed into one or more frequency bands. Each band defines part of the signal in a frequency range. A band is described by a list of haptic effects, each containing a list of keyframes. The haptic signal in a channel can then be reconstructed by combining the data in the different bands as illustrated in [Figure 3](#). For instance, by adding the high and low frequency bands in this figure, the original signal can be reconstructed.

The perception and channel information describe the content (Description part of [Figure 2](#)), while the bands, effects and keyframes contain the data of the encoded signals (Encoding part of [Figure 2](#)).

**Key**

- 1 keyframes
- 2 effects
- A band 1 (*curve band*)
- B band 2 (*vectorial wave band*)
- C channel 1

Figure 3 — Haptic channel (bottom) and its decomposition in two frequency bands (top)

The format proposes four types of bands: transient bands, curve bands, vectorial wave bands and wavelet wave bands. Each band is composed of a series of effects of the same type as the band, each defined by a list of keyframes. The data contained in the effects and keyframes are interpreted differently for different types of bands.

5.2 Haptic experience

The haptic experience defines the root of the hierarchical data model. It provides information on the date of the file, the version of the format and the encoding profile and level. It also describes the haptic experience, lists the different avatars (i.e., body representations) used throughout the experience, and defines all the haptic perceptions. [Table 1](#) details the properties of a haptic experience.

Table 1 — Haptic experience properties

Property	Description
version	Year of the edition and amendment of ISO/IEC 23090-31 that this file conforms to, in the following format: XXXX or XXXX-Y, where XXXX is the year of publication and Y is the amendment number, if any.
profile	Name of the profile used to generate the encoded stream as defined in the normative Annex D .
level	Number of the level used to generate the encoded stream as defined in the normative Annex D .
date	Creation date of this haptic experience.
description	A user description of this haptic experience.

Table 1 (continued)

Property	Description
timescale	Number of ticks per second.
avatars	List of avatars defining body representations used in the haptic experience. See subclause 5.3 .
perceptions	List of perceptions describing a haptic signal. See subclause 5.4 .

5.3 Haptic avatar

Haptic avatars are used as body representations. Avatars can reference a custom 3D mesh from a companion file. Each perception may be associated with an avatar, which allows spatialization of effects at the channel level. The same avatar can be used by multiple perceptions.

A 3D mesh can provide resolution and accuracy with variable vertex density depending on the application. For instance, the density can be representative of the spatial acuity of a specific perception modality ([Figure 4](#)). The format of custom 3D meshes is out of the scope of this document.

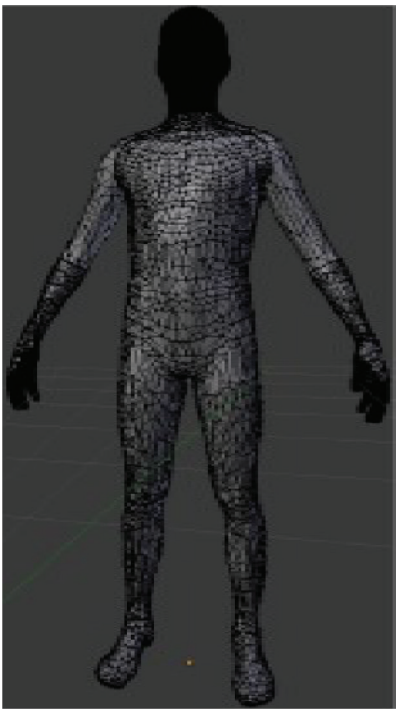


Figure 4 — Example of haptic avatar body representations

[Table 2](#) defines the list of properties of a haptic avatar:

Table 2 — Haptic avatar properties

Property	Description
id	Unique ID of the avatar in this experience.
lod	If the avatar uses a mesh with several levels of detail (LODs), this indicates which LOD should be used for the avatar.
type	Type of haptic perception represented by the avatar. This is related to the spatial acuity of the corresponding haptic modality. The avatar type may be: — Vibration: the avatar is for vibrotactile signals. — Pressure: the avatar is for pressure signals. — Temperature: the avatar is for temperature signals. — Custom: the avatar is a custom avatar. The mesh is provided as a companion file using the mesh URI. The definition of custom mesh is out of the scope of this document.
mesh	URI to access the associated 3D mesh file. The URI conforms to the syntax defined in RFC 3986.

5.4 Haptic perception

A haptic perception is a haptic signal associated with a specific perception modality. The format supports different types of modalities, encoded in function of time (pressure, acceleration, velocity, position, temperature, vibrotactile, water, wind, force, electrotactile) or space (vibrotactile texture, stiffness, friction). The list of supported modalities is provided in [Table 4](#) with the corresponding units and type (aka temporal or spatial).

There is no accepted psychophysical representation set of tactile perception in function of a definite representation space and stimulus. Touch shows remarkable complexity in the relations between physical stimuli and the semantic assigned by humans.

In the case of haptics (touch stimulus mediated by technology) this is somewhat different as haptics is mediated by actuators which have a well-defined set of physical capabilities and representation spaces.

The concept of haptic perception in this specification proposes a simple construct which can be easily understood by implementers and designers alike tying haptic device capabilities to the most similar tactile perception, or at least the one that has most in common with the haptic representation.

For each haptic perception, metadata information is provided on the modality, the corresponding avatar representation and technical characteristics of targeted or compatible haptic devices.

The data associated with a perception may contain multiple channels. A channel is associated with a body location and usually corresponds to a haptic device; for instance, a vibrotactile suit with 16 channels corresponding to 16 vibrotactile actuators or a gamepad with one channel corresponding to a force feedback trigger.

A haptic perception may also contain an effect library.

[Table 3](#) describes the full list of properties of a haptic perception.

Table 3 — Haptic perception properties

Property	Description
id	Unique ID of the perception in this experience.
perception modality	Type of perception modality of the haptic signal. The detailed list of perception modalities and the associated units is given in Table 4 .
description	Description of the haptic data contained in the perception.
priority	Importance of the perception for scalability. A lower value indicates higher priority. Given a limited bandwidth, decoders may ignore perception with a higher priority value.
avatar id	Unique ID of the associated avatar.

Table 3 (continued)

Property	Description
effect library	List of haptic effects as defined in subclause 5.8 . Each effect from the library has an ID and may be referenced directly in a band (see subclause 5.7).
semantic scheme	Scheme URN of the semantic description of effects. The default scheme is identified with <code>urn:mpeg:mpeg:haptics:effectsemantic:2023</code> and is defined in Semantic keyword is a two-layers hierarchical metadata structure illustrated in Table 13 . It may be added as a supplementary information to an effect. At most only one layer 1 and one layer 2 semantic keywords shall be included with one effect. These keywords indicate the desired designer intention. When a presentation engine is incapable of rendering the exact specified effect it may decide to render a similar effect or not render at all. One example is a gunshot designed for a VR controller with a specific frequency range. A VR gun prop may be developed with its own haptics library. Table 13
reference devices	List of targeted haptic reference devices or actuators used for this haptic perception. More details on reference devices are given in subclause 5.5 .
channels	List of haptic channels comprising this perception. More details on haptic channels are given in subclause 5.6 .
unit exponent	Exponent of the powers of 10 for the SI unit of effect positions and keyframe positions of spatial modalities. This property specifies which measurement unit is used to encode the given perception. By default, the considered value is -3 (i.e. millimeters).
perception unit exponent	Exponent of the powers of 10 for the SI unit measure of the dependent variable. This property specifies which measurement unit is used to output the given perception. By default, the considered value is 0. For example, if the perception modality is set to stiffness and this exponent is set to 0, the perception experience is encoded in Newton.

Table 4 — Perception modalities and corresponding units

Modality	Perception unit	SI unit	Temporal (T)/Spatial(S)
Pressure	Pa	s	T
Acceleration	m/s ²	s	T
Velocity	m/s	s	T
Position	m	s	T
Temperature	°C	s	T
Vibrotactile	Normalized to [-1,1]	s	T
Water	m ³	s	T
Wind	m/s	s	T
Force	N	s	T
Electrotactile	Normalized to [-1,1]	s	T
Vibrotactile Texture	Normalized to [-1,1]	m	S
Stiffness	N/m	m	S
Friction	Normalized to [-1,1]	m	S
Humidity	Normalized to -1/1	s	T
User-defined Temporal	Normalized to [-1,1]	s	T
User-defined Spatial	Normalized to [-1,1]	m	S

The haptic library defined at the perception level allows to define a set of shared haptic effects at the perception level. It consists of a list of effects that can be referenced from a band. This allows to avoid unnecessary repetition of identical effects. The effect library is particularly useful for content creators and avoids the duplication of effects inside channels.

As illustrated in [Figure 5](#), the effect library can be used to reduce the memory footprint of haptic data. The same experience can be stored in an optimized manner using a single haptic effect in the library. It is then referenced multiple times at the band level versus storing this effect multiple times directly at the band level.

As described in [Figure 5](#), factoring and linearization can be used to convert from a library-based stream to a simple repetitive-based stream of haptic effects.

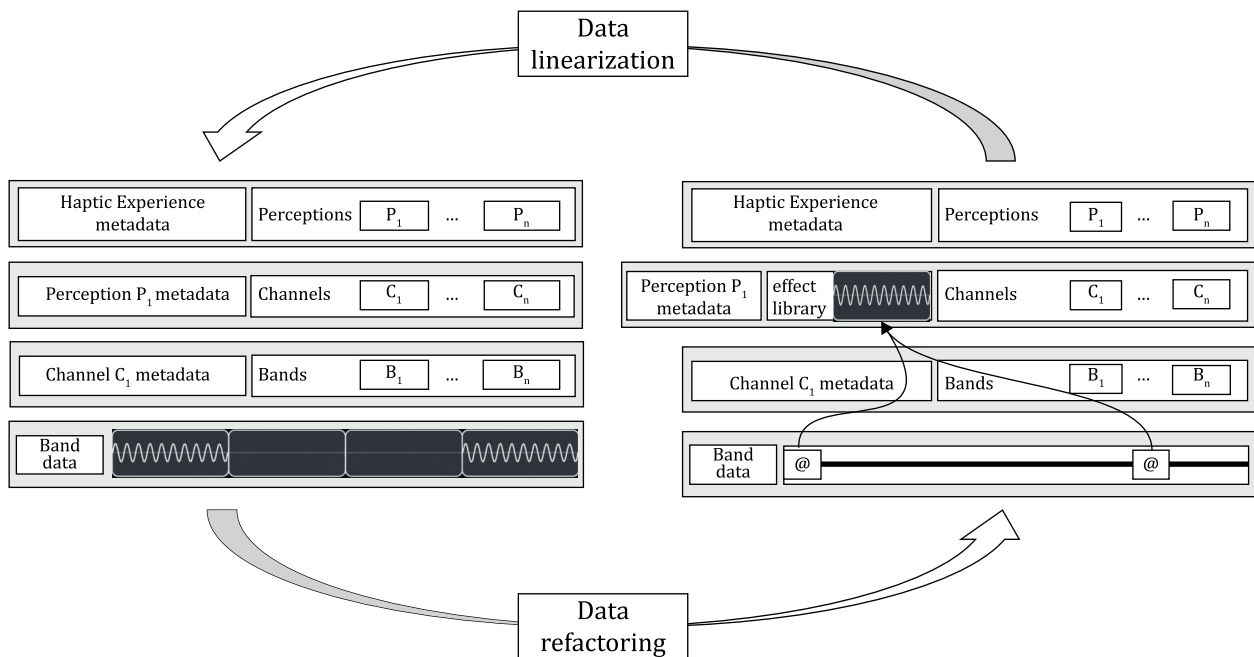


Figure 5 — Example of haptic files containing the same data with and without using the effect library

5.5 Haptic device

A haptic experience is usually defined with a reference setup to validate the experience, or with a number of specific targeted haptic devices in mind (compatible devices). If the experience is played back on different devices with different capabilities, the associated encoded signal may have to be rendered differently. To perform such adaptation, the capabilities of the original device(s) (reference or compatible devices) are needed. For this purpose, each haptic perception defines a list of reference devices (with their detailed characteristics) and each haptic channel may reference the corresponding device.

A haptic reference device is described through a list of characteristics, including the type of device, the frequency range of the device, the maximum voltage of the device. The detailed list of properties is specified in [Table 5](#).

Table 5 — Haptic device properties

Property	Description
id	Unique ID of the device in this experience. This value is used for referencing the device in a channel.
name	Name of the device, for the purpose of human identification.
body part mask	Binary mask specifying the location of the device or actuator on the body based on the body segmentation detailed in Table 7 .
maximum frequency	Maximum frequency of the actuator (Hz).
minimum frequency	Minimum frequency of the actuator (Hz).
resonance frequency	Resonance frequency of the actuator (Hz).
maximum amplitude	Maximum amplitude value of the targeted device according to the perception modality; for instance, the maximum acceleration speed if the perception modality is the acceleration. The corresponding unit is specified in Table 4 in subclause 5.4 .
impedance	Impedance of the actuator (Ω).
maximum voltage	Maximum voltage of the actuator (V).

Table 5 (continued)

Property	Description
maximum current	Maximum current of the actuator (A).
maximum displacement	Maximum displacement of the actuator (mm).
weight	Mass of the device (kg).
size	Bounding sphere diameter of the device (mm).
custom	User-defined data. This parameter may be used to specify additional properties of the targeted device.
type	Type of actuator. Possible types are: — LRA — VCA — ERM — Piezo — Unknown (for other modalities)

5.6 Haptic channel

5.6.1 General

Haptic signals can be encoded on multiple channels. Typically, a haptic channel defines a signal to be rendered at a specific body location with a dedicated actuator or device. Metadata stored at the channel level includes information such as the gain associated with the channel, the mixing coefficient, the desired body location of the haptic feedback, optional reference device and optional direction. Additional information such as the desired sampling frequency or sample count can also be provided. Finally, the haptic data of a channel are contained in a set of haptic bands defined by their frequency range.

The list of properties of a haptic channel is detailed in [Table 6](#).

Table 6 — Haptic channel properties

Property	Description
id	Unique ID of the channel. The identifier is unique among all channels in the perception containing this channel.
description	Description of the channel.
priority	Importance of the channel for scalability. A lower value indicates higher priority. Given a limited bandwidth, decoders may ignore perception with a higher priority value.
reference device id	Targeted reference device from the list defined in the perception.
gain	Gain associated with the channel to adapt the normalized encoded data values to a typical device, according to: $V = gain * x,$ where x corresponds to the normalized encoded data.
mixing coefficient	Weight of the channel when mixing different channels together to produce a mixed signal. The resulting signal is given by: $V = \frac{\sum V_i * coefficient_i}{\sum coefficient_i},$ where V_i corresponds to the signal of channel i . A mixing coefficient of 0 indicates that the channel is not mixed.

Table 6 (continued)

Property	Description
body part mask	Binary mask specifying the location of the effect on the body as defined in Table 7 . The binary mask 0x0 indicates that the body part is not specified. The application can render the effect anywhere. The mask 0xFFFFFFFF corresponds to the full body. It means that the effect is applied on the whole body; for instance, a background effect.
actuator resolution	Reference actuator resolution used to design the haptic experience. This value linked to body part target and actuator target can be used together as an experience spatialization model on the human body.
body part target	Identification of a unique body part or group of body parts on the human body. Table 9 describes all the possible values which can be stored here to construct the targeting command.
actuator target	List of different coordinates to target actuators on the previously identified human body parts.
frequency sampling	Sampling frequency of the original encoded signal (Hz). This may be used by the synthesizer to reconstruct the original signal. However, the synthesizer may sample the output signal at another sampling frequency.
sample count	Present if the frequency sampling value is greater than zero. It is the number of samples of the original encoded signal. This can be used along with the frequency sampling by the synthesizer to ensure that the output signal has the same size and duration as the original file.
vertices	List of the vertices from the avatar impacted by the effect. More precisely, it is the list of indices of the vertices from the mesh associated with the avatar of the perception. If the avatar does not specify a mesh, this field should be ignored. The vertices impacted by the effects of this channel are the body locations where the effects should be applied. The appropriate avatar representation is referenced by the avatar ID indicated at the perception level (see subclause 5.4).
bands	List of haptic bands comprising the channel. A channel can include one or several bands. A band corresponds to a frequency bandwidth as specified in subclause 5.7 . If the bands array is empty, it corresponds to a channel without any haptic effect. As illustrated in Figure 3 , the haptic signal of a channel is the sum of the signals in each band.
direction	Specifies a spatial direction for the channel as detailed in subclause 5.6.3 . This direction metadata should only be used with haptic modalities dependent on space (i.e. Vibrotactile Texture, Stiffness and Friction). It indicates a preferred rendering direction of the haptic perception of the targeted body part. It can be composed with X, Y and Z following the formalism for unit vectors to indicate any direction in the 3D space. Each integer value stored in this vector will be transformed from its initial range $[-127,127]$ to the $[-1,1]$ range to interpret this vector as unitary.

The direction property identifies the preferred rendering direction of the spatialized haptic experience. The direction is related to the relative position of the body part in its own coordinate system. The information is required to specify the rendering of a haptic effect in a predetermined direction.

The spatialization of the haptic stimuli associated with a channel on the body can be specified either by referencing a set of vertices (with the vertices property) from a custom avatar mesh ([Figure 4](#)), or using specific localization maps. Two types of maps are defined, using a body part mask based on the body segmentation illustrated in [Figure 6](#) with 32 defined parts, or using a semantic body parts identification combined with a map of actuators on these body parts ([Figure 7](#)). The body part mask provides a lightweight method to define a rough localization of the effects on the user's body. It does not specify how to map the effect to the device(s), which is left to the device manufacturer. The intent is to provide a simple and low data-rate mechanism to localize effects. Alternatively, the body part target is a more complex representation providing both localization and mapping to device(s). The hierarchical body representation is related to the human skeleton, while the mask only relates to the surface of the body. An additional actuator representation provides the localization and resolution of the device actuator(s) and allows interpolation, when necessary, to adapt the effect to the device configuration (as described in [Annex C](#)). Multiple methods can be combined since each addresses different goals and usages.

5.6.2 Custom mesh avatar

Consists of a set of 3D vertices, with their 3D coordinates such as [Figure 4](#).

5.6.3 Body part mask

The mask has been defined based on [Figure 6](#) segmentation of the human body.

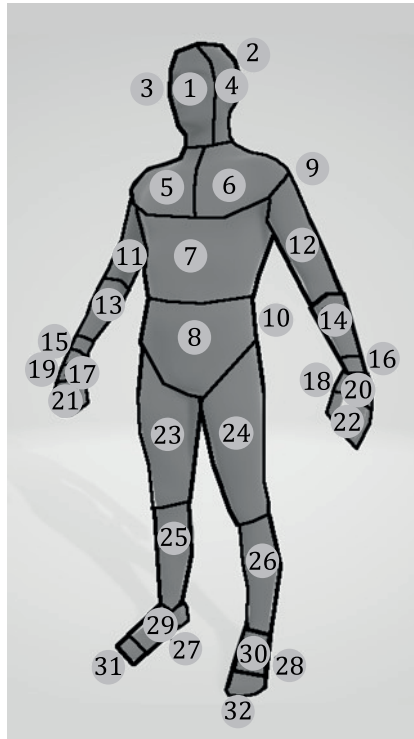


Figure 6 — Body part segmentation

The 32 body parts and their corresponding masks are detailed in [Table 7](#). These binary masks can be logically combined to associate haptics stimuli to multiple body parts. [Table 8](#) gives some common body parts combinations.

Table 7 — Body part masks

	Name	Body_part_mask (binary)	Hexadecimal	Decimal
0	Unspecified	00000000000000000000000000000000	0x00000000	0
1	Head front	00000000000000000000000000000001	0x00000001	1
2	Head back	00000000000000000000000000000010	0x00000002	2
3	Head right	00000000000000000000000000000100	0x00000004	4
4	Head left	00000000000000000000000000001000	0x00000008	8
5	Right upper chest	00000000000000000000000000010000	0x00000010	16
6	Left upper chest	00000000000000000000000000100000	0x00000020	32
7	Abdomen	000000000000000000000000001000000	0x00000040	64
8	Waist	000000000000000000000000010000000	0x00000080	128
9	Upper back	0000000000000000000000000100000000	0x00000100	256
10	Lower back	00000000000000000000000001000000000	0x00000200	512
11	Right upper arm	000000000000000000000000010000000000	0x00000400	1 024
12	Left upper arm	0000000000000000000000000100000000000	0x00000800	2 048
13	Right forearm	00000000000000000000000001000000000000	0x00001000	4 096
14	Left forearm	000000000000000000000000010000000000000	0x00002000	8 192
15	Right wrist	0000000000000000000000000100000000000000	0x00004000	16 384
16	Left wrist	00000000000000000000000001000000000000000	0x00008000	32 768

Table 7 (continued)

	Name	Body_part_mask (binary)	Hexadecimal	Decimal
17	Right hand palm	00000000000000001000000000000000	0x00010000	65 536
18	Left hand palm	00000000000000001000000000000000	0x00020000	131 072
19	Right hand dorsum	00000000000000001000000000000000	0x00040000	262 144
20	Left hand dorsum	00000000000000001000000000000000	0x00080000	524 288
21	Right hand fingers	00000000000010000000000000000000	0x00100000	1 048 576
22	Left hand fingers	00000000000100000000000000000000	0x00200000	2 097 152
23	Right thigh	00000000010000000000000000000000	0x00400000	4 194 304
24	Left thigh	00000000100000000000000000000000	0x00800000	8 388 608
25	Right calf	00000001000000000000000000000000	0x01000000	16 777 216
26	Left calf	00000010000000000000000000000000	0x02000000	33 554 432
27	Right foot palm	00000100000000000000000000000000	0x04000000	67 108 864
28	Left foot palm	00001000000000000000000000000000	0x08000000	134 217 728
29	Right foot dorsum	00010000000000000000000000000000	0x10000000	268 435 456
30	Left foot dorsum	00100000000000000000000000000000	0x20000000	536 870 912
31	Right foot fingers	01000000000000000000000000000000	0x40000000	1 073 741 824
32	Left foot fingers	10000000000000000000000000000000	0x80000000	2 147 483 648

Table 8 — Examples of body part combinations

Name	Body_part_mask (binary)	Hexadecimal	Decimal
Right arm	00000000000101010101010000000000	0x00015540	87 360
Left arm	00000000000101010101010000000000	0x002AA800	2 795 520
Right leg	01010101010000000000000000000000	0x55400000	1 430 257 664
Left leg	10101010100000000000000000000000	0xAA800000	2 860 515 328
Upper body	00000000001111111111111111111111	0x003FFFFFFF	4 194 303
Lower body	11111111110000000000000000000000	0xFFC00000	4 290 772 992
Full body	11111111111111111111111111111111	0xFFFFFFFF	4 294 967 295

5.6.4 Semantic body part and actuator mapping

Alternatively, the haptic signal spatialization on the body representation can be done by a combination of body part targeting and actuator mapping on the body. This approach uses the body part target property, which is a set of successive values configured to represent a human body part or a group of body parts. The range of values is defined in [Table 9](#):

- A Group Node is a shortcut command to target a group of body parts.
- A Locational value is an optional reference to some body part node to add precision.
- A Body part Node is a human body part which is defined hierarchically (see [Figure 7](#)).

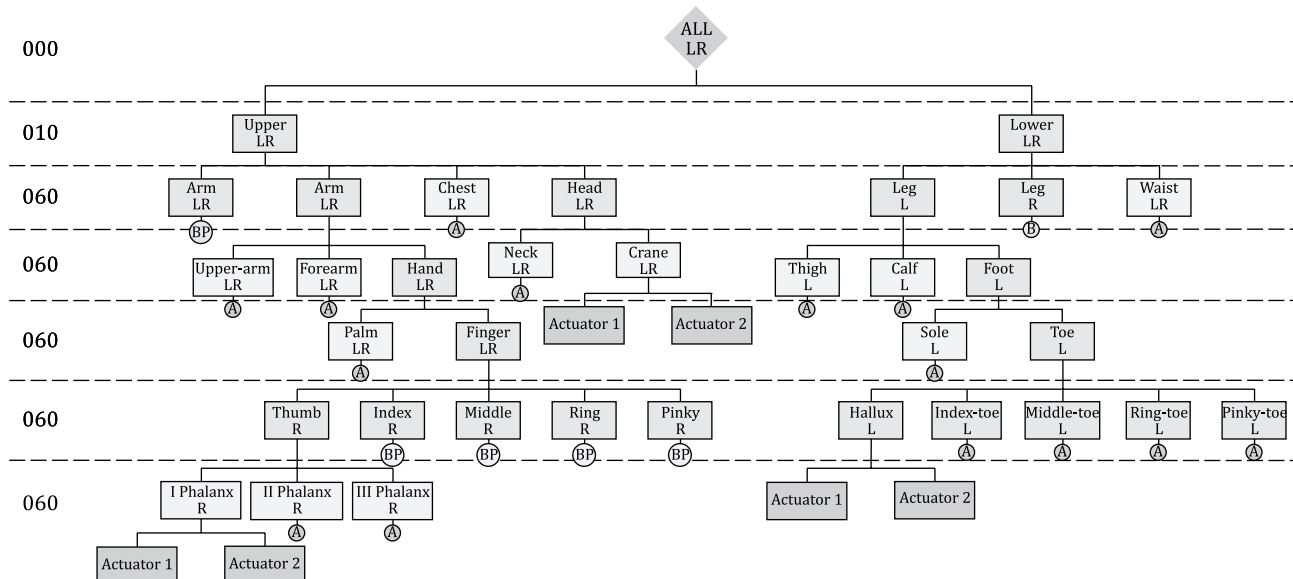


Figure 7 — Tree representation of body part target values

Table 9 — Body part target values

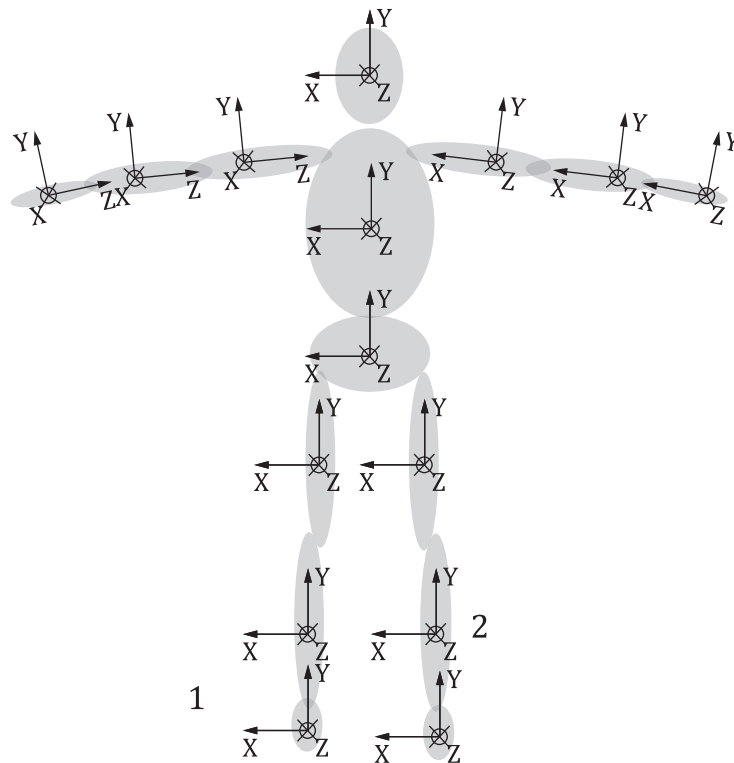
Name	Type	Value
Unknown	Group Node	0
All	Group Node	1
Upper	Group Node / Locational	10
Lower	Group Node / Locational	11
Right	Group Node / Locational	12
Left	Group Node / Locational	13
Front	Group Node / Locational	14
Back	Group Node / Locational	15
Arm	Body part Node	20
Head	Body part Node	21
Chest	Body part Node	22
Waist	Body part Node	23
Leg	Body part Node	24
Upper-arm	Body part Node	30
Forearm	Body part Node	31
Hand	Body part Node	32
Crane	Body part Node	33
Neck	Body part Node	34
Thigh	Body part Node	35
Calf	Body part Node	36
Foot	Body part Node	37
Palm	Body part Node	40
Finger	Body part Node	41
Sole	Body part Node	42
Toe	Body part Node	43
Thumb	Body part Node	50

Table 9 (continued)

Name	Type	Value
Index	Body part Node	51
Middle	Body part Node	52
Ring	Body part Node	53
Pinky	Body part Node	54
Hallux	Body part Node	55
Index-Toe	Body part Node	56
Middle-Toe	Body part Node	57
Ring-Toe	Body part Node	58
Pinky-Toe	Body part Node	59
First Phalanx	Body part Node	60
Second Phalanx	Body part Node	61
Third Phalanx	Body part Node	62
Minus	Operator	254
Plus	Operator	255

Each body part is then divided into a 3D right-handed actuator map which fulfils the following requirements:

- The X axis points to the right.
- The Y axis points up.
- The Z axis points forward.
- The up direction on each body part is based on a standing human looking at his or her hand outstretched in front of himself or herself (see [Figure 8](#) for a visual representation).
- The right direction on each body part is based on a standing human looking at his or her hand outstretched in front of himself or herself (see [Figure 8](#) for a visual representation).

**Key**

- 1 Sole of the root downwards. Same as hand
- 2 Body part model Convention: Right-handed (x, y, z)
- ⊙ towards viewer
- ⊗ away from viewer

Figure 8 — Representation of each body part basis for the actuator map

Each actuator can finally be targeted individually on the targeted body part to improve the precision of a haptic signal spatialization on the human body.

The actuators are mapped on the body part following a 3D mapping based on a vector $[X, Y, Z]$. Each value of the vector indicates the resolution in the respective 3D direction, according to the referential described in [Figure 8](#).

For instance, for only one actuator the mapping is $[1,1,1]$. A resolution of $[2,1,1]$ indicates two actuators on the X axis of the body part. A resolution of $[2,2,1]$ indicates four actuators on the X-Y plane. A resolution of $[2,2,2]$ indicates eight actuators arranged with two surfaces. For haptics devices which do not respect a cubic volumetric arrangement, some of the actuators may be virtual.

More details on the use of this representation are provided in [Annex C](#).

The detailed list of properties defining a mapping vector is given in [Table 10](#).

Table 10 — Mapping vector properties

Property	Description
X	Unit right vector. The range of possible values stored in the X axis is $[-127,127]$.
Y	Unit up vector. The range of possible values stored in the Y axis is $[-127,127]$.
Z	Unit forward vector. The range of possible values stored in the Z axis is $[-127,127]$.

5.7 Haptic band

A haptic band describes the haptic signal of a channel in a given frequency range. Bands are defined by a type and a sequential list of haptic effects each containing a set of keyframes. [Table 11](#) details the list of properties of a haptic band.

Table 11 — Haptic band properties

Property	Description
band type	Type of data contained in the band. There are four types of haptic bands: curve bands, transient bands, vectorial wave bands and wavelet wave bands. For each type of band, the information it contains has a different meaning: — Curve bands represent haptic signals with curves, described by a set of control points and a type of interpolation in-between. — Transient bands represent short momentary haptic effects, described with amplitude and frequency parameters. — Vectorial wave bands represent parametric haptic effects; described by a vector of parameters including temporal or spatial position, amplitude and frequency. The model allows both amplitude and frequency modulation of the signal. — Wavelet wave bands represent haptic effects encoded with wavelet transform decomposition, quantization, binary tree structure, and entropy coding. Subclause 5.9 details precisely how the data contained in keyframes is interpreted depending on the type of bands.
priority	Importance of the band for scalability. A lower value indicates higher priority. Given a limited bandwidth, decoders may ignore perception with a higher priority value.
Curve type	Present only for curve bands. This specifies the interpolation method to be used to synthesize the haptic signal of the band. Possible values are: — Linear — Cubic — Akima — Bézier — B-spline — Unknown (for application specific functions)
block length	This property is only present for wavelet wave bands. It is the duration of a wavelet block.
Lower frequency limit	Lower frequency limit of the band (Hz).
Upper frequency limit	Upper frequency limit of the band (Hz).
Effects	List of haptic effects as defined in 5.8 . If the effect list is empty, the band does not contain haptic data.

[Figure 9](#) illustrates two types of haptic bands: a curve band and a transient band). Points in the figure represent keyframes.

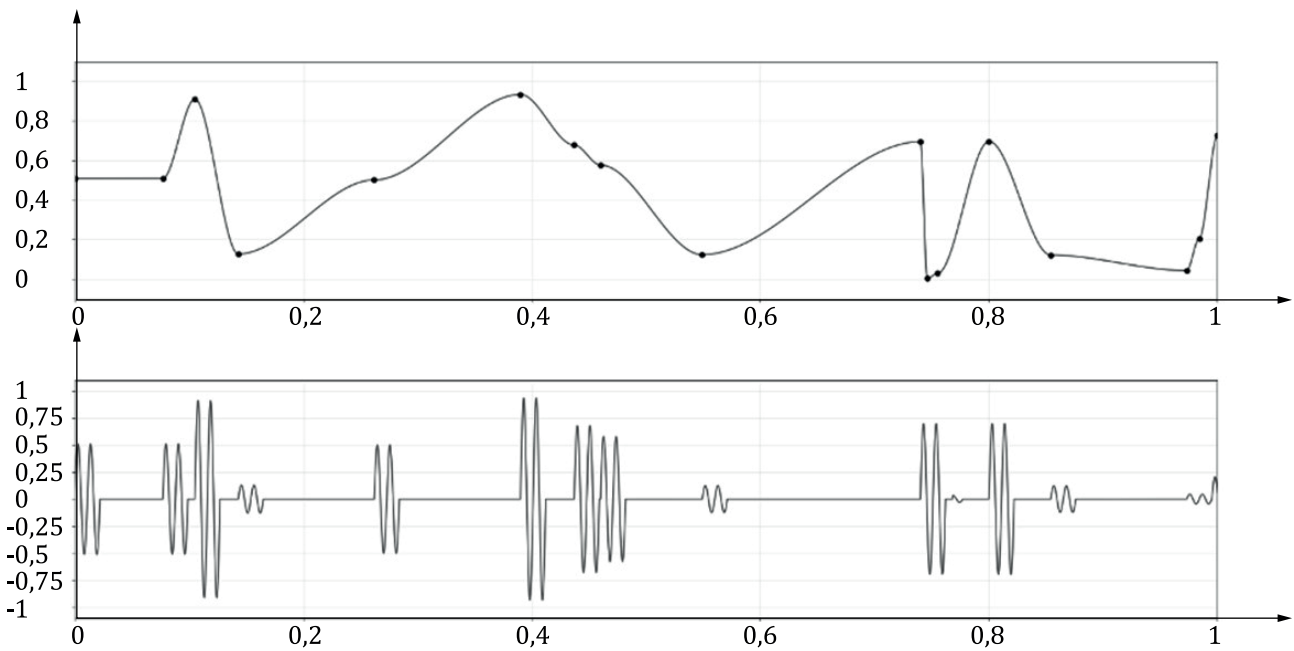


Figure 9 — Representation of a curve band (top) and a transient band (bottom)

For curve bands, different types of curve representations and interpolations can be used. The format currently supports five types of curves: linear, cubic, Akima, Bézier and B-spline.

— Linear

Each keyframe is a point in the curve and intermediate values are computed using linear interpolation.

— Cubic

Each keyframe is a point in the curve and intermediate values are computed using piecewise cubic polynomials.

— Akima

Each keyframe is a point in the curve, and intermediate values are computed using Akima interpolation. The Akima interpolation is a continuously differentiable sub-spline interpolation. It is built from piecewise third order polynomials. Only data from the next neighbour points are used to determine the coefficients of the interpolation polynomial.

For a set of data points:

$$s_i = s(x_i), 1 \leq i \leq k$$

where the interpolation function is defined as:

$$s(x) = a_0 + a_1 \cdot (x - x_i) + a_2 \cdot (x - x_i)^2 + a_3 \cdot (x - x_i)^3, x_i \leq x \leq x_{i+1}$$

To determine the coefficients a_0 , a_1 , a_2 and a_3 of the polynomial interpolation for each interval $[x_i, x_{i+1}]$, the function values s_i and s_{i+1} , and the first derivatives s'_i and s'_{i+1} at the end points of the interval are used.

— Bézier

Bézier curves are parametric curves defined by a set of control points. Bézier curves are generally used to model smooth curves that can be scaled indefinitely. It is defined by a set of control points P_0

through P_n , where n is called the order of the curve ($n = 1$ for linear, 2 for quadratic, 3 for cubic, etc.). The interpolation function of a quadratic Bézier curve is given by:

$$B(t) = (1-t)^2 P_0 + 2(1-t)tP_1 + t^2 P_2, \quad 0 \leq t \leq 1$$

Where $B(t)$ is the interpolation for a parameter t (between 0 and 1), and (P_0, P_1, P_2) are respectively the three control points used for the interpolation (see [Figure 10](#)).

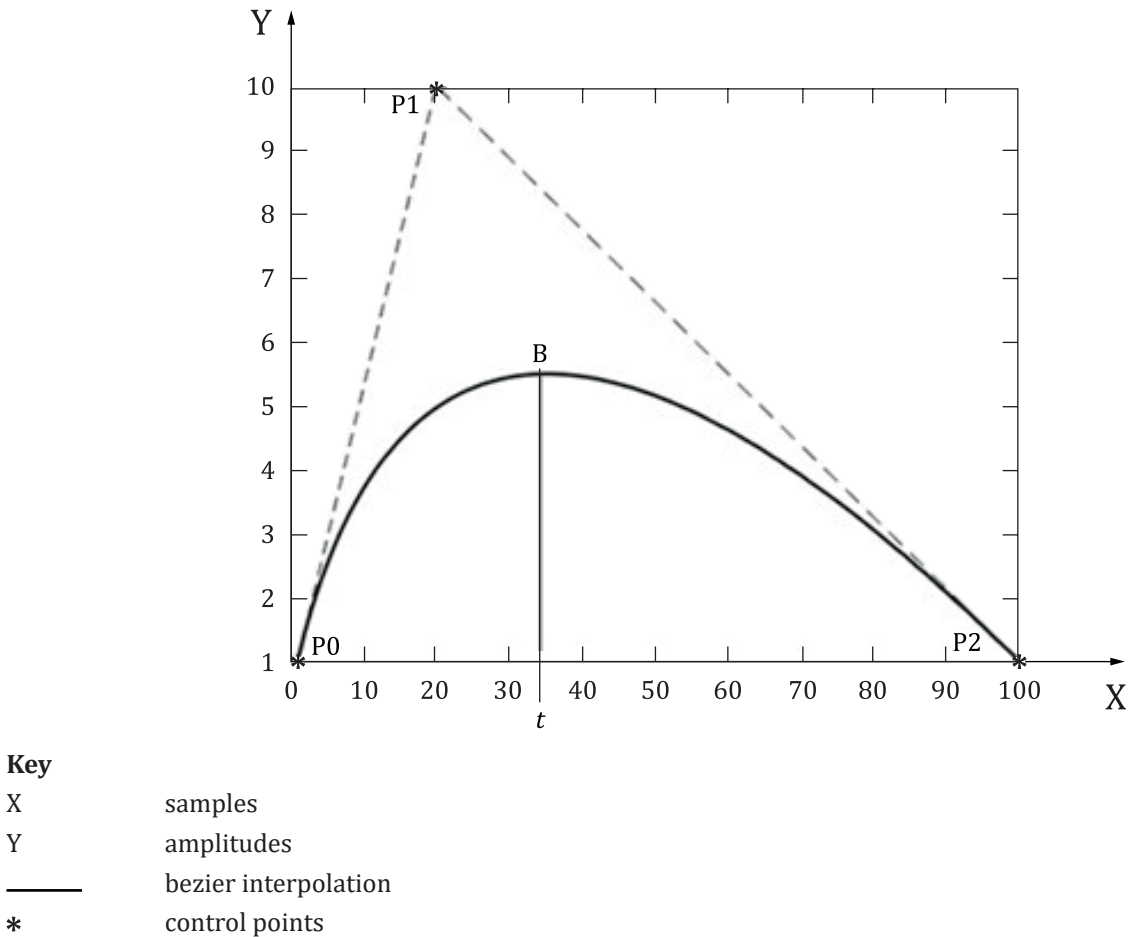


Figure 10 — Bézier curve example, with three control points

Here, the data in curve bands represent piecewise second order Bézier curves: the data are composed of consecutive quadratic Bézier curves, each defined by a set of three control points. The last control point of a Bézier curve is the first control point of the next one. This implies that the data contains at least three control points and an uneven number of control points. The control points contain 2D data defining their amplitude and timestamp. To ensure that the haptic data can be rendered properly, the control points are ordered in time: the timestamp of each control point is higher than the timestamp of the previous point (if it exists) and lower than the timestamp of the next control point (if it exists).

[Figure 11](#) illustrates the curve produced with a set of five control points. The signal is computed using two Bézier curves with P_0, P_1 and P_2 as control points for the first one and P_2, P_3 and P_4 for the second one.

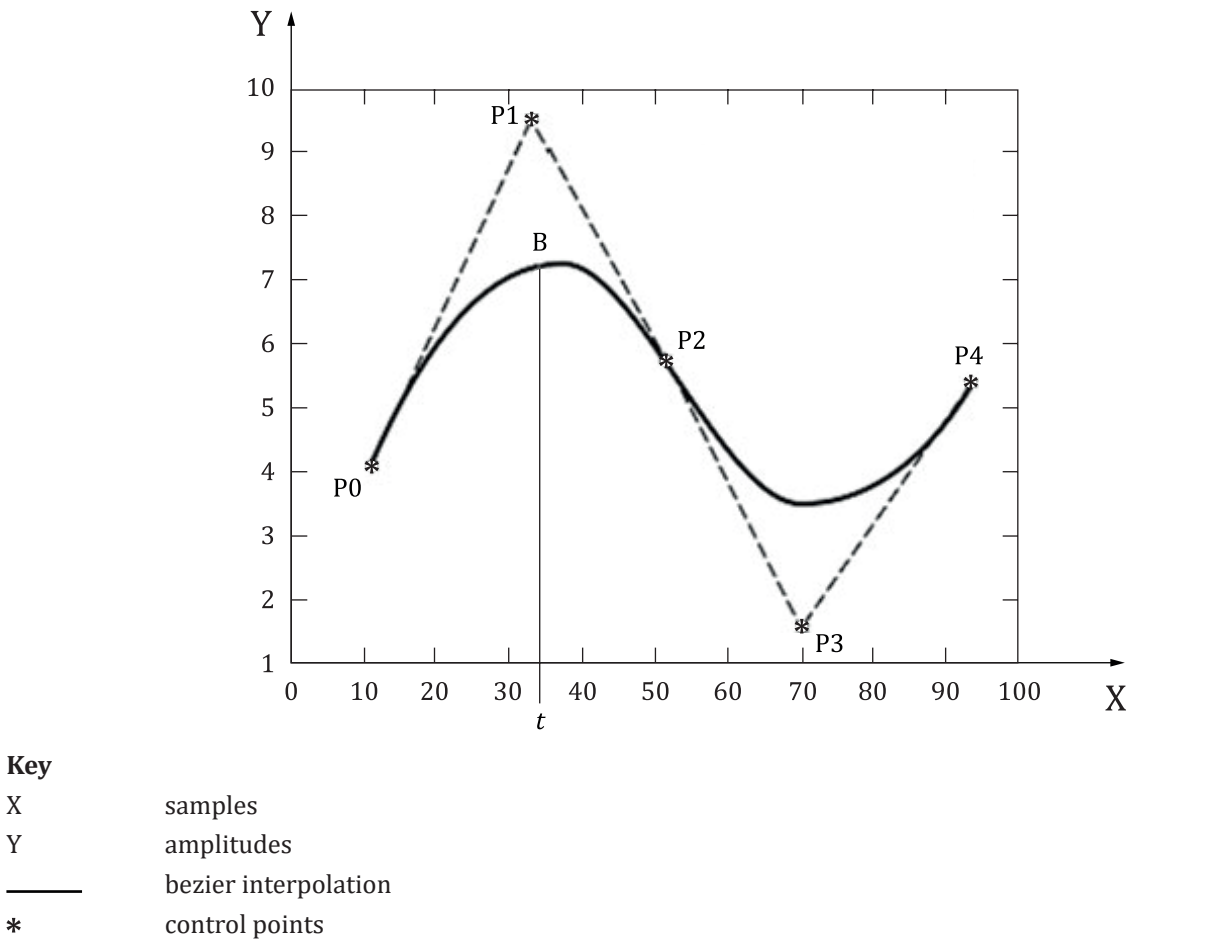


Figure 11 — Example of piecewise Bézier curve defined by five control points

— B-spline

A B-spline function is a combination of flexible segments controlled by a number of control points (P_i) creating smooth curves (see [Figure 12](#)). It is a generalization of Bézier curves, with higher interpolation accuracy. The shape of the curve only depends on the position of the control points. Modifying one control point will only change the curve locally. Here the data in the curve bands represent the control points of the B-spline.

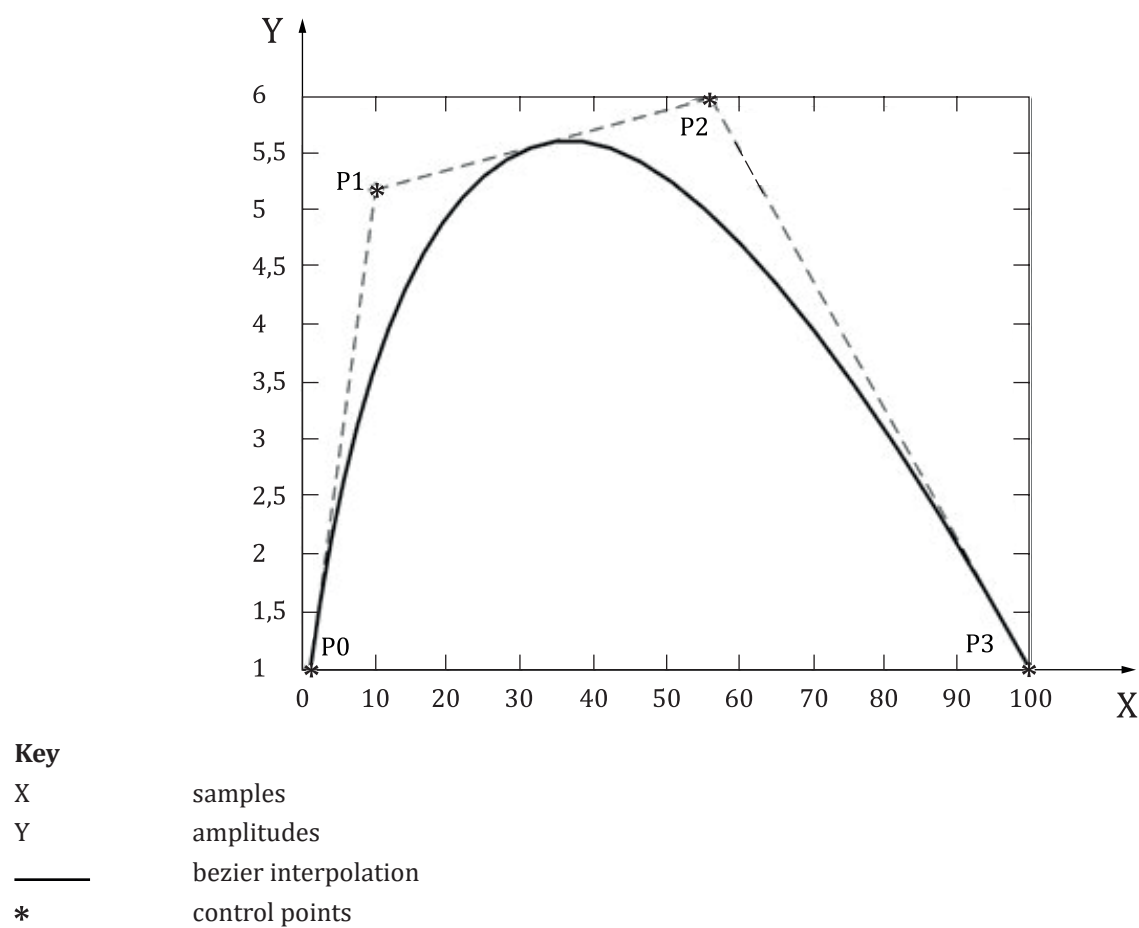


Figure 12 — Example of B-spline defined with four control points

5.8 Haptic effect

For every type of haptic band, haptic effects are defined with at least a position and a type. Depending on the type of band and the type of effect, additional properties is specified, including the phase, the base signal, a composition and a number of consecutive haptic keyframes describing the effect.

The detailed list of properties of a haptic effect is given in [Table 12](#).

Table 12 — Haptic effect properties

Property	Description
Id	Unique ID of the effect. This property is required for effects defined at the perception level in the library and for reference effects. For reference effects, this property corresponds to the ID of the library effect being referenced. The ID is unique among all effect IDs in this haptic experience.
Effect type	Type of haptic effect. Possible values are: — Basis: Effect containing signal data defined through a set of keyframes. — Composite: Effect composed of a set of other effects defined in the composition property. This type of effect does not directly contain keyframes. — Reference: Effect referencing an effect in the library. This type of effect is used at the band level to reference effects defined at the perception level in the effect library using a unique ID.
Semantic keywords	Semantic keywords of the haptic effect. It is an optional field that may be added to an effect.
Position	Indicates the temporal or spatial position of the effect. The value 0 corresponds to the starting position of the experience. The default unit for temporal haptic feedback will be milliseconds while it will be millimetres for spatial haptic feedback. Temporal or spatial is defined by the perception-Modality.
Phase	Indicates the phase of the effect in radian in the range $[0, 2\pi]$. This value is used only for vectorial wave bands and is ignored in effects stored inside a band of a different type.
Base signal	This property is only used for vectorial wave bands. It defines the type of waveform signal to be used by the synthesizer. Possible values are: — Sine — Square — Triangle — Sawtooth up — Sawtooth down
composition	This property can only be used with composite effects. It contains a sub-list of effects.
keyframes	List of keyframes as defined in 5.9 . The keyframes list is only used for basis effects.
Wavelet stream	Encoded wavelet stream.

The duration of a haptic effect is given, depending on the band type, as:

- Transient band: the position of the latest keyframe plus the duration of a transient which can differ among different implementations.
- Curve band: the position of the latest keyframe
- Vectorial wave band: the position of the latest keyframe
- Wavelet wave band: For this type of band, each effect corresponds to a single block. The duration of an effect then corresponds to the block length property specified at the band level.

Semantic keyword is a two-layers hierarchical metadata structure illustrated in [Table 13](#). It may be added as a supplementary information to an effect. At most only one layer 1 and one layer 2 semantic keywords shall be included with one effect. These keywords indicate the desired designer intention. When a presentation engine is incapable of rendering the exact specified effect it may decide to render a similar effect or not render at all. One example is a gunshot designed for a VR controller with a specific frequency range. A VR gun prop may be developed with its own haptics library.

Table 13 — Two-layer semantic keywords

Layer 1	Layer 2
UX	Undefined (default)
	Click
	Double click
	Success
	Error
	Alarm
	Confirmation
	Wrong
	Ring
	Message
Avatar	Undefined (default)
	Jumping
	Fall
	Crawl
	Swim
	Collision
	Grab
	Touch
	Swip
	Footstep
Special effect	Undefined (default)
	Washout
	Noise
Weapons & Combat	Undefined (default)
	Blade
	Hit
	Hand-thrown
	Elastic propulsion
	Pneumatic
	Handguns
	Rifles
	Shotgun
	Gun
	Machinegun
	Taser
	Electric shock
	Mines
	Missile
	Grenade
	Blast
	Undefined (default)
	Wind low
	Heat
	Cold

Table 13 (continued)

Layer 1	Layer 2
Ambient	Rain
	Waterfall
	Water drop
	Electric buzz
	Ignition
	Cracks
	Earthquake
	Sparks
	Thunderbolt
Texture	Undefined (default)
	Rock
	Gravel
	Sand
	Wood
	Metal
	Plastic
Vehicles	Undefined (default)
	Engine
	Doors
	Brake
	Mechanical Contraption
	Drift
	Road friction
	Brake
	Road bump
	Tires
	Air friction
Music	Undefined (default)
	Hard material
	Bouncy material
	Pucking
	Bowing
	Dtriking
	Brass instruments
	Woodwind instruments

5.9 Haptic keyframe

A haptic keyframe encodes a point or sample at a given position in time or space.

Table 14 — Haptic keyframe properties

Property	Description
amplitude modulation	Amplitude of the keyframe. For vibrotactile signals, a negative vibration refers to an opposite polarity of the modulated signal.
Frequency modulation	Relative frequency of the keyframe.
Relative position	Relative position of the keyframe.

The type of band indicates how a keyframe should be interpreted. Depending on the type of band, a keyframe stores one or several of the properties defined in [Table 14](#). The different interpretation of these values based on band type is detailed in [Table 15](#).

Table 15 — Keyframe interpretation for each band type

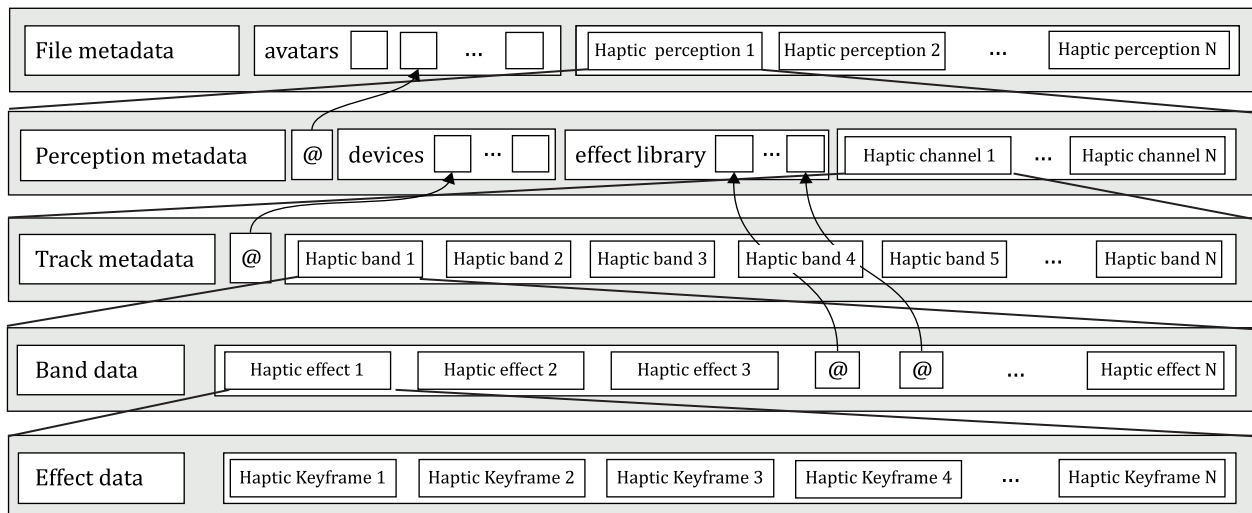
Band type	Keyframe interpretation
transient	Keyframes define with a position, an amplitude and a frequency.
Curve	Keyframes define a position and an amplitude. The keyframes represent the control points of the curve. The type of curve is specified at the band level and is used to compute the signal as detailed in subclause 5.7 .
vectorial wave	Keyframes define a position, an optional amplitude modulation and an optional frequency modulation. A keyframe contains at least one of these two optional parameters. Any interpolation between keyframes can be used.

The combination of multiple of these primitive keyframes allows to describe a full haptic effect.

6 Interchange file format

6.1 Overview

The interchange format is a JSON implementation of the data model. It is not memory-optimized, but it is human-readable and can be manually edited. The data structure of this format is illustrated in [Figure 13](#).

**Figure 13 — Detailed data structure of the interchange format**

As a JSON based format, it shall be compliant with ISO/IEC 21778 and RFC 8259. Each component of the data structure is defined as a JSON object and described below using the JSON Schema specification mechanism. The interchange format is described through the following list of JSON schemas. All the Schemas are provided in the normative [Annex A](#)

Table 16 — JSON schemas describing the interchange format

Name	Brief Description	Subclause
MPEG_haptics	Describes the haptic experience. This refers to subclause 5.2 of the data model.	6.2.1
MPEG_haptics_avatar	Describes a haptic avatar. This refers to subclause 5.3 of the data model. A haptic experience may include several avatars.	6.2.2
MPEG_haptics_perception	Describes a haptic signal associated with a specific perception modality (vibration, force, temperature, ...). This refers to subclause 5.4 of the data model. A haptic experience may include several modalities.	6.2.3
MPEG_haptics_reference_device	Provides a list of characteristics associated with a targeted haptic device. This refers to subclause 5.5 of the data model. A haptic experience may include several reference devices.	6.2.4
MPEG_haptics_channel	Describes a channel within a perception. Each perception may contain multiple channels. This refers to subclause 5.6 of the data model.	6.2.6
MPEG_haptics_vector	Describes the local direction on which the signal should be defined. This information is relevant for spatial based haptic signals. This refers to subclause 5.6.3 of the data model.	6.2.7
MPEG_haptics_band	Describes a frequency band within a haptic channel. This refers to subclause 5.7 of the data model. A haptic channel may contain multiple bands.	6.2.8
MPEG_haptics_effect	Describes a haptic effect within a frequency band. This refers to subclause 5.8 of the data model. A haptic band may contain multiple effects.	6.2.9
MPEG_haptics_keyframe	Describes a keyframe within a haptic effect. Properties contained in a keyframe are interpreted differently depending on the type of haptic band and the specified encoding modality. This refers to subclause 5.9 of the data model.	6.2.10

The following subclause details the specifications of the interchange format. Each component of the data structure is defined as JSON object with a list of properties corresponding to its attributes. The following subclause give a detailed description of the JSON objects and associated schemas as specified in [Table 16](#).

The following subclauses use types defined in [Table 17](#) to describe properties.

Table 17 — Descriptions of the types used in the following subclauses

string	The string type is used for strings of text. It may contain Unicode characters.
Integer	The integer type is used for integral numbers
number	The number type is used for any numeric type, either integers or floating-point numbers.
Boolean	The boolean type matches only two special values: “true” and “false”.
Array	Arrays are used for ordered lists of elements.
Object	Objects are the mapping type in JSON. They map “keys” to “values”. In JSON, the “keys” are always strings. Each of these pairs is conventionally referred to as a “property”. The properties (key-value pairs) on an object are defined using the “properties” keyword.
Enum	The enum keyword is used to restrict a value to a fixed set of values.

6.2 HJIF Specifications

6.2.1 MPEG_haptics

The MPEG_haptics element is the highest level of the format and describes the global haptic experience as defined in [subclause 5.2](#). It includes some metadata such as the date of the file and the version of the format, provides a description of the haptic experience, specifies the shape associated with the haptic experience, lists the different avatars (i.e., body representation) and defines all the perceptions.

The MPEG_haptics metadata are necessary to configure the global experience and include all the data structures introduced above, starting with perceptions. [Table 18](#) details the list of properties of an MPEG_haptics object corresponding to the properties defined in [Table 1](#) of [subclause 5.2](#).

Table 18 — Description of the MPEG_haptics object

Property	Type	Default	Description	Required
version	string	N/A	Year of the edition and amendment of ISO/IEC 23090-31 that this file conforms to, in the following format: XXXX or XXXX-Y, where XXXX is the year of publication and Y is the amendment number, if any. For this document, the value shall be "2023".	Yes
profile	string	N/A	Name of the profile used to generate the encoded stream according to the profile@level definition in the normative Annex D .	Yes
level	integer	N/A	Number of the level used to generate the encoded stream according to the profile@level definition in the normative Annex D . The value shall be equal to or greater than zero.	Yes
date	string	N/A	Indicates the creation date of this haptic experience. The date format shall conform to ISO 8601-1 and ISO 8601-2.	Yes
description	string	N/A	The user-defined description of this haptic experience.	Yes
timescale	integer	1000	Number of ticks per second. The value shall be greater than zero.	No
avatars	array<MPEG_haptics.avatar>	N/A	Provides the List of MPEG_haptics.avatar as described in 6.2.2 .	Yes
perceptions	array<MPEG_haptics.perception>	N/A	Provides the List of MPEG_haptics.perception as described in 6.2.3 . The array shall contain at least one element.	Yes
syncs	array <MPEG_haptics.sync>	N/A	Provides the List of MPEG_haptics.sync as described in 6.2.4 .	No

6.2.2 MPEG_haptics.avatar

The MPEG_haptics.avatar element defines a haptic avatar as a body representation. It is particularly useful to specify on which part of the body the haptic effect should apply. This corresponds to [subclause 5.3](#) of the data model.

Different types of avatars may be used for different applications or different haptic properties. A custom 3D mesh can be specified for the representation of the avatars as described in [subclause 5.3](#). When using a custom 3D mesh representation, a reference URI to a custom mesh from a companion file shall be provided (.obj or .glTF file formats may be used).

[Table 19](#) details the list of properties of an MPEG_haptics.avatar object corresponding to the properties defined in [Table 2](#) of [subclause 5.3](#).

Table 19 — Description of the MPEG_haptics.avatar object

Property	Type	Default	Description	Required
id	integer	N/A	Unique ID of the avatar. The value shall be greater than zero. The 0 value is reserved for unspecified avatars.	Yes
lod	integer	N/A	If the avatar uses a mesh with several levels of detail (LODs), this indicates which LOD to use for the avatar. The value shall be equal to or greater than zero.	Yes
type	enum<string>	N/A	Indicates the type of haptic perception represented by the avatar. Possible values are: — “Vibration” — “Pressure” — “Temperature” — “Custom”	Yes
mesh	string	N/A	URI path to the custom mesh file. The URI shall conform to the syntax defined in RFC 3986.	No

6.2.3 MPEG_haptics.perception

The MPEG_haptics.perception element specifies a haptic perception with an ID, a description of the perception, some metadata information (*e.g.* perception modality and type of encoding), a reference to an avatar, a list of reference devices and the list of channels. [Table 20](#) details the list of properties of an MPEG_haptics.perception object corresponding to the properties defined in [Table 3](#) of [subclause 5.4](#).