



**5G;
Evaluation and Characterization
of Beyond 2D Video Formats and Codecs
(3GPP TR 26.956 version 19.0.0 Release 19)**



Reference

DTR/TSGS-0426956vj00

Keywords

5G

ETSI

650 Route des Lucioles
F-06921 Sophia Antipolis Cedex - FRANCE

Tel.: +33 4 92 94 42 00 Fax: +33 4 93 65 47 16

Siret N° 348 623 562 00017 - APE 7112B
Association à but non lucratif enregistrée à la
Sous-Préfecture de Grasse (06) N° w061004871

Important notice

The present document can be downloaded from the
[ETSI Search & Browse Standards application](#).

The present document may be made available in electronic versions and/or in print. The content of any electronic and/or print versions of the present document shall not be modified without the prior written authorization of ETSI. In case of any existing or perceived difference in contents between such versions and/or in print, the prevailing version of an ETSI deliverable is the one made publicly available in PDF format on [ETSI deliver repository](#).

Users should be aware that the present document may be revised or have its status changed,
this information is available in the [Milestones listing](#).

If you find errors in the present document, please send your comments to
the relevant service listed under [Committee Support Staff](#).

If you find a security vulnerability in the present document, please report it through our
[Coordinated Vulnerability Disclosure \(CVD\)](#) program.

Notice of disclaimer & limitation of liability

The information provided in the present deliverable is directed solely to professionals who have the appropriate degree of experience to understand and interpret its content in accordance with generally accepted engineering or other professional standard and applicable regulations.

No recommendation as to products and services or vendors is made or should be implied.

No representation or warranty is made that this deliverable is technically accurate or sufficient or conforms to any law and/or governmental rule and/or regulation and further, no representation or warranty is made of merchantability or fitness for any particular purpose or against infringement of intellectual property rights.

In no event shall ETSI be held liable for loss of profits or any other incidental or consequential damages.

Any software contained in this deliverable is provided "AS IS" with no warranties, express or implied, including but not limited to, the warranties of merchantability, fitness for a particular purpose and non-infringement of intellectual property rights and ETSI shall not be held liable in any event for any damages whatsoever (including, without limitation, damages for loss of profits, business interruption, loss of information, or any other pecuniary loss) arising out of or related to the use of or inability to use the software.

Copyright Notification

No part may be reproduced or utilized in any form or by any means, electronic or mechanical, including photocopying and microfilm except as authorized by written permission of ETSI.

The content of the PDF version shall not be modified without the written authorization of ETSI.

The copyright and the foregoing restriction extend to reproduction in all media.

© ETSI 2025.
All rights reserved.

Intellectual Property Rights

Essential patents

IPRs essential or potentially essential to normative deliverables may have been declared to ETSI. The declarations pertaining to these essential IPRs, if any, are publicly available for **ETSI members and non-members**, and can be found in ETSI SR 000 314: *"Intellectual Property Rights (IPRs); Essential, or potentially Essential, IPRs notified to ETSI in respect of ETSI standards"*, which is available from the ETSI Secretariat. Latest updates are available on the [ETSI IPR online database](#).

Pursuant to the ETSI Directives including the ETSI IPR Policy, no investigation regarding the essentiality of IPRs, including IPR searches, has been carried out by ETSI. No guarantee can be given as to the existence of other IPRs not referenced in ETSI SR 000 314 (or the updates on the ETSI Web server) which are, or may be, or may become, essential to the present document.

Trademarks

The present document may include trademarks and/or tradenames which are asserted and/or registered by their owners. ETSI claims no ownership of these except for any which are indicated as being the property of ETSI, and conveys no right to use or reproduce any trademark and/or tradename. Mention of those trademarks in the present document does not constitute an endorsement by ETSI of products, services or organizations associated with those trademarks.

DECT™, **PLUGTESTS™**, **UMTS™** and the ETSI logo are trademarks of ETSI registered for the benefit of its Members. **3GPP™**, **LTE™** and **5G™** logo are trademarks of ETSI registered for the benefit of its Members and of the 3GPP Organizational Partners. **oneM2M™** logo is a trademark of ETSI registered for the benefit of its Members and of the oneM2M Partners. **GSM®** and the GSM logo are trademarks registered and owned by the GSM Association.

Legal Notice

This Technical Report (TR) has been produced by ETSI 3rd Generation Partnership Project (3GPP).

The present document may refer to technical specifications or reports using their 3GPP identities. These shall be interpreted as being references to the corresponding ETSI deliverables.

The cross reference between 3GPP and ETSI identities can be found at [3GPP to ETSI numbering cross-referencing](#).

Modal verbs terminology

In the present document "**should**", "**should not**", "**may**", "**need not**", "**will**", "**will not**", "**can**" and "**cannot**" are to be interpreted as described in clause 3.2 of the [ETSI Drafting Rules](#) (Verbal forms for the expression of provisions).

"**must**" and "**must not**" are **NOT** allowed in ETSI deliverables except when used in direct citation.

Contents

Intellectual Property Rights	2
Legal Notice	2
Modal verbs terminology.....	2
Foreword.....	11
Introduction	12
1 Scope	13
2 References	13
3 Definitions of terms, symbols and abbreviations	21
3.1 Terms.....	21
3.2 Symbols.....	21
3.3 Abbreviations	22
4 Beyond 2D Video Formats.....	23
4.1 Introduction	23
4.2 Reference Model for Beyond 2D Video.....	24
4.2.1 Overview	24
4.3 Beyond 2D Video Representation Formats	26
4.3.1 Introduction.....	26
4.3.2 Extensions to Stereoscopic Video Representation Formats	26
4.3.2.1 Definition	26
4.3.2.2 Stereoscopic Video format description according to TS 26.265	26
4.3.2.3 Extensions to Stereoscopic Video Representation formats	28
4.3.2.4 Production and Capturing Systems	30
4.3.2.5 Rendering and Display Systems.....	31
4.3.2.6 Supporting Information.....	31
4.3.2.7 Benefits and Limitations	32
4.3.2.7.1 Benefits.....	32
4.3.2.7.2 Limitations.....	32
4.3.3 Dense Dynamic Point Cloud representation format.....	32
4.3.3.1 Definition	32
4.3.3.2 Production and Capturing Systems	33
4.3.3.3 Rendering and Display Systems.....	33
4.3.3.4 Support Information	35
4.3.3.4.1 Test and reference sequences	35
4.3.3.4.2 Uncompressed data size.....	35
4.3.3.4.3 Known compression technology.....	35
4.3.3.4.4 Conversion from other formats.....	35
4.3.3.4.5 Typical quality criteria	35
4.3.3.5 Benefits and Limitations	36
4.3.3.5.1 Benefits.....	36
4.3.3.5.2 Limitations.....	36
4.3.4 Multi-view video Representation Format	36
4.3.4.1 Definition	36
4.3.4.2 Production and Capturing Systems	38
4.3.4.3 Rendering and Display Systems.....	39
4.3.4.4 Supporting Information	40
4.3.4.4.1 Camera placement	40
4.3.4.4.2 Spatial resolution.....	40
4.3.4.4.2 Objective metrics.....	41
4.3.4.4.3 Coding and delivery options	41
4.3.4.5 Benefits and Limitations	41
4.3.4.5.1 Benefits.....	41
4.3.4.5.2 Limitations.....	41
4.3.5 Dynamic Mesh Representation Format.....	41

4.3.5.1	Definition	41
4.3.5.2	Production and Capturing Systems	42
4.3.5.3	Rendering and Display Systems	43
4.3.5.4	Supporting Information	43
4.3.5.4.1	Test and reference sequences	43
4.3.5.4.2	Uncompressed data size	43
4.3.5.4.3	Known compression technologies	44
4.3.5.4.4	Conversion from other formats	44
4.3.5.4.5	Typical Quality Criteria	44
4.3.5.4.5.1	Objective Metrics	44
4.3.5.4.5.2	Subjective Evaluation	46
4.3.2.7	Benefits and Limitations	46
4.3.2.7.1	Benefits	46
4.3.2.7.2	Limitations	46
4.3.6	Formats under Research	46
4.3.6.1	Neural Radiance Fields	46
4.3.6.1.1	Introduction	46
4.3.6.1.2	Definition	47
4.3.6.1.3	Production and Capturing Systems	47
4.3.6.1.4	Rendering and Display Systems	48
4.3.6.1.5	Supporting Information	48
4.3.6.1.6	Benefits and Limitations	49
4.3.6.1.6.1	Benefits	49
4.3.6.1.6.2	Limitations	49
4.3.6.2	Light Fields Video	49
4.3.6.2.1	Definition	49
4.3.6.2.2	Production and Capturing Systems	51
4.3.6.2.3	Rendering and Display Systems	52
4.3.6.2.4	Supporting Information	52
4.3.6.2.5	Benefits and Limitations	53
4.3.6.2.5.1	Benefits	53
4.3.6.2.5.2	Limitations	53
4.3.6.3	3D Gaussian Splatting	54
4.3.6.3.1	Introduction	54
4.3.6.3.2	Overview	54
4.3.6.3.3	Production and Capturing Systems	56
4.3.6.3.4	Rendering and Display Systems	57
4.3.6.3.5	Supporting Information	57
4.3.6.3.7	Benefits and Limitations	59
4.3.6.3.7.1	Benefits	59
4.3.6.3.7.2	Limitations	59
4.4	AI-Generated Beyond 2D content	60
4.4.1	General	60
4.4.2	AI-Generated Dynamic Mesh	61
4.4.2.1	General	61
4.4.2.2	Image-Generated Dynamic Mesh	61
4.4.2.3	Text-Generated Dynamic Mesh	62
5	Overview of existing "Beyond 2D" Video Capabilities in 3GPP	62
5.1	Introduction	62
5.2	AR Video Capabilities	63
5.3	VR Video Profiles	63
5.4	Messaging Services	64
6	Evaluation and Characterization Framework	65
6.1	Overview	65
6.2	Reference Sequences	65
6.3	Reference Software Tools	65
6.4	Metrics	65
6.5	Encoding Constraints	65
7	Considered Scenarios	66
7.1	Introduction	66

7.2	Scenario 1: UE-to-UE Stereoscopic Video Live Streaming	66
7.2.1	Motivation.....	66
7.2.2	Description of the Anticipated Application	67
7.2.2.1	Overall Description	67
7.2.2.2	Capturing and processing	67
7.2.2.3	Encoding	69
7.2.2.4	Packing and Delivery	69
7.2.2.5	Decoding	69
7.2.2.6	Rendering	69
7.2.3	Source Format Properties.....	69
7.2.4	Encoding and Decoding Constraints.....	69
7.2.5	Performance Metrics	70
7.2.5.1	Objective Metrics for Captured Stereoscopic Video	70
7.2.5.1.1	Quality of individual views	70
7.2.5.1.2	Quality of cyclopean view.....	71
7.2.5.1.3	Quality of depth maps	71
7.2.5.1.4	Weighting constants	71
7.2.5.2	Objective Metrics for Generated Stereoscopic Video	72
7.2.5.3	Subjective Evaluation.....	73
7.2.5.4	Correlation between the objective and subjective metrics	74
7.2.6	Interoperability Consideration	74
7.2.7	Reference Sequences	74
7.2.7.1	Candidate Source Stereoscopic 3D Video Sequences	74
7.2.7.1.1	Public Datasets	74
7.2.7.1.2	Self-Converted Sequences.....	75
7.2.7.1.3	Self-Captured Sequences	75
7.2.8	Test Condition	76
7.2.8.1	Test model and configuration files.....	76
7.2.8.2	Rate points and test conditions.....	76
7.2.8.3	Profiles	76
7.2.8.4	Bitstream Generation	76
7.2.9	External Performance data.....	77
7.2.10	Additional information	77
7.3	Scenario 2: Streaming of professionally produced Volumetric Video with single asset containing people.....	78
7.3.1	Scenario name.....	78
7.3.2	Motivation for the scenario.....	78
7.3.3	Description of the scenario	80
7.3.4	Source format properties.....	85
7.3.4.1	Conversion and quantization for dense dynamic point cloud format.....	85
7.3.4.2	Impact of rendering for dense dynamic point cloud format	86
7.3.4.3	Impact of the background.....	86
7.3.4.4	Visual quality examples sequences in dense dynamic point cloud format.....	86
7.3.4.5.1	Thomas near with representative renderer (splat blend mode) and neutral background.....	87
7.3.4.5.2	Thomas near with representative renderer (cube mode) and neutral background	90
7.3.5	Encoding and decoding constraints and settings.....	93
7.3.6	Performance Metrics and Requirements	93
7.3.6.1	Anchors	93
7.3.6.2	Objective tests	93
7.3.6.3	Subjective tests.....	94
7.3.6.4	Correlation between the objective and subjective metrics	95
7.3.6.5	Verification and crosscheck	95
7.3.7	Interoperability Considerations for the application.....	95
7.3.8	Test Sequences.....	95
7.3.8.1	Candidate source dense point cloud sequences.....	95
7.3.8.2	Selected source dense point cloud sequences.....	95
7.3.8.3	Metadata for source dense point cloud sequences.....	96
7.3.8.3.1	Overview	96
7.3.8.3.2	JSON Scheme.....	97
7.3.8.3.3	Example.....	97
7.3.9	Detailed test conditions.....	98
7.3.9.1	V-PCC test model and configuration files	98
7.3.9.2	Rate points and test conditions.....	98

7.3.9.3	Profiles	99
7.3.9.4	Bitstream Generation, output	99
7.3.9.5	Videos Generation for subjective tests	101
7.3.9.6	Verification / crosschecks	101
7.3.10	External Performance data	101
7.3.11	Additional information	101
7.4	Scenario 3: Streaming of Multi-view plus depth Produced Content	102
7.4.1	Motivation for the scenario	102
7.4.2	Description of the scenario	102
7.4.3	Source format properties	104
7.4.4	Encoding and decoding constraints and settings	104
7.4.5	Performance Metrics and Requirements	106
7.4.6	Interoperability Considerations for the application	107
7.4.7	Test Sequences	107
7.4.8	External Performance data	107
7.4.9	Additional Information	107
8	Common Evaluation Features	107
9	Evaluation of Selected Scenarios	108
9.1	Introduction	108
9.2	Scenario 1: UE-to-UE Stereoscopic Video Live Streaming	108
9.2.1	Evaluation Overview	108
9.2.2	Reference Sequences	108
9.2.3	Performance Metrics	108
9.2.4	Candidate Solutions	108
9.2.4.1	Solution 1: Simulcast HEVC	108
9.2.4.1.1	Introduction	108
9.2.4.1.2	Reference Software	108
9.2.4.1.3	Parameter Settings	109
9.2.4.1.4	Evaluation Results	109
9.2.4.1.5	Network Requirements	109
9.2.4.2	Solution 2: MV-HEVC	109
9.2.4.2.1	Introduction	109
9.2.4.2.2	Reference Software	110
9.2.4.2.3	Parameter Settings	110
9.2.4.2.4	Evaluation Results	110
9.2.4.2.5	Network Requirements	110
9.2.5 Summary of	
	Evaluation	110
9.3	Scenario 2: Streaming of professionally produced Volumetric Video with single asset containing people...	112
9.3.1	Evaluation Overview	112
9.3.2	Reference Sequences	113
9.3.3	Performance Metrics	113
9.3.4	Candidate Solutions	113
9.3.4.1	Solution 1: MPEG V-PCC profile HEVC Main10 V-PCC Basic Rec0	113
9.3.4.1.1	Introduction	113
9.3.4.1.2	Reference Software	113
9.3.4.1.3	Parameter Settings	113
9.3.4.1.4	Distribution	113
9.3.4.1.5	Evaluation Results	113
9.3.4.1.5.1	Objective evaluation	113
9.3.4.1.5.1.1	Objective results of sequence Mitch	114
9.3.4.1.5.1.2	Objective results of sequence JuggleSoccer	115
9.3.4.1.5.1.3	Objective results of sequence Henry	116
9.3.4.1.5.1.4	Objective results of sequence Nathalie	118
9.3.4.1.5.1.5	Objective results of sequence Aliyah	119
9.3.4.1.5.1.6	Bitstream crosschecks	120
9.3.4.1.5.2	Subjective evaluation	120
9.3.4.1.5.3	External evaluation	120
9.3.4.1.5.3.1	External reports	120
9.3.4.1.5.3.2	Evaluation platform	121

9.3.4.1.6	Network Requirements	121
9.4	Scenario 3: Streaming of Multi-view plus depth Produced Content	121
9.4.1	Evaluation Overview	121
9.4.2	Reference Sequences	121
9.4.3	Performance Metrics	121
9.4.4	Candidate Solutions	121
9.4.4.1	Solution 1: HEVC Main10 MIV Main	121
9.4.4.1.1	Introduction	121
9.4.4.1.2	Reference Software	121
9.4.4.1.3	Parameter Settings	122
9.4.4.1.4	Distribution	123
9.4.4.1.5	Evaluation Results	123
9.4.4.1.5.1	Example atlas frames	123
9.4.4.1.5.2	Pixel rate and MIV levels	124
9.4.4.1.5.3	Rate-distortion characteristics	125
9.4.4.1.5.4	Pose trace videos	126
9.4.4.1.5.5	Availability of test data	126
10	Gaps and Optimization Potential	127
10.1	Identified Gaps and Deficiencies with Video Capabilities	127
10.2	Potential Requirements for New Video Capabilities	128
10.3	Potential Network Optimizations	128
11	Conclusions and Proposed Next Steps	129
11.1	Summary and Conclusions	129
11.2	Recommendations	130
Annex A: Scenario Template		132
A.1	Introduction	132
A.2	Template	132
Annex B: Data Formats and Metrics		135
B.1	Introduction	135
B.2	Raw Video Sequences	135
B.2.1	Overview	135
B.2.2	JSON Schema	135
B.2.3	JSON Scheme for Dense Dynamic Point Cloud	137
Annex C: Reference Sequences		143
C.1	Introduction	143
C.2	Test Sequences for Volumetric Video with single asset containing people	143
C.2.1	Overview	143
C.2.2	Juggle Soccer test sequence	143
C.2.2.1	Description	143
C.2.2.2	Sequence properties	144
C.2.2.3	Copyright and license information	144
C.2.3	Mitch test sequence	145
C.2.3.1	Description	145
C.2.3.2	Sequence properties	145
C.2.3.3	Copyright and license information	146
C.2.4	Thomas test sequence	146
C.2.4.1	Description	146
C.2.4.2	Sequence properties	146
C.2.4.3	Copyright and license information	146
C.2.5	Nathalie test sequence	147
C.2.5.1	Description	147
C.2.5.2	Sequence properties	147
C.2.5.3	Copyright and license information	148
C.2.6	Steam Roller test sequence	148

C.2.6.1	Description.....	148
C.2.6.2	Sequence properties	148
C.2.6.3	Copyright and license information.....	148
C.2.7	Aliyah test sequence.....	149
C.2.7.1	Description.....	149
C.2.7.2	Sequence properties	149
C.2.7.3	Copyright and license information.....	150
C.2.8	Henry test sequence.....	150
C.2.8.1	Description.....	150
C.2.8.2	Sequence properties	150
C.2.8.3	Copyright and license information.....	151
C.2.9	Ultra Video Group of Tampere University test sequences	151
C.2.9.1	Description.....	151
C.2.9.2	Sequence properties	151
C.2.9.3	Copyright and license information.....	152
C.2.10	Owlii Inc test sequences	152
C.2.10.1	Description.....	152
C.2.10.2	Sequence properties	152
C.2.10.3	Copyright and license information.....	153
C.2.11	Vologram Ltd test sequences.....	153
C.2.11.1	Description.....	153
C.2.11.2	Sequence properties	153
C.2.11.3	Copyright and license information.....	153
C.2.12	Exercise test sequences	154
C.2.12.1	Description.....	154
C.2.12.2	Sequence properties	154
C.2.12.3	Copyright and license information.....	155
C.3	Test Sequences for UE-to-UE Stereoscopic Video Live Streaming	155
C.3.1	Overview	155
C.3.2	Street View - captured test sequence	155
C.3.2.1	Description.....	155
C.3.2.2	Sequence properties	155
C.3.2.3	Copyright and license information.....	156
C.3.3	Cute Dog - Captured test sequence	156
C.3.3.1	Description.....	156
C.3.3.2	Sequence properties	156
C.3.3.3	Copyright and license information.....	157
C.3.4	Moving Girl - Captured test sequence	157
C.3.4.1	Description.....	157
C.3.4.2	Sequence properties	157
C.3.4.3	Copyright and license information.....	158
C.3.5	Street View - Generated test sequence	158
C.3.5.1	Description.....	158
C.3.5.2	Sequence properties	158
C.3.5.3	Copyright and license information.....	158
C.3.6	Cute Dog - Generated test sequence	159
C.3.6.1	Description.....	159
C.3.6.2	Sequence properties	159
C.3.6.3	Copyright and license information.....	159
C.3.7	Moving Girl - Generated test sequence	160
C.3.7.1	Description.....	160
C.3.7.2	Sequence properties	160
C.3.7.3	Copyright and license information.....	160
C.4	Test Sequences for Streaming of Multi-view plus depth Produced Content.....	160
C.4.1	Overview	160
C.4.2	Breakfast test sequence	160
C.4.2.1	Description.....	160
C.4.2.2	Sequence properties	161
C.4.2.3	Copyright and license information.....	161
C.4.3	Bartender test sequence	161

C.4.3.1	Description.....	161
C.4.3.2	Sequence properties	162
C.4.3.3	Copyright and license information.....	162
C.4.4	DanceMoves test sequence.....	162
C.4.4.1	Description.....	162
C.4.4.2	Sequence properties	163
C.4.4.3	Copyright and license information.....	163

Annex D: Software Package164

D.1	Introduction	164
D.2	Video Processing.....	164
D.2.1	Overview	164
D.2.2	Common Color Conversion.....	164
D.2.3	Video Composition.....	165
D.2.4	FFmpeg Tools	166
D.3	Scenario 2 Processing.....	166
D.3.1	Overview	166
D.3.2	Installation.....	166
D.3.2.1	Cloning	166
D.3.2.2	Working Directory	166
D.3.3	Test sequence preparation	167
D.3.3.1	Dense dynamic point cloud.....	167
D.3.3.1.1	Generation of target dense dynamic point clouds	167
D.3.4	Bitstream and objective metric generation	168
D.3.4.1	Dense dynamic point cloud.....	168
D.3.4.1.1	Executing tests	168
D.3.4.1.2	Objective results.....	169
D.3.5	Video generation	170
D.3.5.1	Dense dynamic point cloud.....	170

Annex E: Testing support material.....172

E.1	3D background models for scenario 2 tests.....	172
E.1.1	Introduction	172
E.2	Crouch End Station 3D background model.....	172
E.2.1	Description	172
E.2.2	Copyright and license information	172
E.3	Great Drawing Room 3D background model.....	173
E.3.1	Description	173
E.3.2	Copyright and license information	173
E.4	Southbank Undercroft Skatepark 3D background model.....	174
E.4.1	Description	174
E.4.2	Copyright and license information	174

Annex F: Data Management and Hosting175

F.1	Reference Sequences.....	175
F.1.1	Hosting	175
F.1.1.1	Scenario 2	175
F.1.1.2	Scenario 3	175
F.1.2	Uploading	176
F.1.3	Downloading	176
F.2	Anchors and Tests	176
F.2.1	Hosting	176
F.2.1.1	Scenario 2	177
F.2.1.2	Scenario 3	177
F.2.2	Uploading	178
F.2.3	Downloading	178

Annex G: Change history.....

179

History

180

Foreword

This Technical Report has been produced by the 3rd Generation Partnership Project (3GPP).

The contents of the present document are subject to continuing work within the TSG and may change following formal TSG approval. Should the TSG modify the contents of the present document, it will be re-released by the TSG with an identifying change of release date and an increase in version number as follows:

Version x.y.z

where:

- x the first digit:
 - 1 presented to TSG for information;
 - 2 presented to TSG for approval;
 - 3 or greater indicates TSG approved document under change control.
- y the second digit is incremented for all changes of substance, i.e. technical enhancements, corrections, updates, etc.
- z the third digit is incremented when editorial only changes have been incorporated in the document.

In the present document, modal verbs have the following meanings:

- shall** indicates a mandatory requirement to do something
- shall not** indicates an interdiction (prohibition) to do something

The constructions "shall" and "shall not" are confined to the context of normative provisions, and do not appear in Technical Reports.

The constructions "must" and "must not" are not used as substitutes for "shall" and "shall not". Their use is avoided insofar as possible, and they are not used in a normative context except in a direct citation from an external, referenced, non-3GPP document, or so as to maintain continuity of style when extending or modifying the provisions of such a referenced document.

- should** indicates a recommendation to do something
- should not** indicates a recommendation not to do something
- may** indicates permission to do something
- need not** indicates permission not to do something

The construction "may not" is ambiguous and is not used in normative elements. The unambiguous constructions "might not" or "shall not" are used instead, depending upon the meaning intended.

- can** indicates that something is possible
- cannot** indicates that something is impossible

The constructions "can" and "cannot" are not substitutes for "may" and "need not".

- will** indicates that something is certain or expected to happen as a result of action taken by an agency the behaviour of which is outside the scope of the present document
- will not** indicates that something is certain or expected not to happen as a result of action taken by an agency the behaviour of which is outside the scope of the present document
- might** indicates a likelihood that something will happen as a result of action taken by some agency the behaviour of which is outside the scope of the present document

might not indicates a likelihood that something will not happen as a result of action taken by some agency the behaviour of which is outside the scope of the present document

In addition:

is (or any other verb in the indicative mood) indicates a statement of fact

is not (or any other negative verb in the indicative mood) indicates a statement of fact

The constructions "is" and "is not" do not indicate requirements.

Introduction

In recent years, video services are evolving from traditional two-dimensional formats to beyond 2D video, which offer users a more lifelike and immersive experience. Research studies indicate that the beyond 2D market was valued at approximately multi-million USD in 2023 and is anticipated to register a CAGR (Compound Annual Growth Rate) of over 24.5% between 2024 and 2032 [2][3][4].

A variety of beyond 2D video formats and video compression technologies are available and emerging. Therefore, in order to determine appropriate beyond 2D video formats for different services, it is essential to evaluate their feasibility and performance, considering implementation constraints, performance indicators, and interoperability considerations. In addition, advanced network capabilities and service extension also need to be investigated to meet the delay and data rate requirements of beyond 2D-related services.

This document provides an overview of available and emerging beyond 2D video formats and compression technologies, which are mostly related to specific types of capturing systems and display technologies; documents a set of end-to-end reference scenarios and workflows for beyond 2D video; analyses 3GPP-defined video compression technologies and potential new technologies to support each documented scenario; identifies gaps and offer recommendations to potentially extend 3GPP video specifications and capabilities.

1 Scope

The present document collects beyond 2D video formats within 3GPP services, as well as a set of beyond 2D video end-to-end reference scenarios and corresponding workflows. It also documents relevant implementation constraints, performance characteristics, and interoperability requirements of existing 3GPP codecs as well as potentially new codecs to support these scenarios. The primary scope of the present document includes the following aspects:

1. Identify and document beyond 2D formats, that are market-relevant within the next few years, generated from established and emerging capturing systems (including cameras for spatial video capturing), contribution, and usable on display technologies (smartphones, VR HMDs, AR glasses, autostereoscopic and multiscopic displays).
2. Establish and document a set of beyond 2D video end-to-end reference scenarios, including real-time communication, streaming services, split rendering, and messaging and corresponding workflows (capturing, encoding, packaging, delivery, decoding, rendering, including general constraints on latency, as well as complexity) to support 3GPP network related delivery and devices leveraging the generation or display technologies. This includes identifying and defining relevant beyond 2D formats in the context of above workflows, and representation technologies to support delivery of these formats within 3GPP networks.
3. Prioritize the scenarios and the associated formats based on market relevance for further evaluation.
4. Define concrete evaluation framework per scenario (test conditions, KPIs, Metrics, test sequences, agreed reference signals) based on the above prioritized reference scenarios, and evaluate the feasibility and performance of existing 3GPP codecs as well as potentially new codecs to support the scenarios.
5. Based on the findings in steps 1, 2, and 4 document (i) interoperability requirements, (ii) traffic characteristics and (iii) potential QoS optimizations or requirements, to support the above workflows and evaluate the feasibility of new formats with different services, considering the implementation constraints and performance indicators such as encoding, decoding, and rendering complexity, bandwidth utilization, and interoperability considerations.
6. Based on the findings in steps 1, 2, 4 and 5, identify potential gaps or deficiencies of existing 3GPP codecs, and offer recommendations to potentially extend 3GPP video specifications and capabilities.

Identify potential areas for normative work as the next phase and communicate with other 3GPP WGs regarding relevant aspects related to the study to the extent needed.

2 References

The following documents contain provisions which, through reference in this text, constitute provisions of the present document.

- References are either specific (identified by date of publication, edition number, version number, etc.) or non-specific.
- For a specific reference, subsequent revisions do not apply.
- For a non-specific reference, the latest version applies. In the case of a reference to a 3GPP document (including a GSM document), a non-specific reference implicitly refers to the latest version of that document *in the same Release as the present document*.

- [1] 3GPP TR 21.905: "Vocabulary for 3GPP Specifications".
- [2] Allied Market Research, "3D Technology Market Size, Share, Competitive Landscape and Trend Analysis Report by Product, Application : Global Opportunity Analysis and Industry Forecast, 2021-2030.", www.alliedmarketresearch.com/3d-technology-market.
- [3] Mordor Intelligence, "Mobile 3D Market Size & Share Analysis - Growth Trends & Forecasts (2024 - 2029).", <https://www.mordorintelligence.com/industry-reports/mobile-3d-market>.

- [4] Grand View Research, "Immersive Technology Market Size, Share & Trends Analysis Report By Component (Hardware, Software, Services), By Technology, By Application, By Industry, By Region, And Segment Forecasts, 2023 - 2030.", <https://www.grandviewresearch.com/industry-analysis/immersive-technology-market-report>.
- [5] 3GPP TR 26.955: "Video codec characteristics for 5G-based services and applications".
- [6] 3GPP TS 26.118: "Virtual Reality (VR) profiles for streaming applications".
- [7] 3GPP TS 26.119: "Media Capabilities for Augmented Reality".
- [8] 3GPP TS 26.143: "Messaging Media Profiles".
- [9] 3GPP TS 26.511: "5G Media Streaming (5GMS); Profiles, codecs and formats".
- [10] 3GPP TR 26.966: "Evaluation of new HEVC coding tools".
- [11] 3GPP TS 26.265: "Media Delivery: Video Capabilities and Operating Points".
- [12] Apple HEVC Stereo Video - Interoperability Profile (Beta), Version 0.9, June 21, 2023, <https://developer.apple.com/av-foundation/HEVC-Stereo-Video-Profile.pdf>
- [13] A. Quested and B. Zegel, "3D-TV production standards - first report of the ITU-R Rapporteurs", EBU Technical Review, 2011 Q2, https://tech.ebu.ch/publications/trev_2011-Q2_3dtv_quested
- [14] Mike Swanson, "Spatial Video", March 7 2024, <https://blog.mikeswanson.com/spatial-video/>
- [15] Video Contour Map Payload, Version 0.9, June 21, 2023, <https://developer.apple.com/av-foundation/Video-Contour-Map-Metadata.pdf>
- [16] ITU-T H.273 (09/23), Coding-independent code points for video signal type identification
- [17] M. Satya, "3D Image Reconstruction From Multi-View Stereo", https://medium.com/@satya15july_11937/3d-image-reconstruction-from-multi-view-stereo-782e6912435b, March, 2023.
- [18] S. Khan and S. S. Channappayya, "Estimating Depth-Salient Edges and Its Application to Stereoscopic Image Quality Assessment," IEEE Transactions on Image Processing, vol. 27, no. 12, pp. 5892 - 5903, 2018, doi: 10.1109/TIP.2018.2860279.
- [19] Greg Turk, The Polygon File Format, Stanford University, 1994.
- [20] Volumetric Format Association VFA, <https://www.volumetricformat.org/>
- [21] ["The OpenGL Graphics System: A Specification" \(PDF\). 4.0 \(Core Profile\). March 11, 2010.](#)
- [22] V-PCC, Visual volumetric video-based coding (V3C) and video-based point cloud compression (V-PCC), ISO/IEC 23090-5 2nd Ed, Nov 2023.
- [23] G-PCC, Geometry-based point cloud compression, ISO/IEC 23090-9, Mar 2023
- [24] Draco Bitstream Specification, <https://google.github.io/draco/spec/>
- [25] MPEG 115, Use cases for Point Cloud Compression, [https://mpeg.chiariglione.org/sites/default/files/files/standards/parts/docs/w16331_Use_Cases_for_Point_Cloud_Compression_\(PCC\)_0.docx](https://mpeg.chiariglione.org/sites/default/files/files/standards/parts/docs/w16331_Use_Cases_for_Point_Cloud_Compression_(PCC)_0.docx)
- [26] OpenCV, <https://opencv.org>.
- [27] Colmap, <https://colmap.github.io/index.html>
- [28] AliceVision Photogrammetric Computer Vision Framework, <https://alicevision.org>.
- [29] Open Multiple View Geometry (openMVG), <https://github.com/openMVG/openMVG>.
- [30] Immersive Video Depth Estimation (IVDE), <https://gitlab.com/mpeg-i-visual/ivde>.

- [31] Test model for MPEG immersive video, <https://gitlab.com/mpeg-i-visual/tmiv>.
- [32] Reference view synthesizer, <https://gitlab.com/mpeg-i-visual/rvs>.
- [33] Open Realtime Depth Image Based Renderer (OpenDIBR), <https://github.com/IDLabMedia/open-dibr>.
- [34] A. Dziembowski, D. Mieloch, J. Stankowski and A. Grzelka, "IV-PSNR – the objective quality metric for immersive video applications," in IEEE Transactions on Circuits and Systems for Video Technology, doi: [10.1109/TCSVT.2022.3179575](https://doi.org/10.1109/TCSVT.2022.3179575), software: <https://gitlab.com/mpeg-i-visual/ivpsnr>
- [35] Quality Metrics for Immersive Video (QMIV), <https://gitlab.com/mpeg-i-visual/ivpsnr>.
- [35] Gerhard Tech, Ying Chen, Karsten Müller, Jens-Rainer Ohm, Anthony Vetro, Ye-Kui Wang, Overview of the Multiview and 3D Extensions of High Efficiency Video Coding, IEEE Transactions on Circuits and Systems for Video Technology, vol. 26, no. 1, January 2016.
- [36] MPEG 136, CfP for Dynamic Mesh Coding, https://www.mpeg.org/wp-content/uploads/mpeg_meetings/136_OnLine/w20972.zip
- [37] MPEG 134, Use cases for Mesh Coding, https://www.mpeg.org/wp-content/uploads/mpeg_meetings/134_OnLine/w20364.zip
- [38] Y. Choi, J. -B. Jeong, S. Lee and E. -S. Ryu, "Overview of the Video-based Dynamic Mesh Coding (V-DMC) Standard Work," 2022 13th International Conference on Information and Communication Technology Convergence (ICTC), Jeju Island, Korea, Republic of, 2022, pp. 578-581, doi: [10.1109/ICTC55196.2022.9952734](https://doi.org/10.1109/ICTC55196.2022.9952734).
- [39] Information technology - Coding of audio-visual objects - Part 16: Animation Framework eXtension (AFX), ISO/IEC 14496-16.
- [40] Mammou, K., Kim, J., Tourapis, A. M., Podborski, D., & Flynn, D. (2022, September). Video and subdivision based mesh coding. In 2022 10th European Workshop on Visual Information Processing (EUVIP) (pp. 1-6). IEEE.
- [41] HS, Yang. and X. de Foy, "RTP Payload for V-DMC", Work in Progress, Internet-Draft, draft-hsyang-avtcore-rtp-vdmc-00, 18 October 2024, <<https://www.ietf.org/id/draft-hsyang-avtcore-rtp-vdmc-00.html>>.
- [42] ISO/IEC 12113:2022, Information technology — Runtime 3D asset delivery format — Khronos glTF™ 2.0, International Organization for Standardization, 2022.
- [43] Dynamic Mesh Documentation - Unigine Developer. Available at: https://developer.unigine.com/en/docs/latest/objects/objects/mesh_dynamic/ (Accessed: 19 February 2025).
- [44] OWLII Dynamic Human Textured Mesh Sequence Dataset MPEG Point Cloud Compression. Available at: <https://mpeg-pcc.org/index.php/pcc-content-database/owlii-dynamic-human-textured-mesh-sequence-dataset/> (Accessed: 19 February 2025).
- [45] Pagés, Rafael & Amplianitis, Konstantinos & Ondrej, Jan & Zerman, Emin & Smolic, Aljosa. (2022). Volograms & V-SENSE Volumetric Video Dataset. [10.13140/RG.2.2.24235.31529/1](https://doi.org/10.13140/RG.2.2.24235.31529/1).
- [46] Q. Yang, J. Jung, T. Deschamps, X. Xu and S. Liu, "TDMD: A Database for Dynamic Color Mesh Quality Assessment Study," in IEEE Transactions on Visualization and Computer Graphics, doi: [10.1109/TVCG.2024.3451526](https://doi.org/10.1109/TVCG.2024.3451526).
- [47] M. Corsini, E. D. Gelasca, T. Ebrahimi, and M. Barni, "Water marked 3-d mesh quality assessment," IEEE Trans. Multimedia, vol. 9, no. 2, pp. 247 - 256, 2007.
- [48] FTorkhani, K. Wang, and J.-M. Chassery, "Perceptual quality assessment of 3d dynamic meshes: Subjective and objective studies," Signal Processing: Image Communication, vol. 31, pp. 185 - 204, 2015.

- [49] B. ITU-R RECOMMENDATION, “Methodology for the subjective assessment of the quality of television pictures,” International Telecommunication Union, 2002.
- [50] P. ITU-T RECOMMENDATION, “Subjective video quality assessment methods for multimedia applications,” International Telecommunication Union, 1999.
- [51] Y. Nehmé, F. Dupont, J.-P. Farrugia, P. Le Callet, and G. Lavoué, “Visual quality of 3d meshes with diffuse colors in virtual reality: Subjective and objective evaluation,” *IEEE Trans. Visualization and Computer Graphics*, vol. 27, no. 3, pp. 2202 – 2219, 2020.
- [52] ITU-T, “Subjective test method for interactive virtual reality applications,” <https://www.itu.int/ITU-T/workprog/wp-item.aspx?isn=17817>.
- [53] MPEG, mpeg-pcc-mmetric, <https://github.com/MPEGGroup/mpeg-pcc-mmetric>
- [54] MPEG, Representative Renderer, <https://github.com/MPEGGroup/mpeg-3dg-renderer>
- [55] ISO/IEC 23090-29 Video-based dynamic mesh coding (V-DMC)
- [56] Ben Mildenhall, Pratul P. Srinivasan, Matthew Tancik, Jonathan T. Barron, Ravi Ramamoorthi, and Ren Ng. 2021. NeRF: representing scenes as neural radiance fields for view synthesis. *Commun. ACM* 65, 1 (January 2022), 99–106. <https://doi.org/10.1145/3503250>
- [57] RABBY, AKM SHAHARIAR AZAD and Chengcui Zhang. “BeyondPixels: A Comprehensive Review of the Evolution of Neural Radiance Fields.” *ArXiv abs/2306.03000* (2023): n. pag.
- [57] Daniel Duckworth, Peter Hedman, Christian Reiser, Peter Zhizhin, Jean-François Thibert, Mario Lučić, Richard Szeliski, and Jonathan T. Barron. 2024. SMERF: Streamable Memory Efficient Radiance Fields for Real-Time Large-Scene Exploration. *ACM Trans. Graph.* 43, 4, Article 63 (July 2024), 13 pages. <https://doi.org/10.1145/3658193>
- [58] Müller, T., Evans, A., Schied, C., & Keller, A. (2022). Instant neural graphics primitives with a multiresolution hash encoding. *ACM transactions on graphics (TOG)*, 41(4), 1-15.
- [59] Gao, Kyle et al. “NeRF: Neural Radiance Field in 3D Vision, A Comprehensive Review.” (2022).
- [59] G. Lafruit, Y. Liao, and G. Bang, “AhG on Implicit Neural Video Representations (INVR),” *ISO/IEC JTC1/SC 29/WG04, M60641*, Oct. 2022. G. Lafruit, Y. Liao, and G. Bang, “AhG on Implicit Neural Video Representations (INVR),” *ISO/IEC JTC1/SC 29/WG04, M60641*, Oct. 2022
- [69] Li, Sicheng et al. “NeRFFCodec: Neural Feature Compression Meets Neural Radiance Fields for Memory-Efficient Scene Representation.” *ArXiv abs/2404.02185* (2024): n. pag.
- [70] Dong-Ha Kim, Jun Young Jeong, Gwangsoon Lee, and Jae-Gon Kim "Compression method of NeRF model using NNC and VVC", *Proc. SPIE 13164, International Workshop on Advanced Imaging Technology (IWAIT) 2024*, 131642V (2 May 2024); <https://doi.org/10.1117/12.3019533>
- [71] Gershun A (1939) The light field. Moscow, 1936. Translated by Moon P, Timoshenko G in *J Math Phys XVIII*:51–151
- [72] Marc Levoy and Pat Hanrahan. 1996. Light field rendering. In *Proceedings of the 23rd annual conference on Computer graphics and interactive techniques (SIGGRAPH '96)*.
- [73] Google Light Stage X4, <https://www.fxguide.com/featured/exclusive-paul-debevec-and-the-light-stage-research-at-google/>
- [74] USC Lightstage X6, <https://vgl.ict.usc.edu/Data/LightStage/>
- [75] Zhou, Taotao, et al. "Relightable neural human assets from multi-view gradient illuminations." *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2023.

- [76] Michael Broxton, John Flynn, Ryan Overbeck, Daniel Erickson, Peter Hedman, Matthew Duvall, Jason Dourgarian, Jay Busch, Matt Whalen, and Paul Debevec. 2020. Immersive light field video with a layered mesh representation. *ACM Trans. Graph.* 39, 4, Article 86 (August 2020), 15 pages. <https://doi.org/10.1145/3386569.3392485>
- [77] M. B. de Carvalho et al., "A 4D DCT-Based Lenslet Light Field Codec," 2018 25th IEEE International Conference on Image Processing (ICIP), Athens, Greece, 2018, pp. 435-439, doi: 10.1109/ICIP.2018.8451684.
- [78] ISO/IEC JTC 1/SC 29/WG 04, w24268, "Overview of lenslet video coding activities", MPEG 147, Sapporo.
- [79] Daniel, Jamison R. et al. "Initial work on development of an open Streaming Media Standard for Field of Light Displays (SMFoLD)." *SD&A* (2018).
- [80] X. Min, J. Zhou, G. Zhai, P. Le Callet, X. Yang and X. Guan, "A Metric for Light Field Reconstruction, Compression, and Display Quality Evaluation," in *IEEE Transactions on Image Processing*, vol. 29, pp. 3790-3804, 2020, doi: 10.1109/TIP.2020.2966081.
- [81] ISO/IEC JTC 1/SC29/WG1 N100306, REQ "Use Cases and Requirements for Light Field Quality Assessment v5.0", 97th JPEG Meeting, Online, October 2022.
- [82] What you should know about light-field,
<https://www.digitalmedia.fraunhofer.de/en/mediainformation/trendbrochures/trendbrochure-2021/what-you-should-know-about-light-field-.html>
- [83] "A Visual Introduction to the Past, Present, and Future of Light Field Technology",
<https://cubicleninjas.com/light-field-technology/>
- [84] Ruben Verhack, "AI-Driven Breakthroughs in Image-Based Rendering: Light Fields, SMOE, Gaussian Splatting, NeRFs and beyond", <https://cubicleninjas.com/light-field-technology/>, Tech Posts Computer Graphics, <https://blog.datameister.ai/ai-driven-breakthroughs-image-based-rendering>, February 2024.
- [85] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkuehler, and George Drettakis. 2023. 3D Gaussian Splatting for Real-Time Radiance Field Rendering. *ACM Trans. Graph.* 42, 4, Article 139 (August 2023), 14 pages. <https://doi.org/10.1145/3592433>
- [86] Wei, Meng et al. "Normal-GS: 3D Gaussian Splatting with Normal-Involved Rendering." *ArXiv abs/2410.20593* (2024): n. pag.
- [87] Özyeşil, Onur, et al. "A survey of structure from motion*." *Acta Numerica* 26 (2017): 305-364.
- [88] T. Wu, Y.-J. Yuan, L.-X. Zhang, J. Yang, Y.-P. Cao, L.-Q. Yan, and L. Gao, "Recent advances in 3d gaussian splatting," *Computational Visual Media*, pp. 1–30, 2024.
- [89] X. Lei, M. Wang, W. Zhou, and H. Li, "Gaussnav: Gaussian splatting for visual navigation," *arXiv preprint arXiv:2403.11625*, 2024.
- [89] 3DGS.zip: A survey on 3D Gaussian Splatting Compression Methods, <https://3dgs.zip/>
- [90] Dalal, Anurag et al. "Gaussian Splatting: 3D Reconstruction and Novel View Synthesis: A Review." *IEEE Access* 12 (2024): 96797-96820.
- [91] Dalal, Anurag & Hagen, Daniel & Robbersmyr, Kjell & Knausgård, Kristian. (2024). Gaussian Splatting: 3D Reconstruction and Novel View Synthesis, a Review. 10.48550/arXiv.2405.03417.
- [92] Nicolas Moenne-Loccoz, Ashkan Mirzaei, Or Perel, Riccardo de Lutio, Janick Martinez Esturo, Gavriel State, Sanja Fidler, Nicholas Sharp and Zan Gojcic; "3D Gaussian Ray Tracing: Fast Tracing of Particle Scenes", *ACM Transactions on Graphics and SIGGRAPH Asia*, 2024.
- [93] Zhu, Huixin et al. "Scene reconstruction techniques for autonomous driving: a review of 3D Gaussian splatting." *Artif. Intell. Rev.* 58 (2024): 30.

- [94] Jorge L. Charco, Angel D. Sappa, Boris X. Vintimilla, and Henry O. Velesaca. 2021. Camera pose estimation in multi-view environments: From virtual scenarios to the real world. *Image Vision Comput.* 110, C (Jun 2021). <https://doi.org/10.1016/j.imavis.2021.104182>
- [95] Huang, Zhentao and Minglun Gong. "Textured-GS: Gaussian Splatting with Spatially Defined Color and Opacity." *ArXiv abs/2407.09733* (2024): n. pag.
- [96] ISO/IEC 23090-14:2024/Amd.1:2025. "ISO/IEC 23090-14 2nd edition DAM 1 Support of MPEG-I immersive audio, scene understanding and other extensions"
- NOTE: The latest version is available as MDS25320_WG03_N01573: "Potential improvement of ISO/IEC 23090-14 2nd edition DAM 1 Support of MPEG-I immersive audio, scene understanding and other extensionsWG 03 MPEG Systems".
- [97] Wang, Y., Lu, Z., Cao, P. et al. How Live Streaming Changes Shopping Decisions in E-commerce: A Study of Live Streaming Commerce. *Comput Supported Coop Work* 31, 701 – 729 (2022). <https://doi.org/10.1007/s10606-022-09439-2>
- [98] Xie, Junyuan et al. "Deep3D: Fully Automatic 2D-to-3D Video Conversion with Deep Convolutional Neural Networks." *European Conference on Computer Vision* (2016).
- [99] Dumić, E. et al.. "Transmission of 3D Video Content. In: Assunção, P., Gotchev, A. (eds) 3D Visual Content Creation, Coding and Delivery." *Signals and Communication Technology* (2019). Springer, Cham. https://doi.org/10.1007/978-3-319-77842-6_8
- [100] Schierl, Thomas and Sam Narasimhan. "Transport and Storage Systems for 3-D Video Using MPEG-2 Systems, RTP, and ISO File Format." *Proceedings of the IEEE* 99 (2011): 671-683.
- [101] 3GPP TR 26.905 V 18.0.0: "Mobile stereoscopic 3D video"
- [102] A. Banitalebi-Dehkordi, M. T. Pourazad and P. Nasiopoulos, "A human visual system-based 3D video quality metric," 2012 International Conference on 3D Imaging (IC3D), Liege, Belgium, 2012, pp. 1-5, doi: 10.1109/IC3D.2012.6615146.
- [103] Banitalebi-Dehkordi, Amin, Mahsa T. Pourazad, and Panos Nasiopoulos. "An efficient human visual system based quality metric for 3D video." *Multimedia Tools and Applications* 75, no. 8 (2016): 4187-4215.
- [104] Recommendation ITU-R BT.500-13, "Methodology for the subjective assessment of the quality of the television pictures", 2012.
- [109] ISO/IEC JTC1/SC29/WG11 (MPEG), Document N12036, "Call for proposals on 3D video coding technology," 96th MPEG meeting, Geneva, March 2011.
- [110] Q. Hyunh-Thu, P. L. Callet, and M. Barkowsky, "Video quality assessment: from 2D to 3D challenges and future trends," *IEEE 17th International Conference on Image Processing, (ICIP)*, pp.4025-4028, 2010.
- [111] Wei Bao, Wei Wang, Yuhua Xu, Yulan Guo, Siyu Hong, Xiaohu Zhang. InStereo2K: A large real dataset for stereo matching in indoor scenes. *SCIENCE CHINA Information Sciences*. 2020.
- [112] D. Scharstein, H. Hirschmüller, Y. Kitajima, G. Krathwohl, N. Nesić, X. Wang, and P. Westling. High-resolution stereo datasets with subpixel-accurate ground truth. In *German Conference on Pattern Recognition (GCPR 2014)*, Münster, Germany, September 2014
- [113] HTM Codec Software, version 16.3, https://hevc.hhi.fraunhofer.de/svn/svn_3DVCSoftware/branches/HTM-16.3-fixes/cfg/MV-HEVC/
- [114] V. Baroncini, K. Müller, and S. Shinya (editors), "MV-HEVC verification test report", JCT3V-N1001, May 2016.
- [115] VQEG 3DTV Group, Test Plan for Evaluation of Video Quality Models for Use with Stereoscopic Three-Dimensional Television Content, 2012

- [116] Recommendation ITU-R P.914, “Display requirements for 3D video quality assessment”, 2016.
- [117] Recommendation ITU-R P.915, “Subjective assessment methods for 3D video quality”, 2016.
- [118] Recommendation ITU-R P.916, “Information and guidelines for assessing and minimizing visual discomfort and visual fatigue from 3D video”, 2016.
- [119] Guodong Chen, Sizhe Wang, Jacob Chakareski, Dimitrios Koutsonikolas, and Mallesham Dasari. 2025. Spatial Video Streaming on Apple Vision Pro XR Headset. In Proceedings of the 26th International Workshop on Mobile Computing Systems and Applications (HotMobile '25). Association for Computing Machinery, New York, NY, USA, 115 – 120. <https://doi.org/10.1145/3708468.3711878>
- [120] Sizhe Wang, Mingkun Liu, Mallesham Dasari, and Dimitrios Koutsonikolas. 2024. A First Look at Apple’s Stereoscopic Video and its Potential in Live Video Streaming for XR Headsets. In Proceedings of the 30th Annual International Conference on Mobile Computing and Networking (ACM MobiCom '24). Association for Computing Machinery, New York, NY, USA, 1617 – 1619. <https://doi.org/10.1145/3636534.3697437>
- [121] KDDI, Transmission experiment using real-time codec compliant with the latest international standard of point cloud compression, <https://www.kddi-research.jp/english/newsrelease/2023/012401.html>
- [122] SBTVD Forum, Brazilian Forum for digital terrestrial TV, https://forumsbtvd.org.br/tv3_0/
- [123] VFA, The Volumetric Format Association, <https://www.volumetricformat.org/>
- [124] Arcturus, on-demand streaming of volumetric video, <https://arcturus.studio/stream/>
- [125] 8i, stream volumetric video to any device, browser, or VR/AR headset, <https://8i.com/stream/>
- [126] Broadpeak, V3C standardized content distribution at scale, <https://broadpeak.tv/newsroom/mpeg-v3c-standardized-content-distribution-at-scale/>
- [127] Arcturus, playback solution with broad delivery capabilities, <https://arcturus.studio/playback/>
- [128] 5G-MAG, V3C Immersive platform, <https://5g-mag.github.io/Getting-Started/pages/v3c-immersive-platform/>
- [129] DVB project, DVB Study Mission Report S101, DVB Bluebook S101
- [130] Ultra Video Group, Voxelized Point Cloud Dataset for Visual Volumetric Video-based Coding, <https://ieeexplore.ieee.org/document/10178589>
- [131] ISO/IEC 23090-5 Visual Volumetric Video-based Coding (V3C) and Video-based Point Cloud Compression (V-PCC) – 3rd edition
- [132] ISO/IEC 23090-10 Carriage of visual volumetric video-based coding data – 1st edition
- [133] By C. Guede et al., IBC 2023 Tech Papers, <https://www.ibc.org/technical-papers/ibc2023-tech-papers-efficient-delivery-and-rendering-on-client-devices-via-mpeg-i-standards-for-emerging-volumetric-video-experiences/10277.article>
- [134] Futuresource Consulting, Spotlight on HEVC, https://www.interdigital.com/white_papers/spotlight-on-hevc-the-codec-of-choice-for-the-video-streaming-industry
- [135] Renderpeople, <https://renderpeople.com/>
- [136] Volucap, <https://volucap.com/>
- [137] MPEG WG7, Call for Proposals for Point Cloud Compression V2, <https://mpeg.chiariglione.org/standards/mpeg-i/point-cloud-compression/call-proposals-point-cloud-compression-v2.html>

- [138] PCQM: A Full-Reference Quality Metric for Colored 3D Point Clouds, Gabriel Meynet, Yana Nehmé, Julie Digne, Guillaume Lavoué, <https://hal.science/hal-02529668/document>
- [139] MPEG, mpeg-pcc-mmetric V1_1_7, <https://github.com/MPEGGroup/mpeg-pcc-mmetric>
- [140] MPEG, Representative Renderer release 8.0, <https://github.com/MPEGGroup/mpeg-3dg-renderer>
- [141] MPEG, Subjective verification test report for V-PCC, https://www.mpeg.org/wp-content/uploads/mpeg_meetings/136_OnLine/w20992.zip
- [142] ISO/IEC 23090-10 Carriage of visual volumetric video-based coding data – 1st edition
- [143] [XD Productions, https://xdprod.com/](https://xdprod.com/)
- [144] Renderpeople free 4D People sample: <https://4dpeople.com/samples/>
- [145] Renderpeople 4D People catalogue: <https://4dpeople.com/>
- [146] [Ultra Video Group – UVG-VPC Dataset, https://ultravideo.fi/UVG-VPC/index.html](https://ultravideo.fi/UVG-VPC/index.html)
- [147] MPEG, V-PCC test model tmc2 release R25.0, <https://github.com/MPEGGroup/mpeg-pcc-tmc2>
- [148] MPEG, Random Access configuration, <https://github.com/MPEGGroup/mpeg-pcc-tmc2/blob/master/cfg/>
- [149] MPEG, Subjective verification test report for V-PCC, https://www.mpeg.org/wp-content/uploads/mpeg_meetings/136_OnLine/w20992.zip
- [150] MPEG, Rate configuration, <https://github.com/MPEGGroup/mpeg-pcc-tmc2/tree/master/cfg/rate>
- [151] List of camera paths and splat blend parameter options for each tested sequence
- [152] SBTVD, SBTVD TV 3.0 test report for video, https://forumsbtvd.org.br/wp-content/uploads/2021/12/SBTVD-TV_3_0-VC-Report.pdf
- [153] <https://www.kddi-research.jp/newsrelease/2025/020601.html>
- [154] CfP for Dynamic Mesh Coding, https://www.mpeg.org/wp-content/uploads/mpeg_meetings/136_OnLine/w21000.zip
- [155] Ilola, L., Kondrad, L., Schwarz, S., & Hamza, A. (2022). An overview of the MPEG standard for storage and transport of visual volumetric video-based coding. *Frontiers in Signal Processing*, 2, 883943.
- [156] Jing, Liqi. “Research on Video-based Dynamic Mesh Compression Technology and Proposed Improvements.” *Advances in engineering research/Advances in Engineering Research*, 2024, pp. 47–57, doi:10.2991/978-94-6463-518-8_6.
- [157] 3GPP TR 22.870 V0.3.1, Study on 6G Use Cases and Service Requirements
- [158] ITU-T Recommendation H.264 (08/2021): "Advanced video coding for generic audiovisual services".
- [159] ITU-T Recommendation H.265 (08/2021): "High efficiency video coding".
- [160] ISO/IEC 23090-10:2022 (Amd1), “Information Technology — Coded Representation of Immersive media — Part 10: Carriage of Visual Volumetric Video-Based Coding Data”
- [161] Guede et al., IBC 2023, “Efficient Delivery and Rendering on Client Devices via MPEG-I Standards for Emerging Volumetric Video Experiences”. <https://www.ibt.org/technical-papers/ibt2023-tech-papers-efficient-delivery-and-rendering-on-client-devices-via-mpeg-i-standards-for-emerging-volumetric-video-experiences/10277.article>
- [162] Dziembowski, B. Kroon, J. Jung (Eds.), Common test conditions for MPEG immersive video, ISO/IEC JTC 1/SC 29/WG 04 N 0372, July 2023, Geneva.

- [163] D. Mieloch (Ed.), Verification test report of MPEG immersive video, ISO/IEC JTC 1/SC 29/WG 04 N 0341, April 2023, Antalya.
- [164] B. Brand, Michel Bätz, Joachim Keinert, Camorph: a toolbox for conversion between camera parameter conversions, The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, url: <https://github.com/Fraunhofer-IIS/camorph>, 2022.
- [165] ISO/IEC 14496-15:2024: Carriage of network abstraction layer (NAL) unit structured video in the ISO base media file format.
- [166] ISO/IEC 23090-12:2023: MPEG immersive video.
- [167] ITU-T H.265:2024 | ISO/IEC 23008-2:2024, Annex G: Multiview high efficiency video coding (MV-HEVC).
- [168] J.Y. Jeong, J. Kim, B.H. Lee, K.J. Yun, W. Cheong, S.H. Yoo, [INVR] Multiview dataset Classroom and Bartender for 3D INVR activity, ISO/IEC JTC1/SC29/WG4 MPEG VC/M69151, Sapporo, Japan, July 2024.
- [169] Overview and Efficiency of Decoder-Side Depth Estimation in MPEG Immersive Video, IEEE Transactions on Circuits and Systems for Video Technology, doi: 10.1109/TCSVT.2022.3162916, ode: <https://gitlab.com/mpeg-i-visual/ivde>.
- [170] Encoder guidelines for MPEG immersive video, ISO/IEC JTC 1/SC 29/WG 04/N 660, April 2025, url: https://www.mpeg.org/wp-content/uploads/mpeg_meetings/150_OnLine/w25085.zip, Online.
- [171] Sun, Y., Lu, A. and Yu, L., 2017. Weighted-to-spherically-uniform quality evaluation for omnidirectional video. IEEE signal processing letters, 24(9), pp.1408-1412.
- [172] A. Dziembowski, W. Nowak, J. Stankowski, "IV-SSIM - The Structural Similarity Metric for Immersive Video", Applied Sciences, Vol. 14, No. 16, Aug 2024, doi: 10.3390/app14167090.

3 Definitions of terms, symbols and abbreviations

This clause and its three subclauses are mandatory. The contents shall be shown as "void" if the TS/TR does not define any terms, symbols, or abbreviations.

3.1 Terms

For the purposes of the present document, the terms given in 3GPP TR 21.905 [1] and the following apply. A term defined in the present document takes precedence over the definition of the same term, if any, in 3GPP TR 21.905 [1].

Beyond 2D (B2D): refers to video technologies that go beyond traditional two-dimensional video, offering enhanced depth, or immersive experiences and may be combined with interactivity.

B2D Video Encoder: executes a processing step that will result in a Beyond 2D video bitstream that includes a digitally compressed version of the B2D video along with optional metadata.

B2D Video Decoder: decodes the B2D video bitstream and recovers a B2D video format.

Volumetric Video: A frame-based immersive experience whereby each frame represents a volumetric region in 3D space in which any point is either non-occupied or has a colour that may depend on the viewing direction.

3.2 Symbols

For the purposes of the present document, the following symbols apply:

Symbol format (EW)

<symbol> <Explanation>

3.3 Abbreviations

For the purposes of the present document, the abbreviations given in 3GPP TR 21.905 [1] and the following apply. An abbreviation defined in the present document takes precedence over the definition of the same abbreviation, if any, in 3GPP TR 21.905 [1].

AAC	Advanced Audio Coding
ABR	Adaptive BitRate
AI	Artificial Intelligence
APK	Android Package
AR	Augmented Reality
API	Application Programming Interface
B2D	Beyond 2D Video
BMP	Bitmap
CBR	Constant BitRate
CGI	Computer-Generated Imagery
CMAF	Common Media Application Format
CNN	Convolutional Neural Network
CPU	Central Processing Unit
CSV	Comma-Separated Values
DASH	Dynamic Adaptive Streaming over HTTP
ERP	Equi-Rectangular Projection
EXE	Executable File
FHD	Full HD
FPS	Frames Per Second
GAN	Generative Adversarial Network
GIF	Graphics Interchange Format
GOP	Group-Of-Pictures
G-PCC	Geometry-based Point Cloud Compression
GPU	Graphics Processing Unit
HDR	High Dynamic Range
HEIF	High Efficiency Image File Format
HEVC	High Dynamic Range
HLS	HTTP Live Streaming
HMD	Head-Mounted Display
HTML	Hypertext Markup Language
HTTP	Hypertext Transfer Protocol
ISO BMFF	ISO Base Media File Format
ITU	International Telecommunication Union
JPEG	Joint Photographic Experts Group
JSON	JavaScript Object Notation
JVET	Joint Video Experts Team
KPI	Key Performance Indicator
LiDAR	Light Detection and Ranging
LVC	Lenslet Video Coding
MDF	Media Descriptor File
MIV	MPEG Immersive Video
MPEG	Moving Picture Experts Group
MSE	Mean Squared Error
MV-HEVC	Multiview High Efficiency Video Coding
NeRF	Neural Radiance Fields
OBJ	Object File Format
OpenGL	Open Graphics Library
OpenXR	Open Extended Reality
PLY	Polygon File Format
PSNR	Peak Signal-to-Noise Ratio
QoE	Quality of Experience
QoS	Quality of Service
RA	Random Access






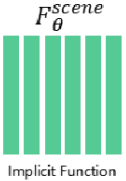
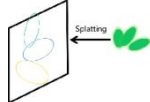
RAP	Random Access Point
RGB	Red, Green, Blue
RTP	Real-time Transport Protocol
SDK	Software Development Kit
SDR	Standard Dynamic Range
SMFoLD	Streaming Media Standard for Field of Light Displays
TIFF	Tagged Image File Format
ToF	Time of Flight
UHD	Ultra High Definition
URL	Uniform Resource Locator
V-DMC	Video-based Dynamic Mesh Coding
V-PCC	Video-based Point Cloud Compression
VR	Virtual Reality
WebGL	Web Graphics Library
WebGPU	Web Graphics Processing Unit
WebXR	Web Extended Reality
YCbCr	Luminance, Blue Chrominance, Red Chrominance
3DGS	3D Gaussian Splatting

4 Beyond 2D Video Formats

4.1 Introduction

This clause provides an overview of the Beyond 2D Video formats that have reached a certain amount of maturity as they can be generated from established and emerging capturing systems (including cameras for spatial video capturing) and can likely be rendered on existing display technologies (smartphones, VR HMDs, AR glasses, autostereoscopic and multiscopic displays). These formats include: stereoscopic 3D video, Multi-view plus Depth, dense dynamic point clouds and dynamic meshes. Emerging formats such as Neural Radiance Fields (NeRF), light fields, and 3D Gaussian Splatting (3DGS) are documented as formats under research. Table 4.1-1 summarizes the Beyond 2D Video formats documented in this study, highlighting their representation principles, advantages, challenges and compression technologies.

Table 4.1-1 Summary of Beyond 2D Video Formats

Name	Example	Definition	Candidate Codecs	Corresponding Section
Stereoscopic 3D and extensions		A Stereoscopic View is defined as the perception of depth created by the brain's ability to fuse two slightly different images from each eye, based on the parallax difference between them.	Frame-packing and HEVC MV-HEVC ...	4.3.2
Multi-view Plus Depth		Multi-view video is a frame-based representation format whereby each frame of the video represents a still that can be viewed from any perspective within a viewing space that is informed by the provided camera positions. The representation optionally supports depth maps of same resolution.	MV-HEVC MIV ...	4.3.4
Dense Dynamic Point Clouds		A volumetric representation using 3D points with spatial coordinates and attributes (e.g., color, reflectance). Contains high-density point sets (>500K points/frame) enabling detailed, closed-surface rendering.	V-PCC G-PCC ...	4.3.3
Dynamic Mesh		A dynamic mesh is an object that represents a collection of vertices, edges and triangular faces (organized in polygons) defining the object's geometry that can be modified procedurally.	Draco V-DMC ...	4.3.5
Light Fields		A light field, or lightfield, is a vector function that describes the amount of light flowing in every direction through every point in a space	LVC ...	4.3.6.2
NeRF		NeRF is the implicit representation of a 3D scene or object using a fully-connected (non-convolutional) deep network.	Under study	4.3.6.1
3D Gaussian Splattings		3D Gaussian Splatting (3DGS), also referred as Gaussian Splatting Radiance Field, is an explicit radiance field based 3D representation that represents 3D scene or objects using a large number of discrete 3D anisotropic balls or particles, each defined by its spatial mean μ and covariance matrix Σ .	Under study	4.3.6.3

4.2 Reference Model for Beyond 2D Video

4.2.1 Overview

In contrast to well-established 2D-based video formats and work flows, for beyond 2D video a variety of emerging formats and reference workflows are under discussion. This aspect makes it more difficult to harmonize specific interop points and formats, also taking into account new developments in the industry and in research. In addition, without systematic and explicit identification of format interop points, beyond 2D scenarios or workflows may look overly complex.

However, basing beyond 2D workflows and scenarios on 2D reference workflows and formats, as for example evaluated in TR 26.955 [5] and extending existing workflows seems to be promising way forward. However, when comparing for example to TR 26.955 [5] for 2D formats or even omnidirectional video formats as defined in TS 26.118

[6], additional aspects may need to be considered for beyond 2D video. To help the situation, a generic reference model for beyond 2D video content is introduced in this sub-clause. This systematic and accurate identification of interoperability points and subcomponents for Beyond 2D video with a high level of abstraction covers the majority of use cases and scenarios.

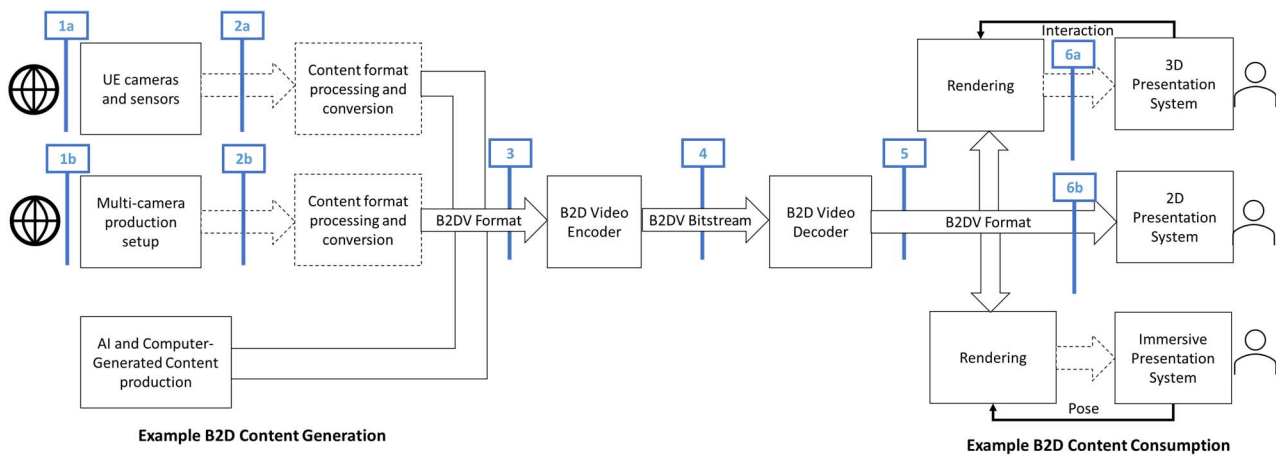


Figure 4.2.1-1 Beyond 2D Video end-to-end reference model

Figure 4.2.1-1 illustrates a generic beyond 2D Video end-to-end reference model. For example, it considers three methods of creation of source content. The first apply a naturalistic way to capture sources indicated in reference point (1) and includes for example variants of UE-based cameras and sensor (1a) or a multi-camera production setup (1b). The third option is based on authoring using computer graphics interfacing technologies or other media production technologies. These may sometimes be combined and, possibly with slight variations, these options cover the majority of media production cases.

The capture of content using cameras, for example light fields using sensors is generalized including traditional *passive sensors*, cameras, camera arrays, or plenoptic cameras. For simplicity we also include *active sensors* LiDaR, Time of Flight in this category. These active sensors also transmit a signal before capturing the reflections. Depending on the setup, the collected data may be quite different depending on the capturing system, expressed in reference point (2) with variants (2a) and (2b).

In the general case, some processing based on the captured data would happen to generate a well defined B2DV format, possibly referred to as *sensed data converter*. This step is mainly about converting the multiple digital image formats plus metadata to a well-defined beyond 2D representation or format, referenced with reference identifier (3).

For typical 5G workflows, a compressed digital representation of the B2DV is needed for efficient transmission. The *B2DV Encoder* executes a processing step that will result in the compressed Beyond 2D video bitstream that includes a digitally compressed lossy version of the B2DV format and optional metadata, referred to as reference point (4). The B2DV bitstream is typically delivered through content delivery protocols and 5G radio systems, not shown in Figure 4.2.1-1.

The *B2DV decoder* decodes the B2DV video bitstream and recovers a B2DV format, presented in reference point (5). The recovered signal is forwarded to the rendering and display system. In some cases, one viewport of the B2DV format may for example be displayed directly in a 2D Presentation System. In a 3D Presentation System, interacting with the rendering component may allow to generate different views on the content. In an immersive presentation system, pose information may be used to render the views of the content. The format generated by the renderer for the presentation system, indicated in reference point (6), is implementation specific as shown in Figure 4.2.1-1.

Generally, beyond 2D video performance measurement should typically be between interop points (3) and (5) based on the B2DV formats. The last block in the diagram includes the user interactions. Some B2DV scenarios may involve some types of user interactions, such as changing the viewpoint or other interactions. These are captured in the reference diagram for completeness.

4.3 Beyond 2D Video Representation Formats

4.3.1 Introduction

As shown in Figure 4.2.1-1, beyond 2D video representation formats may originate from different production systems and have to target different rendering systems. This clause collects relevant Beyond 2D Video representation formats and provides a discussion on the relevancy of the formats. In order to assess the relevancy of the formats, for each format different aspects are collected, among others:

- Definition of the format, this is preferably backed by a specification.
- Typical applications of the format, e.g. knowledge about support of the format in workflows (tools, etc.)
- Production options of the representation format
- Rendering of the representation format
- Benefits and limitation of the format
- Supporting information
 - Typical quality criteria for evaluating the format
 - Existing test and reference sequences
 - Conversion from other formats (lossless, lossy)
 - Uncompressed data size
 - Known compression technologies
 - Extensibility of the format

4.3.2 Extensions to Stereoscopic Video Representation Formats

NOTE: Additional references need to be extracted in further study.

4.3.2.1 Definition

Stereoscopic video presents one image to the user's left eye and another image (typically correlated) to the user's right eye to produce the stereopsis effect, defined as "the perception of depth produced by the reception in the brain of visual stimuli from both eyes in combination; binocular vision." [12].

4.3.2.2 Stereoscopic Video format description according to TS 26.265

Stereoscopic video is defined in TS 26.265 [11]. The focus on the definition in TS 26.265 is on providing a consistent description of stereoscopic video based on professional production, for example Hollywood movies. In this case, a view is available to be presented to the left eye and another view is available to be presented simultaneously to the right eye. The presentation of both the left and right views allows for an effect known as stereopsis which can be defined as "the perception of depth produced by the reception in the brain of visual stimuli from both eyes in combination; binocular vision" For signal representations, [13] recommends that Left and Right eyes comply to regular image formats such as Rec ITU-R BT.709 or Rec ITU-T BT.2100 and any necessary 3D-specific metadata is incorporated with the data. Hence, for stereoscopic video, two synchronized video signals are available, each with identical parameters. The baseline content is described in TS 26.265 as part of the 3GPP Stereoscopic 3D TV Format in clause 4.4.3.4 as follows:

The stereoscopic 3D Cinema format uses two signals, one for the left eye and another view for the right eye as defined in Table 4.4.2-1. The components for each eye closely follow the specifications of the 3GPP HDR signals, but there are some restrictions and extensions, namely:

- Frame rates include high frame rate for movies, namely 48 fps.
- the spatial resolution is restricted to 4K

An informative summary of the parameters of a 3GPP Stereoscopic 3D TV format based on the parameters defined in Table 4.4.2-1 of TS 26.265 is provided in Table 4.3.2.1-1.

Table 4.3.2.1-1 Video Signal Parameters for 3GPP Stereoscopic 3D Cinema format

Parameter	Restrictions
Picture aspect ratio	16:9
Spatial Resolution width x height	3 840 × 2 160, 1 920 × 1 080 NOTE: For 1080, typically the encoded signal has 1088 lines and cropping is applied to remove spatial samples that are not presented.
Scan Type	The source scan type of the pictures as defined in clause 7.3 of Rec. ITU-T H.273 is progressive
Chroma format indicator	The chroma format indicator is 4:2:0.
Colour primaries Transfer Characteristics Matrix Coefficients	Only the following value combinations are permitted: (1, 1, 1), (9, 16, 9), and (9, 18, 9) for SDR, HDR PQ and HDR HLG, respectively.
Bit depth	The permitted value is 10 bit.
Colour primaries	Only the value 9 as defined in clause 8.2 of Rec. ITU-T H.273 is permitted.
Transfer Characteristics	Only the value 16 (for PQ) as defined in clause 8.2 of Rec. ITU-T H.273 is permitted.
Matrix Coefficients	Only the value 9 as defined in clause 8.2 of Rec. ITU-T H.273 is permitted.
Frame rates	The permitted values are 60, 60/1.001, 48, 48/1.001, 50, 30, 30/1.001, 25, 24, 24/1.001 fps.
Frame packing	No frame packing is applied.
Projection	No projection is used.
Sample aspect ratio	The pixel aspect ratio is 1 (square pixel), i.e. only the value 1 as defined in clause 7.3 of Rec. ITU-T H.273 is permitted.
Chroma sample location type	For SDR, the location of chroma samples relative to the luma samples for progressive frames as defined in Rec. ITU-T H.273, clause 8.7 is set to 0 (Chroma samples are colocated with the luma samples at the top-left corner). For HDR PQ and HLG, the location of chroma samples relative to the luma samples for progressive frames as defined in Rec. ITU-T H.273, clause 8.7 is set to 2 (chroma samples are centered horizontally between two luma samples).
Chroma sample location type	The location of chroma samples relative to the luma samples for progressive frames as defined in Rec. ITU-T H.273, clause 8.7 is set to 2 (chroma samples are centered horizontally between two luma samples).
Range	Restricted video range is used.
Stereoscopic Video	A signal for the Left and for the Right Eye is provided whereby the signals have the identical parameters as above and are timely synchronized.

The format focuses on existing professionally generated movie content. This study will not evaluate further this format.

4.3.2.3 Extensions to Stereoscopic Video Representation formats

Extensions to the above content format result from different new use cases:

- 1) User generated stereoscopic content from modern devices that allow to capture beyond 2D video formats.
- 2) Higher-quality projected stereoscopic video that includes higher resolution and projections for professional content

The above use cases require extensions to the Stereoscopic 3D Cinema format defined in TS 26.265.

For user-generated stereoscopic content production systems as introduced in clause 4.3.2.2, extensions to the above basic signals are to be considered. The major extensions are more flexible spatial resolutions beyond 16:9, as well as additional metadata, including alpha and depth information. Offline postprocess can be used to acquire accompanying depth and such information is beneficial in the rendering to reduce parallax effects.

Stereoscopic video for user-generated content may use projections to left and right eye as follows [14]:

- rectangular, traditional 3D
- extensions with additional depth data, also referred to as video contour maps [15].
- extensions with additional alpha maps.

In addition, the detailed signal properties of the video each eye needs to be defined:

- Sample aspect ratio for each eye, defined according to the ITU-T H.273 [16], `SampleAspectRatio`. Typical parameters are 1:1 (value 1) or 4:3 (value 14).
- Picture aspect ratio for each eye. Typical parameters are 1:1 or 16:9.
- Resolutions per eye of left eye and right eye are
 - for picture aspect ratio 1:1: 1080x1080, 1440x1440, 2160x2160
 - for picture aspect ratio 16:9: 1280x720, 1440x1080 (with sample aspect ratio 4:3), 3840x2160
- Framerates for each eye are: 30 fps, 50fps, 60 fps, and 90 fps and possibly fractional variants.
- Signal characteristics
 - The video signal is YUV with 4:2:0 chroma subsampling.
 - Bit depth: 8 or 10 bits
 - Colour primaries, defined according to the ITU-T H.273 [16], `ColourPrimaries`. Typical parameters are BT-709 (value 1), and BT-2020/BT-2100 (value 9).
 - Transfer characteristics, defined according to the ITU-T H.273 [16], `TransferCharacteristics`. Typical parameters are BT-709 (value 1), BT-2020 (value 14), BT-2100 PQ (value 16) and BT-2100 HLG (value 18).
 - Matrix coefficients, defined according to the ITU-T H.273 [16], `MatrixCoefficients`. Typical parameters are BT-709 (value 1), and BT-2020/BT-2100 non-constant luminance (value 9).
 - Typical combined values are BT-709 SDR with (1,1,1) and HDR PQ with (9,16,9).

Additional metadata may be present, either on a static or per frame basis, as follows:

- hero eye: A value that indicates which eye is the primary eye when rendering in 2D.
- camera parameters: camera parameters are typically represented in a 3×4 projection matrix called the camera matrix. The extrinsic parameters define the camera pose (position and orientation) while the intrinsic parameters specify the camera image format, specifically:
 - extrinsic parameters denote the coordinate system transformations from 3D world coordinates to 3D camera coordinates. For details see: https://en.wikipedia.org/wiki/Camera_resectioning#Extrinsic_parameters

- intrinsic parameters describe a specific camera model. These parameters encompass focal length, image sensor format, and camera principal point. For details see:
https://en.wikipedia.org/wiki/Camera_resectioning#Intrinsic_parameters
- disparity adjustment:
 - horizontal disparity adjustment, a value that indicates a relative shift of the left and right images, which changes the zero-parallax plane.
- Disparity/depth map: 10bit, same resolution as source content, monochrome, can possibly be sub-sampled
- alpha maps: 8 bit, same resolution as source content
- Line time (per camera) – rolling shutter readout time, only relevant in poorer quality/reduced functionality camera pipelines typically used on HMD tracking cameras.
- Examples:
https://github.com/MPEGGroup/FileFormatConformance/tree/m62054_exintrinsic/data/file_features/under_consideration

For higher-quality projected stereoscopic video, the following are the core extensions beyond the TS 26.265 3D TV Format:

- projected video
- higher resolutions up to 8K
- additional depth and alpha data

For higher-quality projected stereoscopic video extensions content may use projections to left and right eye as follows [14]:

- spherically-projected 3D video as defined in TS 26.118 [6].
- extended with additional depth data, also referred to as video contour maps [15].

In addition, the detailed signal properties of the video each eye needs to be defined:

- Sample aspect ratio for each eye, defined according to the ITU-T H.273 [16], `SampleAspectRatio`. Typical parameters are 1:1 (value 1) or 4:3 (value 14).
- Picture aspect ratio for each eye. Typical parameters are 1:1 or 16:9.
- Resolutions per eye of left eye and right eye are
 - for picture aspect ratio 1:1: 2160x2160, 4320x4320
 - for picture aspect ratio 16:9: 3840x2160, 7680x4320

NOTE: 8K resolution is supported in TS 26.118 [6], and also supported in terms of decoding on modern mobile systems-on-chip. Whether 8K is supported in a full end-to-end workflow is application dependent, but with appropriate capability negotiation, a suitable resolution can be determined.

- Framerates for each eye are: 24 fps, 30 fps, 48fps, 50fps, 60 fps, 90 fps, 120 fps, 144 fps and possibly fractional variants.

NOTE: 120 and 144 fps are supported in terms of decoding on modern mobile systems-on-chip. Whether such high-frame rates supported in a full end-to-end workflow is application dependent, but with appropriate capability negotiation, a suitable resolution can be determined.

- Signal characteristics
 - The video signal is YUV with 4:2:0 chroma subsampling.
 - Bit depth: 10 bits

- Colour primaries, defined according to the ITU-T H.273 [16], ColourPrimaries being BT-2100 (value 9).
- Transfer characteristics, defined according to the ITU-T H.273 [16], TransferCharacteristics being BT-2100 PQ (value 16).
- Matrix coefficients, defined according to the ITU-T H.273 [16], MatrixCoefficients being BT-2100 non-constant luminance (value 9).
- The core presentation format is HDR PQ with (9,16,9).
- Projection parameters:
 - Projection: fisheye, equirectangular
 - Field-of-view and restricted coverage, typically 180 degree.

NOTE: The parameters may be aligned with TS 26.118 [6]

Additional metadata may be present, either on a static or per frame basis, as follows:

- hero eye: A value that indicates which eye is the primary eye when rendering in 2D.
- disparity adjustment:
 - horizontal disparity adjustment, a value that indicates a relative shift of the left and right images, which changes the zero-parallax plane.
- Disparity/depth map: 10bit, same resolution as source content, monochrome, can possibly be sub-sampled.

4.3.2.4 Production and Capturing Systems

The formats as defined in clause 4.3.2.1 may be captured at least with a reduced set of parameters by mobile devices and Head Mounted Displays (HMD) – for more details refer to the following information:

- <https://techcrunch.com/2023/12/11/apple-releases-spatial-video-recording-on-iphone-15-pro/>
 - Spatial Video with 1080p at 30fps
- <https://9to5mac.com/2024/01/04/will-the-iphone-16-be-able-to-record-4k-spatial-video/>
 - Spatial Video with 4K is expected to be available
- <https://appleinsider.com/articles/24/03/06/capturing-spatial-video-apple-vision-pro-vs-iphone-15-pro>
 - The spatial video captured is in a square 1:1 format at 2200 pixels by 2200 pixels. It is a near-perfect recreation of the passthrough viewed by the user.
 - Once stereo is captured on supporting phones, offline postprocess can be used to acquire accompanying depth (using for example Depth-Anything <https://github.com/DepthAnything/Depth-Anything-V2/tree/main> and [ZoeDepth](https://github.com/isl-org/ZoeDepth) <https://github.com/isl-org/ZoeDepth> or similar).
- Meta Quest™ can record spatial video: <https://360rumors.com/quest-3-3d-videos/>
 - After recording, the video or photo is captured in side-by-side format, with a square aspect ratio. Photos will also be side-by-side but they are stretched vertically, and need to be edited to fix that.
- <https://deovr.com/blog/84-record-vr-footage-on-the-meta-quest-3>
 - The Meta Quest 3™ features two cameras that deliver full-color passthrough, allowing users to record content in 4K (2k per eye), using the Meta Quest Developer HUB (<https://developer.oculus.com/documentation/unity/ts-odh>).
 - The Quest 3's passthrough cameras record footage that is flat 120-100 (possibly 90) degrees.

NOTE: In TV productions it was known that there were issues with visual fatigue, nausea due to bad content production. Guidelines that professional producers can take into account have been provided which minimize these effects. Indications whether this also is an issue for user generated content is for further study.

Beyond user-generated stereoscopic content, an ecosystem is developing around this format including movie production, documentaries and live sports. Examples are mentioned here:

- <https://www.apple.com/newsroom/2024/02/2024-mls-season-kicks-off-today-exclusively-on-mls-season-pass-on-apple-tv/>
- <https://www.apple.com/newsroom/2024/01/apple-previews-new-entertainment-experiences-launching-with-apple-vision-pro/>
- <https://www.macrumors.com/2024/01/08/vision-pro-movies-games/>

Latest information on content production can for example also be found here:

<https://www.provideocoalition.com/creating-stereoscopic-video-for-the-apple-vision-pro/>.

4.3.2.5 Rendering and Display Systems

Stereoscopic video with the above parameters can be viewed on different rendering and display systems, including

- Backward-compatible to 2D (just view one eye), hence can be viewed on regular phones. The stereoscopic effect is lost in this case.
- Apple Vision Pro TM
- Meta Quest TM: <https://techcrunch.com/2024/02/01/meta-quest-adds-support-for-apples-spatial-video-ahead-of-vision-pro-launch/>

In addition, OpenXR and WebXR define APIs to render stereoscopic video with additional metadata.

- OpenXR APIs exist
- WebXR APIs exist

For rendering multi-view stereo video, including 3D reconstruction, refer to [17]. It is shown, how additional metadata as defined in clause 4.3.2.1 can be used to improve rendering.

4.3.2.6 Supporting Information

The baseline video can be encoded using HEVC-based encoding tools:

- framepacking (see for example TS 26.118 [6])
- MV-HEVC (see TR 26.966 [10] and TS 26.265 [11])

The content can be delivered using regular ISO BMFF based distribution, including streaming with DASH/HLS/CMAF.

Uncompressed data rate can be computed as $2 \times \text{height} \times \text{width} \times (1.5 + \text{depthflag} + \text{alphaflag}) \times \text{framerate} \times \text{bitdepth}$. Some examples are provided in Table 4.3.2.6-1.

Table 4.3.2.6-1 Uncompressed data rate examples

Signal	Data rate
Stereoscopic 3D TV Format HD	$2 \times 1080 \times 1920 \times (1.5 + 0 + 0) \times 24 \times 10 = 1.39 \text{ Gbit/s}$
Stereoscopic 3D TV Format UHD	$2 \times 2160 \times 3840 \times (1.5 + 0 + 0) \times 24 \times 10 = 5.56 \text{ Gbit/s}$
User Generated Stereoscopic Content HD	$2 \times 1080 \times 1080 \times (1.5 + 0 + 0) \times 30 \times 8 = 800 \text{ Mbit/s}$

User Generated Stereoscopic Content UHD with depth	$2 \times 2160 \times 2160 \times (1.5 + 1 + 0) \times 30 \times 10 = 6.52 \text{ Gbit/s}$
Higher-quality projected stereoscopic video at 8K with alpha and depth	$2 \times 4320 \times 7680 \times (1.5 + 1 + 1) \times 60 \times 10 = 129.77 \text{ Gbit/s}$

Typical quality criteria for evaluation the stereoscopic video is define in clause 7.2.5, and the test and reference sequences are documented in Annex C.3.

4.3.2.7 Benefits and Limitations

4.3.2.7.1 Benefits

The extended stereoscopic video format has the following benefits:

- Simplicity: The technology is supported by existing content production workflows
- Device Support: The technology is supported by emerging devices on the market
- In device decoding and rendering: The technology generally allows that decoding and rendering can be done in the device, which makes it robust against impaired or lossy network connections.
- Content Industry starts to embrace the format, for details see clause 4.3.2.2
- The format is extensible to add additional metadata, for details see clause 4.3.2.1
- User-generated content production workflows exist.
- Backward-compatible rendering. The content can be rendered on 2D displays.
- Very good B2D user experiences have been reported, when the content is properly produced and suitable devices for playback and rendering are used [14].

4.3.2.7.2 Limitations

The format is primarily used to support lean-back and seated experiences, typically head movements with 3DOF and 3DOF+ can be supported, but may be extended in the future to address additional degrees freedom.

NOTE: More Benefits and limitations is for further study.

4.3.3 Dense Dynamic Point Cloud representation format

There are many applications for point clouds such as representing highly accurate maps of landscapes, buildings, infrastructure, etc... but the format is also used to represent people, animals, objects and scenes composed from these. More precisely, for representing people and objects dense dynamic point clouds are in focus.

4.3.3.1 Definition

A point cloud frame is defined as set of (x,y,z) coordinates, where x,y,z have finite precision and dynamic range, depending on the data type that is used for representing the coordinates. Each (x,y,z) can have multiple attributes associated to it (a1 ,a2, a3 ...), where the attributes may correspond to color, reflectance, transparency, normals or other properties of the object/scene that would be associated with a point. Colour is typically represented as RGB and a normal is a normal to a point which can be used by the renderer for handling lighting. Typically, each point in a point cloud frame has the same number of attributes attached to it. Dynamic point clouds consist of several consecutive point cloud frames with the same coordinate system, precisions and attributes. The number of points typically changes from one frame to the other and there is no relation between a point of one frame to the other frame. A dense point cloud contains a high density of points with close neighbors (typically more than 500.000 points per frame for a person or object), where a renderer is able to produce a closed surface allowing for a highly detailed representation.

A simple and often used file format for point clouds is the Polygon File Format (PLY) that has been developed by Greg Turk at Stanford University in 1994 [19]. Other formats, like the Object File Format (OBJ) can also be used to represent point clouds.

MPEG has defined in 2016 several use cases for point cloud compression, including *Real-time 3D Immersive Telepresence*, *Content AR/VR viewing with Interactive Parallax* and *3D Free viewpoint Sport Replays Broadcasting* [VOL-XX]. The typical characteristics of the point clouds in these use cases are summarized in Table. 4.3.3.1-1

Table 4.3.3.1-1 Typical Characteristics of Point Clouds in MPEG-Defined Use Cases

Use Case	Number of Points	Color Representation	Additional Properties
Real-time 3D Immersive Telepresence	To represent a reconstructed human: Between 100,000 and 10,000,000 points per frame	8-10 bits per color component	Normals and/or material properties for shader rendering
Content AR/VR Viewing with Interactive Parallax	To represent closeby objects in the scene: Between 100,000 and 10,000,000 points per frame	8-10 bits per color component	Global parameters defining the spatial constraints of the rendering viewport
3D Free Viewpoint Sport Replays Broadcasting	100,000 – 100,000,000 points per frame	8-12 bits per color component	Can contain multiple clusters/groups of points (different players)

4.3.3.2 Production and Capturing Systems

Professional capturing of volumetric video is typically done with a rig of synchronized cameras aligned around the asset(s) to be captured. Depending on the rig, there can be one or more layers of cameras at different height positions, with each layer consisting of up to 60 cameras. Cameras can be equipped with depth sensors. Hardware such as cameras and depth sensors are typically off the shelf equipment, but the assembly in the rig is vendor dependent and proprietary.

The various camera and depth sensor signals are fed into a production pipeline that produces the asset. Production includes stitching the various signals, filling holes, correcting occlusions, etc. Persons or physical objects (e.g., a ball or an instrument) can be combined in an asset or separate assets can be used for each person or object. The representation format of a produced asset is typically a dense dynamic point cloud or a dynamic mesh.

The Volumetric Format Association (VFA) [20] aims to “Drive the development of volumetric video as the next revolution for content creation, editing 3D content, distribution of 3D content and creating entirely new ways to tell stories and communicate with each other”. One result of their work is an end-to-end workflow consisting of Volumetric Capturing, Volumetric Processing, Volumetric Encoding and Decode/Render. The workflow can be downloaded from their website in [PDF](https://www.volumetricformat.org/_files/ugd/f2416f_3e1aeca4db234afcae9a8c15ea4f610a.pdf) (https://www.volumetricformat.org/_files/ugd/f2416f_3e1aeca4db234afcae9a8c15ea4f610a.pdf) format. Volumetric Capturing is in line with our description above. Volumetric Processing shows the dynamic point cloud representation format as a central element. First a raw point cloud is created, and which is further processed (e.g. fill holes) and converted to the produced asset. Representation formats for the produced assets is either a dynamic point cloud (in the workflow named as a patch-based format) or a dynamic mesh.

The Volumetric Encoding step includes both options, point cloud and mesh. Once streamed and received on a device, the Decode/Render step includes rendering the mesh, the point-cloud as is or generating mesh or voxels prior to rendering.

4.3.3.3 Rendering and Display Systems

The dense dynamic point cloud representation format can be rendered to 2D displays such as in mobile phones, tablets, TV sets but also to HMDs or other 3D type displays.

The visual viewing quality of the point cloud format depends heavily on how voxels are rendered. Just reconstructing voxels in 3D space may bring a limited viewing experience and holes/cracks may become visible. To show the impact of rendering two renderers are investigated:

- MPEG renderer: Each voxel is replaced by a cube of a configurable fixed size. This renderer is deliberately simple for studying the pure impact of compression.
- Representative renderer: Each voxel is replaced by a splat of a size that depends on the viewing distance and some blending is implemented to avoid flickering of points. There are no sophisticated techniques such as lighting or use of normals integrated. It represents a minimum of what a device manufacturer would do to prevent holes or cracks to preserve a good subjective experience. It is not state-of-the-art or most sophisticated renderer possible.

In the following we give an example of the impact of the renderer on the head of the sequence Thomas with Vox 10 conversion:



Figure 4.3.3.3-1 Vox 10 MPEG renderer

Figure 4.3.3.3-2 Vox 10 Representative renderer

Both snapshots are rendered from the same Vox 10 sequence. In Figure 4.3.3.3-1 (content courtesy by Volucap [136]), we see far more cracks and holes and the borderline of the sequence is less smooth. However, the eyebrows look a bit sharper in Figure 4.3.3.3-2 (content courtesy by Volucap [136]), a high-end industry renderer may do better than the renderers illustrated here.

When evaluating or comparing the point cloud representation format it is essential to select a renderer that is representative of a minimum of what the industry would implement, as holes and cracks in images would influence evaluations negatively.

More sophisticated renderers in products could fill better potential holes, recreate detail and apply lighting depending on the scene. The point cloud representation format supports normals which are useful for lighting the scene. When rendering a point cloud sequence in a scene, correct lighting including shadows and colour alignment can greatly impact the realism of the resulting experience.

POINTS_GL is the simplest OpenGL[21] primitive type used for rendering (lines and polygons are others that are also commonly used) and a point cloud can be interpreted as a vertex stream that represents points (after ordering of the points). Therefore, a point cloud can be rendered in an extremely straightforward way using native OpenGL vertex shaders. The supported rendering in the standard OpenGL specified by the Khronos consortium implies that point clouds can be rendered on devices that support OpenGL which is rather common today. OpenGL vertex shader renders points size larger than zero, this can be set GL_PROGRAM_POINT_SIZE as a configuration of the rendering.

Specific optimizations for rendering are device manufacturer dependent.

4.3.3.4 Support Information

4.3.3.4.1 Test and reference sequences

The test and reference sequences for dense dynamic point clouds are documented in Annex C.2.

4.3.3.4.2 Uncompressed data size

The uncompressed data size of a point cloud frame depends on the number of points and the number of attributes. The following table gives data size examples and raw bitrates for the sequence Thomas.

Table 4.3.3.4.2-1 Uncompressed data size and bitrate

Sequence	Quantization	#frames	#points	mean frame size (bytes)	bitrate (mbps)
Thomas	Vox10	32	19012250	4010396	979.10
Thomas	Vox11	32	76336020	16996692	4149.58
Thomas	Vox12	32	305897397	71694702	17503.59

4.3.3.4.3 Known compression technology

Visual volumetric video-based coding (V3C) and video-based point cloud compression (V-PCC) [22]

Geometry-based point cloud compression (G-PCC) [23]

Draco [24]

4.3.3.4.4 Conversion from other formats

Point clouds can be obtained by sampling from surface-based formats such as meshes. Such transformation is lossy. There are different sampling methods (e.g. methods based on face sampling, on texture map sampling, on ray casting from a grid, etc.) and it's up to the content provider to select the appropriate sampling method depending on the content and creative intent.

4.3.3.4.5 Typical quality criteria

The visual quality of a point cloud depends on the number of points (density) in the point cloud. For attributes colour is mandatory and there may be reflectance, transparency and normal. Colour is typically in RGB with each in 8 bits. Reflectance, transparency and normal can be used by the renderer when the point cloud is rendered in a scene.

Point clouds of around 1M points/frame allow to watch from a wider distance (e.g. from 3m*) and 2M points/frame allow to get closer (e.g. to around 1.5m distance) at good quality for the target scenario. Emotional facial expressions and buttons and tissue structure of cloths is visible. More points per frame improve the details, but this may not be required for the target scenario. But if a scenario would require it, a professional volumetric video production system is able to capture details from e.g. skin or finer details of tissue and it can be represented with the point cloud representation format.

* A typical demonstration scenario would be to use e.g., a smartphone or tablet running a volumetric video application showing a real person of e.g., 3m distance on the screen captured by the camera and rendering at the same time a second person rendered from a point cloud next to the first person.

Other scenarios may require the representation of the full detail of a person and the number of required points can be approximated as follows:

Assumptions:

- The visual resolution of the human eye is 1/60 of a degree
- Average human body surface is about 1.9m²

- For simplification the body surface is approximated as a square

Number of points = $1.9/((\tan 1/60 * d)^2)$, where d is the viewing distance from the person.

This leads to the following number of points:

- 1.5m distance: 10 M pixels
- 3m distance: 2.5 M pixels

4.3.3.5 Benefits and Limitations

4.3.3.5.1 Benefits

Point cloud representation is simple in structure and representation, has high accuracy and resolution, is faithful to original data, and is easy to acquire from sensors or cameras. Point cloud generation needs less pre-processing as there is no need for surface reconstruction, if sensor data is not so noisy.

A point cloud can be rendered in an extremely straightforward way using native OpenGL vertex shaders.

4.3.3.5.2 Limitations

Point-cloud data does not include information on surfaces and is harder to edit or transform.

4.3.4 Multi-view video Representation Format

4.3.4.1 Definition

Multi-view video is a frame-based immersive experience whereby each frame of the video represents a still that can be viewed from any perspective within a viewing space that is informed by the provided camera positions. The viewer can interact with the content by seamlessly moving and reorienting a virtual viewport. This serves two goals from the user perspective: it is possible to look around objects, and it is possible to freely choose a viewpoint. The first goal is best achieved by having a dense group of cameras around a scene with nearby subjects. This creates a sense of immersion. The second goal is best achieved by having a sparse group of cameras in an arc around a scene. This creates the free-viewpoint functionality which is arguably less immersive, but enables the viewer to observe an action in more detail, i.e. by "being the director".

Note that in some contexts like in JCT-3V a narrower definition of multi-view was used whereby the cameras are expected to be in a 1D linear or coplanar arrangement [35]. For this representation there is no such restriction.

Recently, multi-view and multi-view plus depth video representations have been used as an in-between step to create point clouds, meshes, light field and radiance field approximations including NeRF (clause 4.3.6.1) and 3DGS (clause 4.3.6.3). Note that when depth maps are not directly available from range-sensing cameras, they can be estimated using open source or commercial tools. The same holds for the estimation of intrinsic and extrinsic camera parameters. Hence, the multi-view plus depth representation is widely recognized and understood.

The multi-view video representation consists of multiple frames of multiple synchronized physical or virtual camera views. Each camera view is represented by a colour image (YCbCr), camera intrinsics and camera extrinsics. The combination of video and metadata allows for novel view synthesis (6DoF rendering).

A typical spatial resolution for each of the views is 1920×1080 (FullHD). For this representation in this study, we expect resolutions in a range around this number. A typical number of views is 2-4 for real-time capture with range-sensing cameras like the Azure KinectTM, and typically 10-20 for offline capture with industrial or professional cameras. Typically, the frame rate is 25 or 30 Hz, and capture beyond 60 Hz is not expected for the coming years.

Optionally there is also a depth image of equal resolution. It is possible to have multi-view content for which some or all views lack depth information. This choice originates from the production and capturing system and thus it is the same for all frames of a view. The depth map image, if present, may also indicate that individual samples are missing. This indication can be used for range-sensing cameras that cannot sense depth in certain situations like object edges, non-reflecting and specular reflecting scene elements. It can also be useful in a production system to remove parts of an image that are not wanted (e.g. revealing camera rigs) or are also present in other views (scene background).

If a view has a depth map, then it must have corresponding depth quantization parameters: quantization type (normalized disparity or linear depth), nearest depth in scene units, furthest depth in scene units, and indication of invalid values. Normalized disparity [m^{-1}] is more commonly used when depth is estimated because it places the code points in a way that correlates with the amount of parallax, and it allows for far away scene elements like the horizon or the star field. Linear depth [m] is commonly used with range-sensing cameras (ToF, LiDAR, etc.) because they often have a limited depth range with equal depth resolution for that entire range.

The camera intrinsics are a model of the projection of points in space in the reference system of the camera to the image sensor (projection plane). Typical parameters include projection type (perspective, fisheye, etc.), and projection-type specific parameters, such as principal point and focal length for perspective projection. Optionally lens distortion parameters may be provided if the camera images are not already corrected for that.

The camera extrinsics model the translation and rotation of a camera in space with respect to the reference system of the scene.

The source format has at least two views. It is expected that most or all test data will have perspective projection (PSP), but test data with equirectangular projection (ERP) may be included.

While this representation allows for 6DoF rendering, it depends on the position and field of view of the cameras, if such a rendering has an acceptable quality. Preferably, the virtual viewpoints are within a viewing space that can be provided as metadata or implicitly derived from the parameters of the set of source views.

Each view has the following video components and metadata:

Table 4.3.4.1-1 Multi-view video component and metadata

Component	Texture (mandatory)	Depth (optional)
Spatial resolution	At least 960×540 At most 3840×2160	The same as the texture component
Chroma format	YCbCr	Luma only or YCbCr with chroma planes set to neutral gray
Chroma subsampling	4:2:0	4:0:0 or 4:2:0 with chroma planes set to neutral gray
Pixel aspect ratio	1:1	1:1
Frame rate	30, 50, 60	The same as the texture component
Colour space format	ITU-R BT.709 or ITU-R BT.2100	Undefined
Transfer characteristics	Limited range or full range with transfer characteristics matching to the colour space format. Mastering characteristics such as MDCV (master display colour volume) and CLLI (content light level information) SEI (supplementary enhancement information) messages defined in TS 26.116 Section 4.5.5.7 will be considered.	Full range, linear transfer
Bit depth	Either 8 bits or 10 bits for all channels	At least 8 bits At most 16 bits
Metadata	Camera intrinsics: Projection type (Perspective, ERP)	Depth quantization parameters: - Quantization type: - either: normalized disparity

	<ul style="list-style-type: none"> - Projection type (Perspective, ERP) - For perspective projection: <ul style="list-style-type: none"> - Focal length [px] - Principal point [px × px] - For equirectangular projection: <ul style="list-style-type: none"> - Latitudinal angle range [rad × rad] - Longitudinal angle range [rad × rad] - Lens distortion parameters (optional) <p>Camera extrinsics:</p> <ul style="list-style-type: none"> - Camera position (x, y, z) [m] - Camera orientation as normalized quaternion ($q = iq_x + jq_y + kq_z + q_w$) 	<ul style="list-style-type: none"> - or: linear depth - Near depth [m] - Far depth [m] - Has invalid pixels flag
--	---	--

4.3.4.2 Production and Capturing Systems

Multi-view video and multi-view + depth are well-known formats that have many public tools including OpenCV [26], COLMAP [27], AliceVision [28] and OpenMVG [29]. Also, MPEG has published tools for camera calibration and depth estimation [30].

There are four typical workflows for multi-view (+ depth):

- Use color cameras to capture multi-view and estimate depth with multi-view consistency.
- Use range-sensing cameras to capture multi-view + depth and refine depth with multi-view consistency.
- Use AI or CG pipelines to raytrace views.
- Combinations of the above.

The beyond 2D video is captured and processed using multiple cameras. Zero or more of those cameras may be range-sensing cameras, and more than one of the cameras has color sensors. In the case of two or more cameras that are not rigidly connected, camera extrinsics are online calibrated. Depth estimation is performed to associate a full depth map with each of the camera views, thus resulting in a multi-view + depth representation.

Additional steps such as object instance segmentation and foreground/background separation may be performed to reduce the sample rate of the representation. This would result in a multi-view + depth + transparency/occupancy representation. All processing may be offline or with a delay of a few seconds.

Figure 4.3.4.2-1 provides an example processing flow with the following operations:

- Multi Camera Capture: capture of images from multiple cameras
- Intrinsic Calibration: estimation of principal point, focal length and distortion parameters
- Extrinsic Calibration: estimation of camera orientation and translation (e.g. using COLMAP)
- Scene Calibration: estimation of static ground plane geometry and background geometry
- Undistort Images: all images undistorted to one and the same reference intrinsics

- Object Instance Segmentation: determine segments for known objects such as ‘person’/’ball’
- Depth Estimation: determine a dense depth map for each view
- Depth Segmentation: determine sub-instance depth segments consisting of smooth surfaces

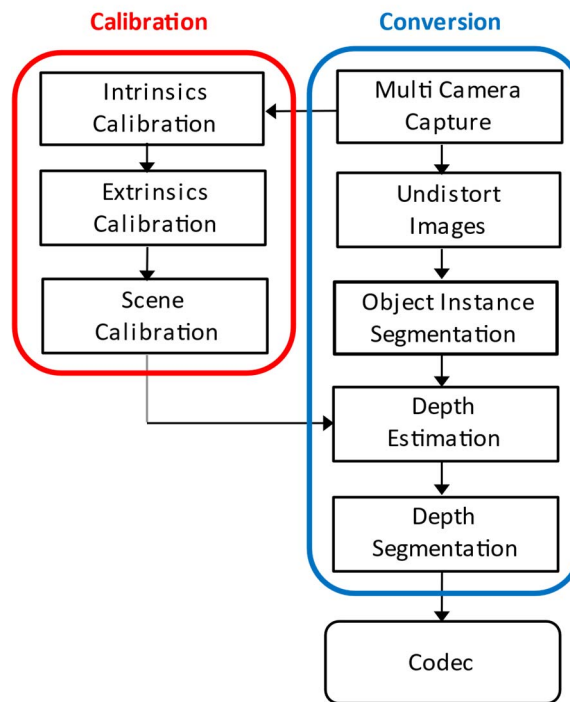


Figure 4.3.4.2-1: Example processing flow

4.3.4.3 Rendering and Display Systems

Some examples of open source rendering implementations for the multi-view representation are the Reference view synthesizer [31], Test model for MPEG immersive video [32], and OpenDIBR [33]. More implementations exist.

Real-time rendering is typically performed on a GPU without dedicated hardware.

Rendering can be on:

- a device for 2D presentation (fixed viewpoint) such as a phone,
- a device for 3D presentation (multiple viewpoints) such as an autostereoscopic display,
- a device for 6DoF presentation (dynamic viewports) such as an HMD or an autostereoscopic display with eye tracking.

When a viewing space is used, then:

- What is rendered is one or two viewports with perspective projection and with 6 degrees of freedom (3-D position and 3-D orientation).
- The pose of the viewport is within a viewing space that can be signaled or implicitly determined from a decoded frame. A viewing space can limit both position, orientation or both in combination. For instance, it is generally not intended for a viewport to intersect with scene elements.
- When a viewport is rendered that is outside of the viewing space, then the renderer has to perform a mitigation to avoid a viewing experience that is not intended by the content provider.

4.3.4.4 Supporting Information

4.3.4.4.1 Camera placement

For range-sensing cameras a minimum requirement is that scene elements are present in at least one view frustum (seen by at least one camera). This implies that view frustums of adjacent cameras are overlapping at and beyond the nearest object distance.

With multi-view depth estimation, the minimum requirement is that scene elements are present in at least two view frustums (seen by at least two cameras). The additional overlap is needed for stereo correspondence checks.

- Given the above, cameras do not have to be placed on a line, plane or any specific geometry.

4.3.4.4.2 Spatial resolution

The 3-D spatial resolution relates to video resolution and depth map bit depth. Because perspective projection and normalized disparity are most common, equations are provided for this case only.

Perspective unprojection maps a sample position $\mathbf{x}_{\text{image}}$ to a scene position in respect to the camera $\mathbf{x}_{\text{camera}}$:

$$\mathbf{x}_{\text{camera}} = \begin{bmatrix} x_{\text{camera}} \\ y_{\text{camera}} \\ z_{\text{camera}} \end{bmatrix} = \frac{r}{f} \begin{bmatrix} x_{\text{image}} - p_x \\ y_{\text{image}} - p_y \\ f \end{bmatrix},$$

with principal point \mathbf{p} and focal length f , both in pixel units. The coordinate system is only a convenient example.

Normalized disparity expansion maps sample value i to depth range value r in scene units, e.g. meters:

$$r = \frac{1}{\frac{1}{r_{\text{far}}} + \left(\frac{1}{r_{\text{near}}} - \frac{1}{r_{\text{far}}}\right) \frac{i}{i_{\text{max}}}} = \frac{r_{\text{far}} r_{\text{near}}}{r_{\text{near}} + (r_{\text{far}} - r_{\text{near}}) \frac{i}{i_{\text{max}}}}$$

One may check that this maps $i \in \{0, 1 \dots i_{\text{max}}\}$ to $r \in [r_{\text{near}}, r_{\text{far}}]$. Note that nearby objects appear brighter when viewing the depth map directly (as if using a flashlight in a dark room).

In-plane spatial resolution refers to the ability of an imaging system to distinguish between two adjacent points within the same imaging plane. It is a measure of how close two objects can be to each other in the imaging plane while still being resolved as separate entities.

The in-plane spatial resolution can be derived from the first equation:

$$\frac{\delta x_{\text{camera}}}{\delta x_{\text{image}}} = \frac{r}{f}$$

This indicates that a) the in-plane spatial resolution depends on the focal length in pixel units, and b) the spatial resolution decreases with distance from the camera. As an example, when an object is at 1 meter distance, and the focal length is 1000 pixels, a horizontal or vertical shift of one pixel corresponds to a shift of 1 mm in 3-D space.

Out-of-plane spatial resolution refers to the ability of an imaging system to distinguish between points along the axis perpendicular to the imaging plane (typically the z-axis). It represents the system's capability to resolve depth information or separate structures at different depths.

The out-of-plane spatial resolution can be derived by combining the first and second equations:

$$\frac{\delta z_{\text{camera}}}{\delta i} = \frac{\delta r}{\delta i} = \frac{r^2}{i_{\text{max}}} \frac{r_{\text{far}} - r_{\text{near}}}{r_{\text{far}} r_{\text{near}}}$$

With r_{far} much larger than r_{near} , this approximates to:

$$\frac{\delta z_{\text{camera}}}{\delta i} \approx \frac{r^2}{i_{\text{max}} r_{\text{near}}}$$

This indicates that a) the out-of-plane spatial resolution depends mainly on the nearest object distance and the depth map bit depth, and b) the spatial resolution decreases quadratically with distance from the camera. As an example, when

an object is at $r_{\text{near}} = 1$ meter distance and bit depth is 8 bit ($i_{\text{max}} = 255$), then $\Delta z_{\text{camera}} \approx 4$ mm. When instead an object is at 10-meter distance, the step size is about 0.4 m.

4.3.4.4.2 Objective metrics

Objective evaluation on multiview video may be performed by applying 2D video objective metrics (PSNR, SSIM, VMAF, etc.) on each of the source view positions, and averaging them in the correct domain. A higher correlation with subjective evaluation may be obtained by applying immersive video metrics [34] [35] that consider that view synthesis may cause pixel shifts that have only a minor influence on subjective scores, but cause PSNR to degrade.

4.3.4.4.3 Coding and delivery options

The content can be encoded using:

- MPEG Immersive Video (MIV)
- MV-HEVC (albeit with some restrictions)

The content can be delivered using regular ISO BMFF based distribution, including streaming with DASH, or delivered in real-time using RTP-based transport.

The detailed encoding and decoding constraints and settings for multi-view video is defined in clause 7.4.4.

4.3.4.5 Benefits and Limitations

4.3.4.5.1 Benefits

The multiview video representation has the following main benefits:

- Real-time capture is feasible.
- This format is often used as an intermediate step in photogrammetry pipelines such as [28].
- The renderings have the appearance of natural video content, as opposed to computer graphics, because all optical effects are baked into the multiple views.

4.3.4.5.2 Limitations

The multiview video representation has the following limitations:

- Need to handle large number of pixels, e.g. by selection.
- For novel view synthesis, multiple views need to be blended for optimal rendering results, to handle non-Lambertian effects.
- Content production depends on the availability of good and efficient depth estimation/refinement tools. Recently, there is a strong progress in the field of computer vision and volumetric approaches specifically, which will benefit applications of this representation.

4.3.5 Dynamic Mesh Representation Format

4.3.5.1 Definition

A mesh is a structure composed of several polygons that define the boundary surface of a volumetric object. It typically includes five components: connectivity information, geometry information, mapping information, vertex attributes, and attribute maps. From MPEG, a dynamic mesh is defined as a mesh where at least one of these five components is varying in time [36]. Such change can result from prescribed motion, flow induced rigid body motion, fluid structure interaction or adaptive mesh refinement. In the industry, a dynamic mesh is an object that represents a collection of vertices, edges and triangular faces (organized in polygons) defining the object's geometry that can be modified procedurally (e.g., in an avatar's facial expression or body movement) [43].

Dynamic meshes are one of the immersive contents that are widely used in the commercial markets. For example, they can be used to represent 3D objects or digital avatar in VR/AR, digital twin city and etc. The demand for processing and visualizing such rich 3D content has led to the increasing popularity of dynamic meshes, as they are natively supported by virtually all the 3D software and graphic hardware, friendly to GPU rendering, and have a strong applicability to interactive and real-time 3D task [38].

Many different formats can be used for storing dynamic mesh representation data. For example, the PoLYgon (PLY) format is introduced in section 4.6.3.5.2 of TR 26.928 [26.928], the OBJ file format is used in section 4.3.5.4.1.2, and also the glTF format as specified by the Khronos Group [42].

MPEG has defined in 2021 several use cases for dynamic mesh compression, including *Real-time 3D Immersive Telepresence*, *Content AR/VR viewing with Interactive Parallax* and *3D Free viewpoint Sport Replays Broadcasting* [37]. The typical characteristics of the meshes in these use cases are summarized in Table. 4.3.5.1-1.

Table 4.3.5.1-1 Typical Characteristics of Meshes in MPEG-Defined Use Cases

Use Case	Triangle Count	Color Representation	Additional Properties
Real-time 3D Immersive Telepresence	To represent a reconstructed human: - 40,000 – 100,000 triangles (with color per vertex) - 10,000 – 50,000 triangles (with texture maps)	- Texture maps: 2K–8K square pixels - Color per vertex	- Normals and/or material properties for shader rendering
Content AR/VR Viewing with Interactive Parallax	To represent a reconstructed human: - 10,000 – 100,000 triangles	- Texture maps: 2K – 8K square pixels - Color per vertex	- Normals and/or material properties for shader rendering - Global parameters for spatial constraints - The meshes may be a part only of the total content transmitted
3D Free Viewpoint Sport Replays Broadcasting	To represent a reconstructed human: - 20,000 – 200,000 triangles	- Texture maps: 8 – 12 bits per color component - Color per vertex	- Multiple clusters/groups of meshes (e.g., different players)

4.3.5.2 Production and Capturing Systems

Dynamic meshes cannot be directly captured through 3D scanning devices. Instead, meshes can be generated either manually by artists or automatically through 3D generation algorithms. The current production methods include:

- **Manual Creation:** artists use 3D modeling software packages (such as Blender TM, Maya TM, and etc.) to manually create dynamic meshes. The artist-created meshes capture not only the external appearance of objects but also their intrinsic properties and construction details through mesh topology. High-quality meshes used in games and movies are almost exclusively created by artists.
- **Volumetric Capture Studio:** An array of multi-camera or multi-stereo cameras is placed around a recording space to capture a subject within that space. After the capture, a generation and production process is required to create the dynamic meshes. For further details, refer to Clause 4.6.7 of TR 26.928.
- **Converted from other formats:** A mesh can be extracted algorithmically from other beyond 2D representations, such as 3D Gaussians, neural fields, voxels and point clouds.
- **AI-generated meshes:** An emerging line of research generates 3D meshes in a data-driven fashion by learning from artist-created meshes using machine learning algorithms. For example, PolyGen (<https://polygen.io/>), MeshGPT (<https://nihalsid.github.io/mesh-gpt/>), MeshAnything (v1:<https://buaacyw.github.io/mesh-anything/>)

and v2: <https://buaacyw.github.io/meshanything-v2/>), and MeshXL (<https://meshxl.github.io/>). These methods show significant promise, particularly in terms of automating 3D asset creation. However, they are still limited by scalability, with the best method handling up to approximately 1.6K faces, and the resulting meshes often exhibit a significant quality gap compared to those crafted by artists.

4.3.5.3 Rendering and Display Systems

Dynamic meshes can be rendered directly on GPUs that are highly optimized for mesh-based rendering. The following are the rendering APIs and engines for dynamic mesh processing:

- Low-Level rendering APIs:
 - OpenGL: <https://learnopengl.com/Model-Loading/Mesh>
 - DirectX 12: <https://microsoft.github.io/DirectX-Specs/d3d/MeshShader.html>
 - Vulkan: https://docs.vulkan.org/spec/latest/chapters/VK_NV_mesh_shader/mesh.html
- Graphic Engines:
 - Unity TM: <https://docs.unity3d.com/6000.0/Documentation/ScriptReference/Mesh.MarkDynamic.html>
 - Unreal Engine TM: <https://dev.epicgames.com/documentation/en-us/unreal-engine/BlueprintAPI/DynamicMesh>
 - NVIDIA RTX / OptiX TM: <https://developer.nvidia.com/rtx/ray-tracing/optix>
- Web-Based rendering APIs:
 - WebGL: <https://www.khronos.org/webgl/>
 - WebGPU: <https://webgpu.github.io/webgpu-samples/?sample=skinnedMesh>
 - High-level APIs: high-level libraries use WebGL and WebGPU underneath to provide an easy-to-use, lightweight, and cross-browser solution for general-purpose 3D rendering. For example, Three.js (<https://threejs.org/docs/api/en/objects/Mesh.html>); Babylon.js (<https://doc.babylonjs.com/features/featuresDeepDive/mesh>).

Rendering can be on:

- a device for 2D presentation such as a phone
- a device for 3D presentation such as an autostereoscopic display, providing depth perception for dynamic meshes
- a device for 6DoF presentation such as VR/AR devices like Meta Quest TM, HTC Vive TM, and Apple Vision Pro TM support real-time rendering of dynamic content.

4.3.5.4 Supporting Information

4.3.5.4.1 Test and reference sequences

Collected candidate raw dynamic mesh sequences that are available for testing are documented in Annex C.2

4.3.5.4.2 Uncompressed data size

The uncompressed data size of dynamic meshes depends on several factors, including vertex count, attribute information, level of detail (LOD), animation/deformation data and etc. A dynamic mesh sequence may require a large amount of data since it may consist of a significant amount of information changing in time. The size of dynamic mesh sequences typically ranges from a few gigabytes to several dozen gigabytes. For example, the *basketball_player*

sequence proposed by OwlII Inc. contains around 40K triangles, with texture maps at a resolution of 2048 x 2048. With a total of 600 frames, its raw data size is about 45.2GB.

4.3.5.4.3 Known compression technologies

Existing compression technologies for dynamic meshes include:

- Google's Draco [23], a C++ compression library designed for static meshes. Dynamic meshes are typically encoded as independent frame and the temporal coherence and redundancies in dynamic meshes are not leveraged.
- Mesh compression standards such as IC, MESHGRID, and FAMC [39], previously developed by MPEG can only compress dynamic mesh sequences with constant topological information (same vertex counts and face connections). These method can't handle dynamic meshes with time-varying topology, geometry and attribute information.
- V-DMC [40], a new mesh compression standard to directly handle dynamic meshes with time varying connectivity information and optionally time varying attribute maps. The initial V-DMC test model was released by MPEG in July 2022, it is currently under Draft International Standard (DIS) status, and has been submitted to ISO with the reference ISO/IEC 23090-29. The WD1.0 of V-DMC conformance and the reference software N1047 have also been provided, including streaming with DASH, or transmitted in real-time using RTP-based transport [41].

4.3.5.4.4 Conversion from other formats

Dynamic meshes can be converted from point clouds as defined in clause 4.3.3 or voxels using software like MeshLab (<https://github.com/cnr-isti-vclab/meshlab>), CloudCompare (<https://github.com/CloudCompare/CloudCompare>), or Autodesk (<https://github.com/Autodesk>). Such transformation is lossy.

4.3.5.4.5 Typical Quality Criteria

4.3.5.4.5.1 Objective Metrics

MPEG WG7 proposes two methods for dynamic mesh evaluation: one based on the well-known D1/D2 metric used in point cloud compression (**point-based metric**), and another one based on evaluation of projected images (**image-based metric**) Annex B [36]. Both methods are implemented in the `mpeg_pcc_mmetric` software which is available on the MPEG GIT [53].

For **point-based metric**, it directly uses the raw data from the reference and distorted meshes to extract features and predict quality. It includes two steps, as shown in Figure 4.3.5.4.5.1-1. First, the input meshes are sampled to be converted into their respective point cloud representations. Second, with the sampled surface point clouds, point cloud objective metrics D1/D2, Y-PSNR, and PCQM are calculated to assess quality. According to experimental results [46], the point-based metrics (particularly PCQM_P) show high performance in dynamic mesh quality assessment and do not need rendering. However, these metrics heavily rely on mesh sampling step, which requires dense sampling to achieve accurate results. This can be computationally intensive and significantly increase processing time.

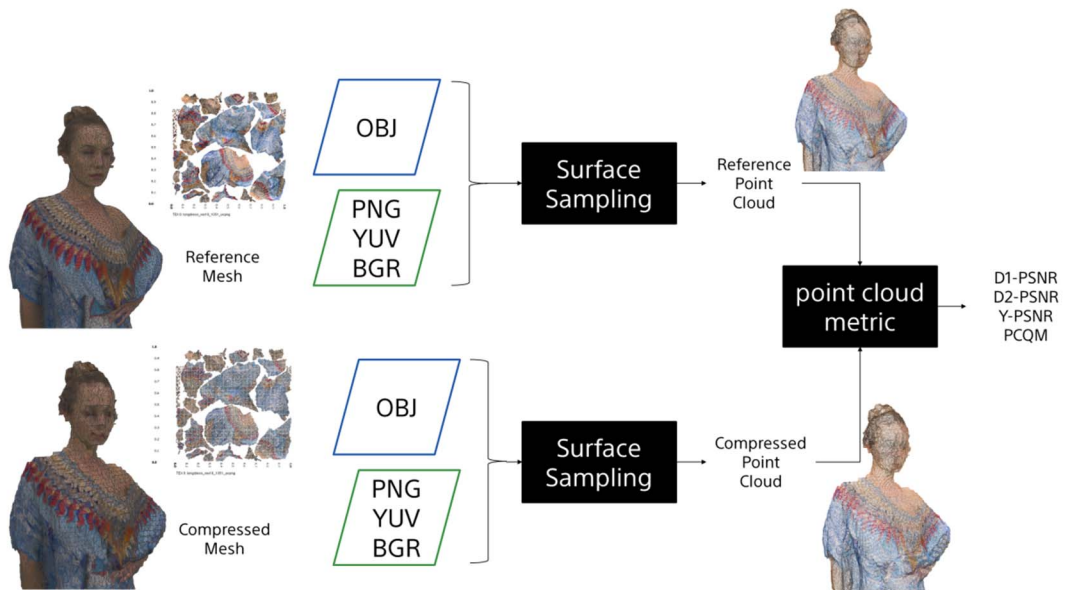


Figure 4.3.5.4.5.1-1: Point-based method for mesh evaluation

For **image-based metric** as shown in Figure 4.3.5.4.5.1-2, the reference and the distorted meshes are rendered for multiple view directions vd_i , using an orthographic projection. The images obtained from the rendering of reference and distorted models are then compared using some adapted image MSE/PSNR metrics. The results are averaged over a set of view directions for the frame and over the frames of the sequence. According to experimental results [46], projecting dynamic mesh into colored image and then applying metrics like rgb_{psnr} and yuv_{psnr} is more effective than only capturing depth information to use geo_{psnr} . In addition, increasing the number of projected images improves the stability of image-based metrics but also leads to a higher calculation complexity.

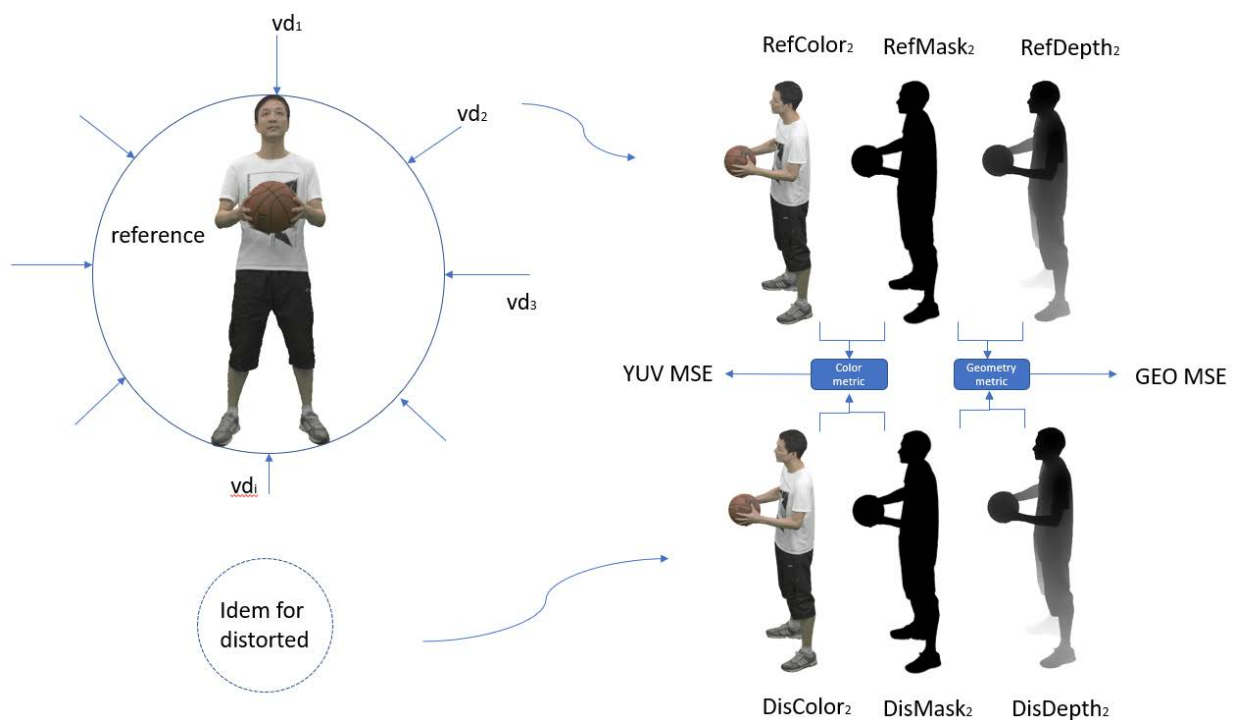


Figure 4.3.5.4.5.1-2: Image-based method for mesh evaluation

NOTE: Other objective metrics for dynamic mesh evaluation is FFS.

4.3.5.4.5.2 Subjective Evaluation

There are two prevalent methods to rendering dynamic mesh samples for subjective evaluation: **2D video-based** and **VR-based methods**.

For **video-based subjective evaluation** [47] [48], it uses a 2D monitor to display dynamic meshes with compression and surface noise distortions, and refer to ITU-R BT.500 [49] and ITU-T P.910 Recommendations [50] to conduct the subjective experiment. MPEG describes a video-based subjective evaluation in Annex D of [36]. The associated renderer is available on the MPEG Git [54].

For **VR-based subjective evaluation**, although there is a lot of academic research to explore the principles of the subjective experiment in a VR environment [51], a standardized protocol has yet to be established. This remains an ongoing effort within ITU-T SG12/Q7 P.IntVR [52].

NOTE: Other subjective methods for dynamic mesh evaluation is FFS.

4.3.2.7 Benefits and Limitations

4.3.2.7.1 Benefits

The dynamic mesh format has the following benefits:

- Good visual quality. Meshes define the object's shape and structure in a fairly realistic way, allowing for finer details and realistic shading and rendering through texture mapping, making 3D assets look photorealistic.
- Most important and widely used representation for 3D assets in the commercial market. De facto standard in the film, design and gaming industries.
- Natively supported by virtually all 3D software and graphic hardware.
- Friendly to GPU, can be used for real-time rendering.
- Backward-compatible rendering. The content can be rendered on 2D displays.

4.3.2.7.2 Limitations

A dynamic mesh sequence may require a large amount of data since it may consist of a significant amount of information changing in time. Standardized interoperable efficient compression, storage, and transmission of dynamic meshes have been specified.

4.3.6 Formats under Research

NOTE: Formats in that section will not be part of the evaluation framework of release 19, due to their maturity status, or complexity. However, it is recommended that 3GPP follows the research work on NeRF, INVR and GS and awaits stabilization in the industry to commonly agreed formats.

4.3.6.1 Neural Radiance Fields

4.3.6.1.1 Introduction

Neural Radiance Field (NeRF) is a technology at the intersection of Artificial Intelligence (AI) and 3D graphics, and has gained interest based on remarkable progress in computer vision, neural processing units and graphics processing. NeRF was an important research area over the last few years, but recently the interest in NeRF has declined and more attention is given to other formats documented in the remainder of this clause 4.3.6. The documentation reflects the state of the art at the time of writing, but the technology has reached a level of maturity.

4.3.6.1.2 Definition

NeRF is the implicit representation of a 3D scene or object using a fully-connected (non-convolutional) deep network, whose input is a single continuous 5D coordinate (spatial location (x, y, z) and viewing direction (Θ, Φ)) and whose output is the volume density (α) and view-dependent emitted radiance (r, g, b) at that spatial location [56].

The key idea behind NeRF is to represent the appearance of a scene as a function of 3D position and viewing direction, known as the radiance field. The radiance field describes how light travels through the scene and interacts with its surfaces and can be used to generate images from arbitrary viewpoints [57].

The following is an overview pipeline for NeRF:

Field representation: For each point in space the NeRF represents a view dependent radiance.

Positional encoding: The input coordinates (x, y, z, θ, ϕ) need to be encoded to a higher dimensional space prior to being input into the network.

Rendering: NeRF rely on classic volumetric rendering techniques to composite the points into a predicted color.

Sampling: NeRF use a hierarchical sampling scheme that first uses a uniform sampler and is followed by a PDF sampler.

4.3.6.1.3 Production and Capturing Systems

Mobile apps such as NeRFCapture (<https://github.com/jc211/NeRFCapture>), Spectacular AI (<https://github.com/SpectacularAI>), or Record3D (<https://record3d.app/>) are available to capture NeRFs.

A tutorial for capturing NeRFs is provided here: https://github.com/NVlabs/instant-ngp/blob/master/docs/nerf_dataset_tips.md.

The NeRFCapture app allows any iPhone™ or iPad™ to quickly collect or stream posed images to InstantNGP. If your device has a LiDAR, the depth images will be saved/streamed as well. It has two modes: Offline and Online. In Offline mode, the dataset is saved to the device and can be accessed in the Files App in the NeRFCapture folder. Online mode uses CycloneDDS to publish the posed images on the network. A Python script then collects the images and provides them to InstantNGP.

The Spectacular AI SDK and apps can be used to capture data from various devices:

- iPhones™ (with LiDAR)
- OAK-D cameras
- RealSense™ D455/D435i
- Azure Kinect DK™

The Record3D can create a dataset with an iPhone 12 Pro™ or newer (based on ARKit), a python code is needed to convert the captured data to NeRF (<https://github.com/NVlabs/instant-ngp/blob/master/scripts/record3d2nerf.py>)

The state-of-art of NeRF at the time of writing includes:

- SMERF (Streamable Memory Efficient Radiance Fields for Real-Time Large-Scene Exploration) is a view synthesis approach that achieves state-of-the-art accuracy among real-time methods on large scenes with footprints up to 300 m² at a volumetric resolution of 3.5 mm³ [57]. It enables fully 6DoF navigation within a web browser, and renders real-time on smartphones and laptops.
- Instant Neural Graphics Primitives (Instant-NGP) using multi-resolution hash encoding to split the processing into multiple chunks and using parallel processing using cuda software to effectively change run time from hours to seconds [58]. Instant-NGP is a method that uses hash-grid and a shallow MLP to accelerate training and rendering. This method reaches speedups of 1000x and train very fast (~6 min) and renders also fast ~3 FPS.
- [NerfStudio](https://docs.nerf.studio/) (<https://docs.nerf.studio/>), which is open-source and combines many radiance fields methods, and supports the storage of NeRF data in a structured format, which includes key elements as follows.

Camera intrinsics:


```
{
  "camera_model": "OPENCV_FISHEYE", // camera model type [OPENCV, OPENCV_FISHEYE]
  "fl_x": 1072.0, // focal length x
  "fl_y": 1068.0, // focal length y
  "cx": 1504.0, // principal point x
  "cy": 1000.0, // principal point y
  "w": 3008, // image width
  "h": 2000, // image height
  "k1": 0.0312, // first radial distortion parameter, used by [OPENCV, OPENCV_FISHEYE]
  "k2": 0.0051, // second radial distortion parameter, used by [OPENCV, OPENCV_FISHEYE]
  "k3": 0.0006, // third radial distortion parameter, used by [OPENCV_FISHEYE]
  "k4": 0.0001, // fourth radial distortion parameter, used by [OPENCV_FISHEYE]
  "p1": -6.47e-5, // first tangential distortion parameter, used by [OPENCV]
  "p2": -1.37e-7, // second tangential distortion parameter, used by [OPENCV]
  "frames": // ... per-frame intrinsics and extrinsics parameters
}
```

Camera extrinsics:

```
{
  // ...
  "frames": [
    {
      "file_path": "images/frame_00001.jpeg",
      "transform_matrix": [
        // [+X0 +Y0 +Z0 X]
        // [+X1 +Y1 +Z1 Y]
        // [+X2 +Y2 +Z2 Z]
        // [0.0 0.0 0.0 1]
        [1.0, 0.0, 0.0, 0.0],
        [0.0, 1.0, 0.0, 0.0],
        [0.0, 0.0, 1.0, 0.0],
        [0.0, 0.0, 0.0, 1.0]
      ]
    }
  ]
}
```

Depth images:

```
{
  "frames": [
    {
      // ...
      "depth_file_path": "depth/0001.png"
    }
  ]
}
```

Masks:

```
{
  "frames": [
    {
      // ...
      "mask_path": "masks/mask.jpeg"
    }
  ]
}
```

4.3.6.1.4 Rendering and Display Systems

NeRF heavily relies on the volumetric rendering process to obtain rendered pixels. This rendering function is differentiable, so scene representation can be optimized by minimizing the residual between synthesized and ground truth observed images. The rendering process requires sampling tens to hundreds of points along each ray and inputting them into the neural network to produce the final imaging result. Consequently, rendering a single 1080p image necessitates on the order of 108 neural network forward passes, which often takes several seconds [59].

Display System: VR HMD, mobile devices.

4.3.6.1.5 Supporting Information

- Typical quality criteria for evaluating the format
- Evaluation metrics such as PSNR (Peak Signal-to-Noise Ratio), SSIM (Structural Similarity Index), and LPIPS (LearnedPerceptual Image Patch Similarity)

- Training iteration, training time, inference speed.
- Conversion from other formats (lossless, lossy)
 - Meshes, point clouds
- Uncompressed data size

The original NeRF model has 8 fully connected layers, with a layer width of 256, and each pixel is synthesized based on 128 samplings along the ray. The standard NeRF model demands an impractical 5,600 Terabytes cache size.

- Known compression technologies:

Early research on NeRF compression is ongoing. The MPEG established the ad-hoc group called Implicit Neural Visual Representation (INVR) and is currently exploring the potential standardization of 6 Degree of Freedom (6DoF) video compression using NeRF-based technologies[59]. The following methods are applied in current research for NeRF compression and encoding:

- Parameter quantization techniques, transform coding, and entropy coding [69]
- VVC and NNC [70]
- Extensibility of the format
 - Mip-NeRF, Point-NeRF, KiloNeRF, Mega-NeRF and etc [58].

4.3.6.1.6 Benefits and Limitations

4.3.6.1.6.1 Benefits

- High-quality 3D representation: NeRF can create photo-realistic 3D reconstructions of complex scenes, including fine surface details, reflections and realistic lighting effects.
- Improved view synthesis capabilities: NeRF can synthesize novel views of a scene or object from a small number of input images, allowing rendering from any viewpoint.
- Flexibility: NeRF can handle non-rigid and dynamic scenes, adapting well to varying spatial conditions and changes over time.
- Unsupervised training: NeRF can learn to reconstruct a scene or object without explicit supervision.

4.3.6.1.6.2 Limitations

- More computationally demanding and slower to render compared to photogrammetry and 3D Gaussian Splatting.
- Not reductionistic: The entire scene is encoded in a single NeRF function, which makes it challenging to segment the scene into parts, edit individual objects within the scene, or combine different NeRF scenes into one.
- Currently, NeRF representation formats do not seem to effectively handle dynamic content within 3D scenes.

4.3.6.2 Light Fields Video

4.3.6.2.1 Definition

A light field, or lightfield, is a vector function that describes the amount of light flowing in every direction through every point in a space. The term *light field* was first coined in 1936 by Andrey Gershun [71], for describing the radiometric properties of light in three-dimensional space. Scientifically, light rays are described by the 5-dimensional plenoptic function, which each ray is defined by three coordinates in 3D space (3 dimensions) and two angles to specify their direction in 3D space (2 dimensions).

Light fields were introduced into computer graphics in 1996 for image-based rendering applications, i.e. to compute new views of a scene from pre-existing views without the need for scene geometry [72].

Note that also the term "radiance field" may be used to refer to similar, or identical concepts such as light fields. In an extension, neural radiance fields (NeRFs) are a subject of latest research, for details on NeRFs refer to clause 4.3.6.1.

Light field representations can be divided into two types:

- **the plenoptic function:** as shown in Figure 4.3.6.2-1(a), the plenoptic function represents the light field in a 7-dimensional function, which includes the position of viewer's eye (x, y, z), direction (Θ, Φ in polar coordinates), wavelength (λ), and time (t). However, in practical applications, the 4D light field function $L(u, v, s, t)$ based on two planes, as shown in Figure 4.3.6.2-1(b), is more commonly used. In 4D light field function, (u, v) are the coordinates of the first plane, representing the angular information of the light field, while (s, t) are the coordinates of the second plane, representing the spatial information of the light field.

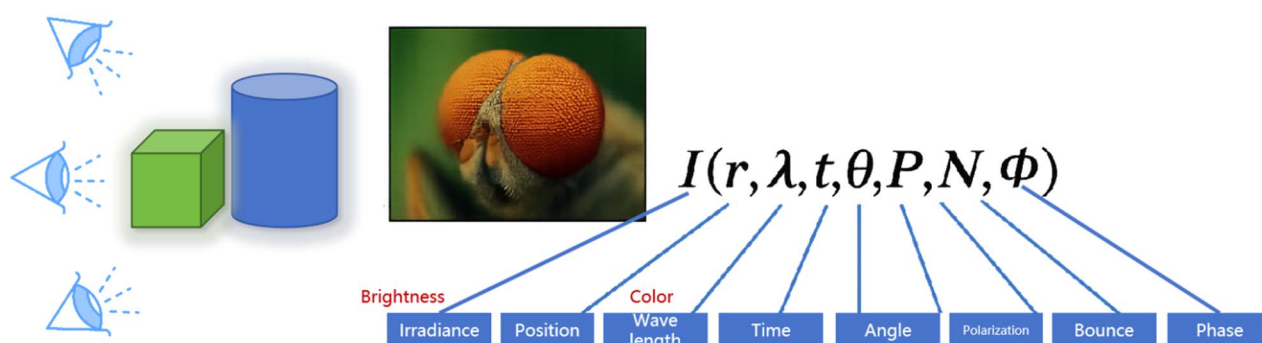


Figure 4.3.6.2-1(a), 7D plenoptic function

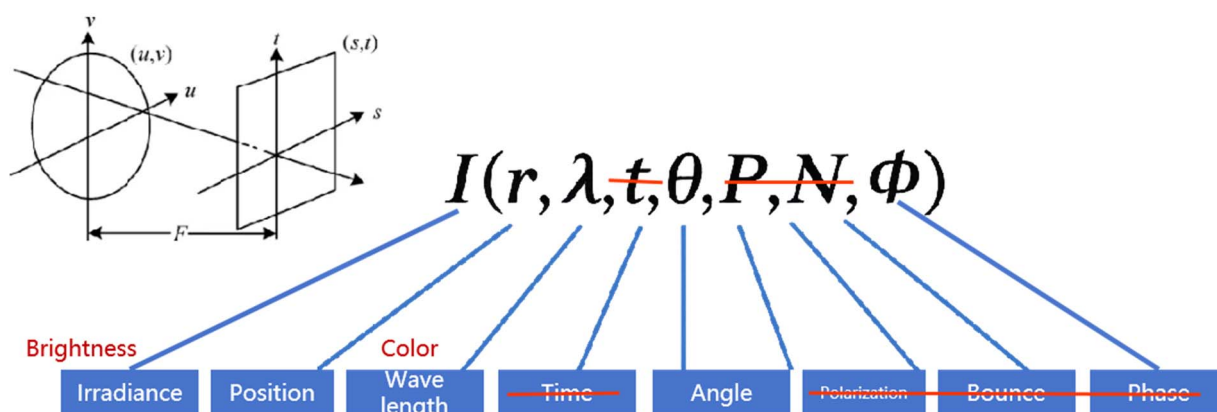


Figure 4.3.6.2-1(b), 4D plenoptic function

- **the reflectance field:** as shown in Figure 4.3.6.2-2, the reflectance field corresponds to the plenoptic function but focuses on the emission surface. It describes the transport of light between the incident light on an object and the light exiting from it.

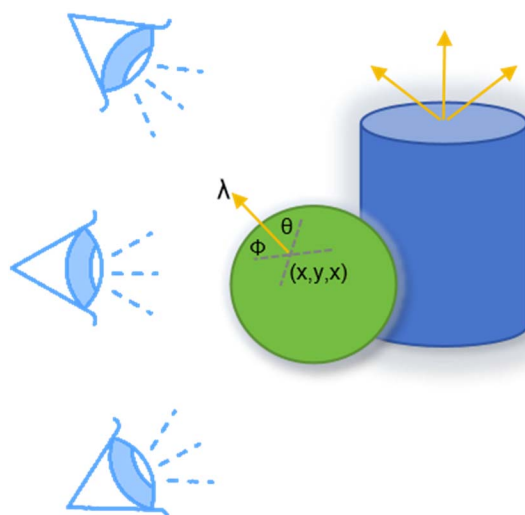


Figure 4.3.6.2-2, reflectance field

A light field can capture multiple-dimensional data, including spatial, angular, spectral, and temporal domains. Compared to traditional 2D video, the light field provides more 3D information, offering viewers an immersive visual experience. It also enables intelligent system to better understand the 3D environment or objects. Therefore, light field-related technologies are gradually being applied in fields such as industrial inspection, embodied intelligence, virtual/augmented reality, glasses-free 3D displays and so on, demonstrating vast application potential. For example, in the field of industrial inspection, data captured by light field cameras can be used to compute and render multi-view images, synthesize focus stacks, flexibly control depth of field, and achieve full volumetric reconstruction, thereby improving the accuracy of industrial inspection and quality control. Light field video can also be used in high-precision 3D reconstruction of people, objects, and environments, reducing the cost of digital content production, which can be applied in the metaverse and digital twin cities.

Table 4.3.6..21-1 provides an overview of existing lightfield technologies and their properties.

Table 4.3.6.2.1-1 Existing lightfield technologies and their properties

Properties	Google Lightstage X4 [73]	USC Lightstage X6 [74]	PlenOptic Stage [75]
Type	Color gradient, polarized light, directional light, OLAT	White gradient, OLAT	Gradient light, polarized light, directional light, color gradient, OLAT
Number of Lights	331	1111	162~460
Number of Cameras	90~100+	3	6~32
Acquisition speed	60fps	990fps	5fps
Resolution	4K	2K	6K/16K
Output frame rate	30fps	30fps	5fps

NOTE: Other resolution may include 1536 * 2048, 3840 * 2160, 7680*4320 (45-100 views).

4.3.6.2.2 Production and Capturing Systems

Light fields are typically produced either by rendering a 3D model or by photographing a real scene. In either case, to produce a light field, views must be obtained for a large collection of viewpoints. Depending on the parameterization, this collection typically spans some portion of a line, circle, plane, sphere, or other shape, although unstructured

collections are possible. Devices for capturing light fields photographically may include a moving handheld camera or a robotically controlled camera, an arc of cameras (as in the bullet time effect used in *The Matrix*), a dense array of cameras, or other optical systems. The number and arrangement of images in a light field, as well as the resolution of each image, are referred to as the "sampling" of the 4D light field.

Some capturing systems for light field videos can be captured using several techniques, including:

- **Sequential Imaging:** Sequential imaging involves capturing multiple images with a moving camera, recording light field information through a sequence of images. For example, a mechanical system can control the camera to move along a spherical trajectory, thus capturing static light field information inside the sphere. While the sequential imaging method can obtain light field with high spatial and angular resolution, the process is time-consuming due to the camera's mechanical movement, making it unsuitable for light field video capture of dynamic objects or scenes.
- **Camera Array or camera modules:** To improve the efficiency of light field acquisition, multiple cameras can be combined into a camera array, capturing the scene from different angles and positions to obtain light field video. Camera array-based light field video acquisition systems are highly time-efficient and can capture dynamic objects. However, due to the typically large size of the camera array and the high hardware costs, along with the challenges of synchronization and data transmission between multiple cameras, its practical application has certain limitations.
- **Plenoptic Camera:** To balance system cost and acquisition time, some researchers have proposed using a microlens array combined with a single camera for light field acquisition. A plenoptic camera places the microlens array between the main lens and the image sensor, offering the advantages of portability and low cost. This device can efficiently capture dynamic scenes, therefore they are increasingly being used in light field acquisition. PlenoptiCam is an open-source, cross-platform software tool that processes raw plenoptic camera images into light field data through 4D image alignment and calibration: <https://github.com/hahnec/plenoptcam>.

4.3.6.2.3 Rendering and Display Systems

Light field video can be viewed on different rendering and display systems, including:

-AR glasses: CREAL™ light-field AR glasses, <https://creal.com/ar/>

- VR HMD: Meta Quest, CREAL™ light-field VR HMD, <https://creal.com/vr/>
- Light Field Display: Looking Glass (<https://www.holoxica.com/looking-glass-4k>), Leia

For rendering light-field video, it takes image rendering toward a “no-geometry-required” solution but use multiple image views. For example, there is an open-source interactive light field renderer using dynamically reparameterized light fields in GitHub: <https://github.com/linusmossberg/light-field-renderer>.

4.3.6.2.4 Supporting Information

The following is the end-to-end Light Field Video System proposed by Google [76], including capturing, reconstructing, compressing, and rendering high quality immersive light field video. The immersive light fields are recorded using a custom array of 46 time-synchronized cameras distributed on the surface of a hemispherical, 92cm diameter dome. From this data it produces 6DOF volumetric videos with a wide 80-cm viewing baseline, 10 pixels per degree angular resolution, and a wide field of view (>220 degrees), at 30fps video frame rates. The resulting RGB, alpha, and depth channels in these layers are then compressed using conventional texture atlasing and video compression techniques. The final compressed representation is lightweight and can be rendered on mobile VR/AR platforms or in a web browser.

Multiple compression techniques have been proposed for the light field video, they are mainly categorized into three groups:

- Transform-based methods: these approaches typically involving using a transformer, such as the discrete cosine transform (DCT) or the discrete wavelet transform (DWT). For example, the MuLE [77], which has been adopted by the JPEG pleno standardization committee.
- Prediction-based methods: some studies interpret light fields as multi-view sequences and use multi-view extension such as MV-HEVC for compression.

- Learning-based methods: these are recent development in light field compression. They leverage the power of machine learning techniques (e.g., GAN or CNN) to improve encoding efficiency,

MPEG is currently working on Lenslet Video Coding (LVC), which focuses on dense light field representations, use cases and requirements, and dedicated codecs [78].

The content can be delivered with a DASH-compliant framework, an open streaming media standard for field of light displays (SMFoLD) [79] had also been developed for streaming dynamic scene. However, the work appears to have been discontinued, as there have been no updates since 2018.

Typical quality criteria for evaluating light field video includes:

- 3D IQA, PSNR and SSIM Quality Metrics
- FR light field metric [80], which considers global spatial quality metrics based on viewing structure matching, local spatial quality based on near-edge MSE, and angular quality based on multiview quality analysis.
- JPEG is also preparing standardization activities [81] in the domains of objective and subjective quality assessment for light fields, improved light field coding modes, and learning-based light field coding.

4.3.6.2.5 Benefits and Limitations

4.3.6.2.5.1 Benefits

The light field videos offer the following main benefit:

- Immersive visual experience, “cover all the different perspectives so the user can choose which one he wants to have and of course different users can choose different perspectives.” [82]

Some advantages and benefits of lightfields are provided in [83]:

- the light field content production approach has the benefit that it is a full representation of what is captured, is high resolution, one can observe in stereo and can move within this content, to a predefined degree.
- Content can be captured by a micro lens array or produced by computer graphics. Light fields allow the high fidelity of models, textures, lighting, and reflections.
- Lightfield captures are holographic, i.e. they contain all possible views within a preset range that can be defined.
- Light field captures have parallax, an overlap of object depth based on head movement. This increases immersion and presence, which is not available in 3DOF omnidirectional videos.
- Captures can have higher quality reflections, complex lighting, or realistic physics which would not be possible in real-time 3D graphics.
- Light fields focus or aperture can be adjusted on the fly based on depth versus a 2D video which needs to be in focus at the time of filming. This is because light fields can estimate depth based on the data acquired.
- Just like 3D models, multiple light field captures can be incorporated together and can be manipulated.
- Most of the captured data is not needed and can be removed. For example, thousands of high quality photos of small movements can be compressed to be a fraction of this size by removing this redundant visual info.

4.3.6.2.5.2 Limitations

The large number of views and captures in light field videos generate massive amounts of data, leading to significant storage and bandwidth demands. Even when content is delivered promptly, current handheld devices struggle to load these resources in real-time.

Some disadvantages and limitations of lightfields are provided in [83]:

- Lightfields require a massive amount of high quality pixel based information, regardless whether it is from a real camera or from a virtual camera. In addition, the more movement in a scene means more individual captures of this information.

- Lightfields have restrictions in the volume they can capture, so they really only work for content where edges are not visible. Light fields are not useful for everything, but very targeted use cases.
- Lightfields are not truly volumetric (aka 3D), but fake 3D. If you try to go past the window of the capture area, the illusion disappears.
- Though light field technology had been on computer visionaries' radar since the early 1990's, even the biggest players struggled to launch a tangible version to market. Often light fields have been relegated to white papers, rarely finding their way into a commercial project.

4.3.6.3 3D Gaussian Splatting

4.3.6.3.1 Introduction

3D Gaussian Splatting (3DGS) is an emerging method that's gaining attention for its ability to render highly realistic scenes with impressive efficiency and speed. It shows potential in addressing many of the limitations associated with other representation formats. As 3D Gaussian Splatting is a rapidly evolving field, new developments and insights are emerging regularly. This documentation reflects the state of the art at the time of writing and may not capture the most recent advancements. However, a comprehensive, searchable database of 3D Gaussian Splatting papers is available through the following link, which will help you stay updated with the anticipated surge of research in the coming months:

- **Awesome 3D Gaussian Splatting Paper List:** <https://mrnerf.github.io/awesome-3D-gaussian-splatting/>

4.3.6.3.2 Overview

3D Gaussian Splatting (3DGS) [85], also referred as Gaussian Splatting Radiance Field, is an explicit radiance field based 3D representation that represents 3D scene or objects using a large number of discrete 3D anisotropic balls or particles, each defined by its spatial mean μ and covariance matrix Σ [86]:

$$G(p) = \exp\left(-\frac{1}{2}(p - \mu)^T \Sigma^{-1}(p - \mu)\right)$$

The covariance matrix Σ is parameterized by using a scaling matrix S and a rotation matrix R , such that $\Sigma = RSS^T R^T$. Each 3D Gaussian is associated with a color c and an opacity α . During rendering, these Gaussians are projected (rasterized) onto the image plane, forming 2D Gaussian splats $G'(x)$. The 2D Gaussian splats are sorted from front to back tile-wisely, and α -blending is performed for each pixel x to render its color as follows:

$$C(x) = \sum_{i \in N} c_i \sigma_i \prod_{j=1}^{i-1} (1 - \sigma_j), \sigma_j = \alpha_j G'_j(x)$$

The color of each Gaussian, c , is represented by Spherical Harmonics (SH) as $k_l^m Y_l^m(\omega_{view})$ to provide view-dependent effects, where (l, m) is the degree and order of the SH basis Y_l^m , k_l^m is the corresponding SH coefficient, and ω_{view} specifies the viewing direction.

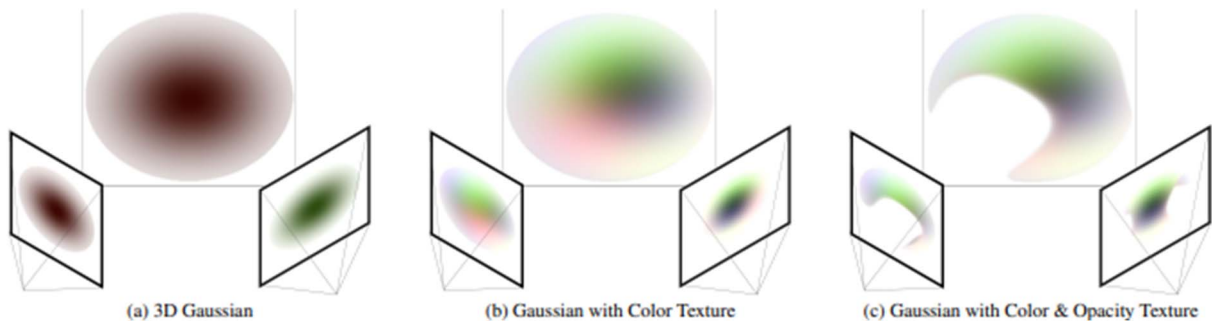


Figure 4.3.6.3.2-1 3D Gaussian Splatting (3DGS) representation [95]

The data need to perform this 3DGS rendering process are, for each point, known as a Gaussian:

- 3 position values
- 4 rotation values
- 3 color values
- 3 scale values
- 1 transparency value
- 45 spherical harmonics values

Currently, the supported formats for 3D Gaussian Splatting data can be:

- A .PLY file, in which a detailed splat of an outdoor scene might exceed 250 MB (1888 bits by points for float 32 values).
- .splat format, a Javascript types serialized version of .PLY datas.
- The .SPZ format, shrinks the file size from a standard PLY file by using more compressed representation by reduce the different values that are associated with each Gaussian splat. This format reduces the size of the 3DGS files by quantizing the data, reducing the number of spherical harmonics used and compressing the quantized data, using lossless data compression process. This reduction in data in term of quantization and reduction disturbs the 3D object and is not lossless.

The Table 4.3.6.3.2-1 summarized the parameter types for .PLY and .SPZ format:

Table 4.3.6.3.2-1 Existing 3D Gaussian Splatting (3DGS) representation Formats

Element	SPZ Format	PLY Format
Positions	24-bit fixed point integer with adjustable fractional bits	32-bit or 64-bit floating-point
Rotation	3 components of a quaternion stored as 8-bit signed integers	4 components of quaternion as 32-bit floats
Color (RGB)	8-bit unsigned integers per channel	Typically 8-bit or 32-bit floats per channel
Scales	8-bit log-encoded integer	Typically 32-bit or 64-bit floating-point
Alphas	8-bit unsigned integer	Typically 32-bit float
Spherical Harmonics	8-bit signed integers for coefficients, with 4-5 bits of precision	Varies, but usually stored with higher precision (e.g., 32-bit floats)
Number of Spherical Harmonics	0, 9, 24 or 45 values according to compression parameter	45 values

The implementation of 3D Gaussian Splatting (3DGS) involves numerous elaborate steps, which may include:

- **Pre-process:** Estimate 6DOF poses corresponding to each view, and also obtain camera intrinsic parameters. These 6DOF poses (extrinsics), camera intrinsics, as well as RGBs are then processed in SfM block to obtain: (i) 3D point cloud - i.e., 3D projection of salient points observed in multiple images, as well as (ii) refined camera intrinsics and poses, optimized in a large nonlinear solver system.
- **Structure from motion:** This process starts by creating a point cloud from images e.g., using the SFM method [87] with the COLMAP library

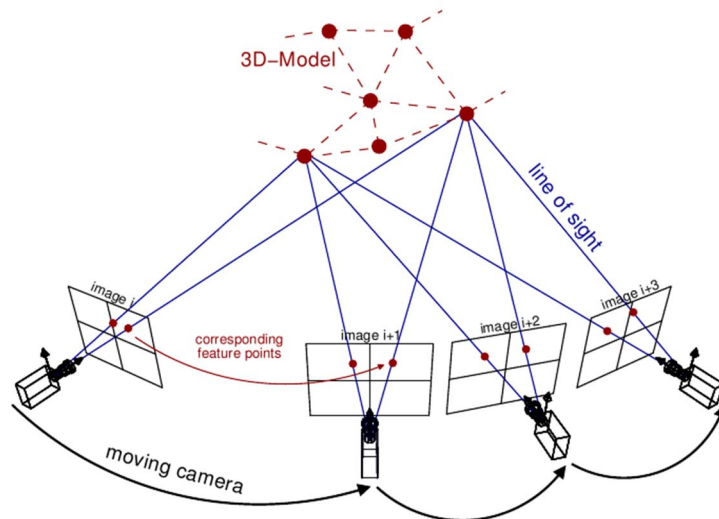


Figure 4.3.6.3.2-2 Structure from Motion (SfM) photogrammetric principle [94]

- **Convert to gaussian splats:** Each point is then converted to a Gaussian splat, which is described by parameters such as position, covariance, color, and transparency.
- **Differentiable Gaussian rasterization:** Every 2D Gaussian requires differentiable Gaussian rasterization to be projected from the viewpoint of the camera, sorted according to depth, then repeated both backwards and forwards combined for every pixel.
- **Adaptive Density Control:** The method for dynamically adjusting the number, density, and parameters of Gaussians to accurately and efficiently represent the 3D scene. This involves two steps:
 - **Pruning:** If the opacity of is too small or the gaussian is too large, then it's being removed.
 - **Densification:** this step handles two issues:
 - Over-reconstruction: regions of a 3D scene are represented by excessively large or overlapping gaussians, leading to redundant and inefficient coverage of the geometry. To solve this the large gaussian is split in two parts (bottom row).
 - Under-reconstruction: regions of a 3D scene lack sufficient Gaussian coverage, resulting in missing or poorly represented geometric details. This is solved by merging/cloning two or more gaussians associated with the area (top row).
- **Training:** iterative calculations on the content itself are used to determine additional details about how the point could be stretched/scaled (covariance) and its opacity using Stochastic Gradient Descent (SGD). Through this process, a model is created with millions of points containing data such as position, color, covariance, and opacity. The inputs to 3DGS training are: (1) Refined 6DoF poses (camera extrinsics), (2) Refined (or original) camera intrinsics - a.k.a., projection matrices, (3) Images/RGBs corresponding to these poses and intrinsics, and (4) 3D points cloud - obtained using SfM.

4.3.6.3.3 Production and Capturing Systems

The formats as defined in clause 4.3.6.3 may be captured by mobile devices with several mobile apps (both Android and iOS devices) or online services utilize 3DGS technology. However, smartphones often apply automatic enhancement to each captured images, aiming for the best result. These optimizations, such as sharpening edges, adjusting exposure, and optimizing colors, can introduce noise or errors that are not suitable for Gaussian Splatting training. To mitigate these challenges, a photogrammetry capture method with professional-grade cameras (e.g., Interchangeable Lens Camera), standardized settings, and a streamlined capture process can be utilized to meet the requirements.

The New York Times (NYT) development team explored the practical applications of 3D Gaussian Splatting (3DGS) for spatial journalism. They tested a range of capture and processing techniques using various hardware and software, evaluating solutions for both desktop and mobile devices, and give an overview of the practical takeaways they learned exploring gaussian splatting for spatial journalism.

- Quality: High resolution RAW stills when photographing a stationary object or space that requires lots of detail. More resolution generally equates to a more detailed splat — up to a point. They tested images up to 45 megapixels and video up to 8K and found little increase in quality above 20 megapixels or 6K video. If you opt for extreme resolutions, you may need to rescale your images or video before processing.
- Speed: Using a high frame rate during video capture (e.g., 120fps vs 24fps) can ensure less subject movement between frames, more overlap between adjacent frames, decrease motion blur, and help guarantee sufficient coverage. Using a burst mode or automated continuous capture mode when capturing still images similarly speeds up the process.

4.3.6.3.4 Rendering and Display Systems

During the rendering process, each Gaussian is rasterized onto the screen according to its parameters. Alpha blending techniques are used to smoothly blend the transparenced splats to create a continuous surface appearance.

In the first step of the rendering process, the 3D points are sorted according to the viewer's position to be projected in the correct order on the screen and allow for proper color mixing. Rasterization of Gaussian splats works for each point by projecting the ellipsoid made based on the covariance matrix on the screen and for each pixel covered by the current splat, the color is calculated using the user's position and the values of the spherical harmonics. Alpha blending adds the colors of all the splats covering the current pixels to obtain the final color and achieve high-quality rendering.

Various implementations of the rendering process have been proposed in the literature. The standard implementation of the rasterization process can be made on CPU or on GPU using rendering shaders or compute shaders with various 3D graphics API: OpenGL, Vulkan, DirectX, CUDA, etc...

To facilitate high-frame-rate and high-resolution differentiable rendering, a tile-based rasterization process has been proposed in [88]. For example, the rasterizer divides the image into a set number of tiles, assigning an index to each tile. For each Gaussian primitive, the rasterizer determines which tiles the primitive's projection intersects and generates a key-value pair for each intersecting tile. By constructing these key-value pairs, the rasterizer only needs to perform a global sort on all pairs, eliminating the need for additional sorting of primitives for each pixel.

After sorting, the key-value pairs derived from each tile are stored in contiguous memory intervals. The rendering process for each tile is then managed by a CUDA thread block, with the number of threads within each block matching the number of pixels in the tile. Each thread is responsible for the alpha blending process for its corresponding pixel, completing the final rendering.

To improve the quality of the rendered screen [92] proposes to render a 3D Gaussian splat scene with the ray tracing process. This method improves the quality of rendering by allowing for precise soft shadows, reflections, and transparency, surpassing the previously mentioned rasterization processes in terms of realism, but greatly increasing complexity.

4.3.6.3.5 Supporting Information

- Typical quality criteria for evaluating the format
 - In the image domain: The Peak Signal-to-Noise Ratio (PSNR), Structural Similarity Index (SSIM), Learned Perceptual Image Patch Similarity (LPIPS), the resultant size in megabytes (MB), the training time, and required storage.
 - In the 3D point domain: point to point/ point to plane, / point to surface for position. Attribute PSNR.
- Conversion from other formats (lossless, lossy)
 - Point Clouds: There are tools, e.g., 3D Gaussian Splatting Converter (<https://github.com/francescofugazzi/3dgsconverter>), can seamlessly convert 3DGS .ply files to a Cloud Compare-friendly format and vice-versa. Converting point cloud representations to 3DGS can effectively fill in the point clouds' holes, which is typically done after high-precision reconstruction of the point clouds. Conversely, 3DGS can also be converted into point clouds, followed by voxelizing the point cloud into 3D voxels and then projecting them onto 2D BEV grids [89].

NOTE: filling holes in point cloud, can also be achieved by different techniques.

- Mesh: Research works have discussed how to convert 3DGS to Mesh. Once converted, the quality of mesh can be further optimized to achieve better geometric and appearance,

- Uncompressed data size

A Gaussian splat of a scene is a representation of 3D points. On average, a splat contains between 0.5 and 5 million of these 3D points. Each 3D point has unique parameters that represent the scene as accurately as possible.

- Known compression technologies:

Early research on 3DGS compression is ongoing. Vector quantization (VQ) [89] had been applied in some research for 3DGS compression and encoding. Between 50:1 and 200:1 compression ratios are quite standard these days. MPEG has started an exploration that is looking at the most appropriate representation formats for 3DGS and various coding strategies.

- Extensibility of the format

The format is undergoing massive academic and industrial research, can be further expanded for more capabilities and is hence not stable. Example of use cases include: 3DGS can be further expanded for more capabilities, including dynamic 3DGS [88, 90], surface representation from 3DGS [88, 90], editable 3DGS [88, 90], 3DGS with semantic understanding [88, 90], and 3DGS-based physics simulation [88, 90].

4.3.6.3.6 glTF as a Gaussian Splat format

Gaussian Splats (GS) provide an efficient representation for static and dynamic 3D scenes by compactly encoding local geometry and view-dependent appearance. To address interoperability, backward compatibility, and progressive rendering in various applications, integrating GS into the glTF (GL Transmission Format) standard [42] is essential.

A structured method to store GS in glTF 2.0, supporting a broader set of devices, progressive downloads, and dynamic content via MPEG extensions is available in ISO/IEC 23090-14:2024/Amd.1:2025 [96]. Gaussian Splats are proposed to be stored in glTF mainly using application-specific attributes, ensuring backward compatibility by enabling legacy receivers to interpret data as a traditional point cloud representation showing the base color for each splat.

Each Gaussian splat is defined by attributes derived from the INRIA Gaussian Splat PLY format, mapped directly to glTF attributes according to Table 4.3.6.3.6-1.

Figure 4.3.6.3.6-1 Mapping of GS attributes to glTF primitive attributes via MPEG-SD-GS extension

GS Property	Corresponding glTF Attribute primitive via MPEG SD extensions
x, y, z	POSITION
f_dc_[0-2]	COLOR_n
opacity	Alpha channel of COLOR_n
rot_[0-3]	MPEG_GS_ORIENTATION (x,y,z,w)
scale_[0-2]	_MPEG_GS_SCALE
f_rest_[0-14]	_MPEG_GS_SH_COEFF_R (R channel SH coeffs)
f_rest_[15-29]	_MPEG_GS_SH_COEFF_G (G channel SH coeffs)
f_rest_[30-44]	_MPEG_GS_SH_COEFF_B (B channel SH coeffs)

MPEG's Gaussian Splats extension is added to glTF 2.0 primitive elements, explicitly supporting both static and dynamic Gaussian Splats leveraging MPEG timed media extensions as defined in ISO/IEC 23090-14. The primitive mode is set to 0 (POINTS), with the COLOR_n attribute referencing Vec4 type, incorporating opacity.

To facilitate progressive download and rendering, Gaussian Splat attributes are structured hierarchically in buffer views according to significance and detail level:

- Initial buffer views store POSITION attributes followed immediately by COLOR_n attributes, providing an initial coarse representation suitable for immediate visualization.
- Subsequent buffer views contain ORIENTATION and SCALE attributes.
- Spherical harmonics (SH) attributes for color are grouped by spherical harmonic order, enabling progressive refinement:
 - Level-of-Detail 0 (LoD0): 3 base color components in COLOR_n.
 - Level-of-Detail 1 (LoD1): 1st-order SH, resulting in 9 coefficients.

- Level-of-Detail 2 (LoD2): 2nd-order SH, resulting in additional 15 coefficients.
- Level-of-Detail 3 (LoD3): 3rd-order SH, resulting in additional 21 coefficients.

Different attribute sets support progressive refinement:

- `_MPEG_GS_SH_COEFF_FIRST` provides the 9 coefficients of 1st-order SH.
- `_MPEG_GS_SH_COEFF_SECOND` provides the 15 coefficients of 2nd-order SH.
- `_MPEG_GS_SH_COEFF_THIRD` provides the 21 coefficients of 3rd-order SH.

This hierarchical data organization enables efficient progressive streaming and immediate visual feedback on the receiver's device.

Reference tools are expected to be available by August 2025 to support conversion of proprietary formats into glTF or standardized MPEG scene descriptions extensions. Additionally, a renderer for the glTF GS format is expected to be made available as well.

4.3.6.3.7 Benefits and Limitations

4.3.6.3.7.1 Benefits

- Real-time Rendering with GPU acceleration.
- Accurate Reconstruction, it can capture the geometry accurately
- Explicit representation
- Ability to render complex scenes in real-time
- Interpretability of the representation, an explanation of the mathematical mechanism, i.e., the working principle, of 3DGS can help researchers analyze the complex relationships in 3D scene reconstruction technology and reveal the performance characteristics of 3DGS in depth. [93]
- Gaussian Splatting can deliver high-quality, real-time visualizations.[91].
- Gaussian Splatting has evolved to handle dynamic and deformable objects [91].
- Gaussian Splatting can be applicable to various application space, such as digital avatars and SLAM [91].
- Gaussian Splats can be stored in glTF 2.0, supporting broader set of devices, progressive downloads, and dynamic content.
- Gaussian Splats facilitate progressive download and rendering, if GS attributes are structured hierarchically in buffer views according to significance and detail level as proposed for the glTF extension [96].

4.3.6.3.7.2 Limitations

- There is a lack of industry agreement on the 3DGS format(s), due to no stable representation and compression format exists for static and dynamic 3DGS.
- Static and Dynamic 3DGS formats is evolving, multiple options are considered in current academic and industrial research. For dynamic, such research include modeling in 4 dimensions (i.e. temporal), time evolving 3DGS, and MLP predicted motion for 3DGS among others.
- High memory usage
- Not yet fully compatible with existing rendering pipelines
- Computation complexity [91], the computational demands of handling large numbers of splats, especially for high-resolution rendering and complex scenes; it requires to process large datasets for training, which can also be time-consuming and resource-intensive.
- Edge artifacts [91].

4.4 AI-Generated Beyond 2D content

4.4.1 General

Creating and capturing high-quality Beyond2D content is often a labour-intensive task that demands substantial time, expertise, or specialized capturing tools/devices, which limits the widespread adoption of Beyond 2D media. Artificial Intelligence Generated Content (AIGC) leverages AI technologies to autonomously produce content. For example, in clause 7.2.2.2, AI-powered 2D to stereoscopic 3D video methodology is introduced, which effectively reduces the reliance on high-end capture devices. Beyond this, AIGC encompasses a range of emerging technologies, including:

- Image-to-dynamic Mesh Generation,
- Text-to-dynamic Mesh Generation

which are described in the following sections. The commercialization of AIGC has attracted attention from both academia and industry, driving innovation in Beyond 2D content creation, compression technologies, and quality assessment methodologies.

Figure 4.4.1-1 illustrates a reference workflow for AI-generated beyond 2D content. The workflow positions a Media Generation AI/ML model at the core of logical reasoning, including a large language model (LLM) transforming different inputs, such as text, image, video, 3D models, actuator signals and etc into a unified tensor representation. After reasoning and inference by the AI/ML model, the output tensor is mapped back to the target modality.

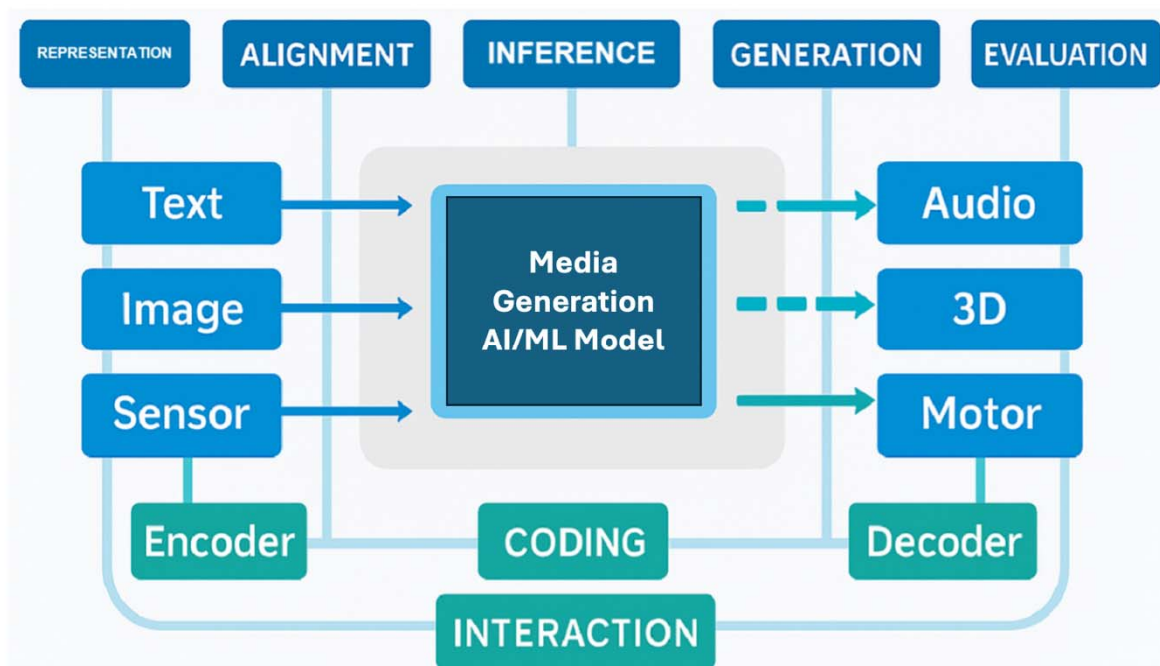


Figure 4.4-1: Workflow for AI-generated beyond 2D content

- **Representation:** The model should effectively represent and process different media types, such as text, images, video, and 3D models. Appropriate representation format should be selected for each type (e.g., CNNs for image features) to enable downstream processing and analysis.
- **Alignment:** Alignment refers to the process of matching and correlating data across different media types, enabling the model to comprehend their interrelationships. For instance, attention mechanisms can be employed to establish semantic correspondences between text and images.

- **Inference:** The model should be capable of inference capabilities, it can analyze and understand input data to extract useful information. A common approach is to leverage pre-trained large language models (LLM) to perform inference tasks.
- **Generation:** The generative modeling techniques e.g., diffusion models, should be capable of generating new content, for instance, creating 3D mesh from text prompts.
- **Evaluation:** Assessing model performance (include both subjective methodologies and objective metric) is critical to ensure output relevance and reliability.

4.4.2 AI-Generated Dynamic Mesh

4.4.2.1 General

A growing number of AI-generated mesh tools now enable the direct generation of mesh models and textures from inputs such as text or images. Compared to traditional mesh production workflows, these tools offer significant advantages in terms of time efficiency. The examples of commercial services are provided below:

- AssetGen 2.0™: Meta's AI-powered 3D mesh generation system that produces models with "geometric consistency and fine-grained details" <https://developers.meta.com/horizon/blog/AssetGen2/>
- Hunyuan 3D™: Tencent's 3D Mesh generation platform <https://3d.hunyuan.tencent.com/>
- Meshy™: <https://www.meshy.ai/>

As the technology continues to advance, the quality and efficiency of AI-generated meshes are improving. However, there are still common issues that need to be addressed, including: excessively high polygon counts, poor topology, fragmented or irregular UV layouts, coarse texture details, baked-in lighting information in the textures, and insufficient accuracy in complex scenarios (e.g., clothing wrinkle simulation errors exceeding 15%).

4.4.2.2 Image-Generated Dynamic Mesh

The task of generating dynamic meshes from images demands not only the creation of multiview geometric models based on the input image but also the extension into the temporal dimension to produce dynamic spatio-temporal content (4D). There are two main approaches for generating dynamic meshes, inference-based and optimization-based methods. As shown in Figure 4.4.2-1, the pipelines for these approaches include:

- Direct Generation: Directly generating dynamic meshes from input parameters without intermediate steps.
- Indirect Generation: Leverages diffusion models to produce multi-temporal and multi-view training data.
- Implicit Distillation: The process generates dynamic meshes through a multi-stage training framework, which combines multiple diffusion models via implicit distillation to derive generative priors.
- Explicit Supervision: Uses multi-modal data to provide explicit supervisory signals for dynamic mesh generation.

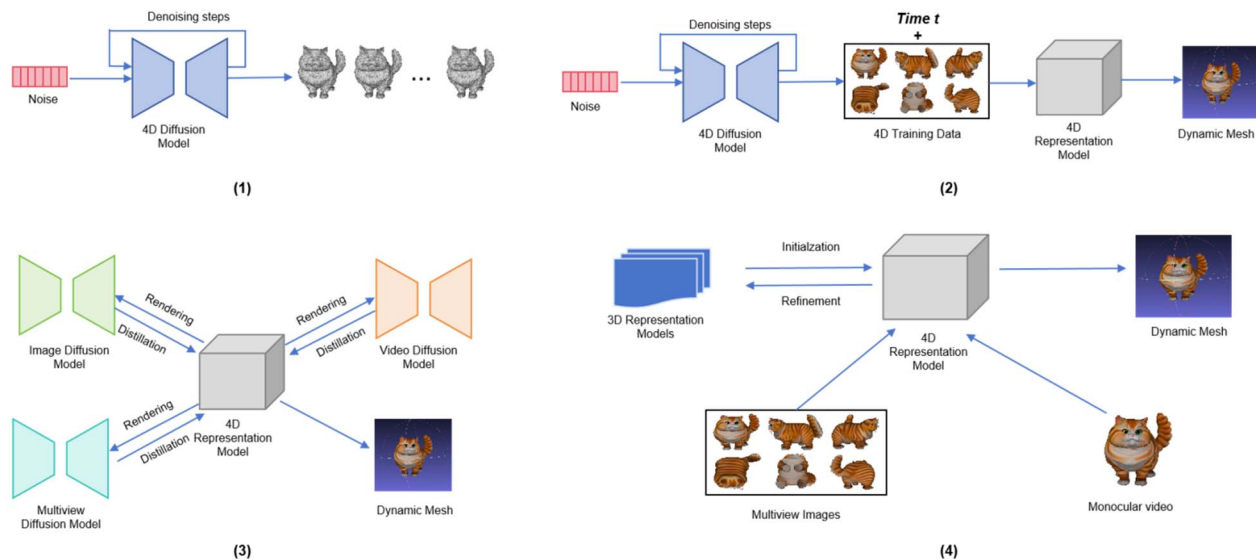


Figure 4.4.2-1 Pipelines for image-based Dynamic Mesh generation

4.4.2.3 Text-Generated Dynamic Mesh

Text-generated dynamic mesh requires both precise alignment between the object's geometry and texture semantics, and accurate synchronization of its motion dynamics with describe actions or movements (4D). For example, a typical workflow may involve the following steps:

- Text Prompt Generation: Using large language models (e.g., GPT-4) to generate text prompts.
- Image Generation: Using diffusion models to generate single-view images based on these text prompts.
- Multi-view Synthesis: Leverage video or multi-view diffusion models to generate multi-view images from single-view images rendered from different angles.
- Dynamic Mesh Animation: Reconstruct 3D mesh and create a dynamic mesh (4D) by animating the vertices over time.

5 Overview of existing "Beyond 2D" Video Capabilities in 3GPP

NOTE: This clause summarized existing beyond 2D video capabilities in 3GPP from at least TS.26.119 and TS.26.118.

5.1 Introduction

This clause summarizes the existing beyond 2D video capabilities in relevant 3GPP-based services.

In Release 18, the beyond 2D video capabilities include support for 3GPP video codecs, H.264 (AVC) [158] and H.265 (HEVC) [159]. Both codecs are defined as part of AR Video capabilities in 3GPP TS 26.119 [7], and VR Video Profiles in 3GPP TS 26.118 [6]. The highest defined profile/level combinations are:

- **AVC-8K-Dec-8:** The capability of supporting up to eight (N=8) concurrent decoder instances with the aggregate capabilities of H.264 (AVC) Progressive High Profile Level 6.1[158].
- **HEVC-8K-Dec-8:** The capability of supporting up to eight (N=8) concurrent decoder instances with the aggregate capabilities of H.265 (HEVC) Main10 Profile, Main Tier, Level 6.1[159].

Support for stereoscopic MV-HEVC for low delay stereoscopic 3D video applications was recommended by TR 26.966 [10] and is being addressed in a Release 19 work TS 26.265 [11]. More details on the beyond 2D video capabilities for different services are provided in the remainder of this clause.

5.2 AR Video Capabilities

3GPP TS 26.119 [7] specifies the mandatory and optional media capabilities and profiles to be supported for each XR device type. These media capabilities include support for video codecs (AVC and HEVC), audio codecs (EVS, IVAS and AAC-ELDv2), scene description formats, and XR system capabilities. Table 5.2-1 summarized the Beyond 2D video capabilities defined in clause 7 of TS 26.119 [7].

NOTE: The definition of concurrent video decoder instances can be found in clause 7.1.2.1 of TS 26.119 [7].

Table 5.2-1: Summary of Operation Points

Operation Point Name	Max Concurrent Video Decoder Instances	Decoding Capabilities
AVC-FullHD-Dec-2	2	Aggregate decoding capabilities of H.264/AVC HP@L4.0
AVC-UHD-Dec-4	4	Aggregate decoding capabilities of H.264/AVC HP@L5.1
HEVC-UHD-Dec-4	4	Aggregate decoding capabilities of H.265/HEVC MP10@L5.1
UHD-Dec-4	4	Aggregate capabilities of <i>AVC-UHD-Dec-4</i>
		Aggregate capabilities of <i>HEVC-UHD-Dec-4</i>
		Decoding up to 4 bitstreams, each not exceeding the capabilities of H.264/AVC HP@L4.0 or H.265/HEVC MP10@L4.1.
AVC-8K-Dec-8	8	Aggregate capabilities of H.264/AVC HP@L6.1
HEVC-8K-Dec-8	8	Aggregate capabilities of H.265/HEVC MP10@L6.1
8K-Dec-8	8	Aggregate capabilities of <i>AVC-8K-Dec-8</i>
		Aggregate capabilities of <i>HEVC-8K-Dec-8</i>
		Decoding up to 8 bitstreams, each not exceeding the capabilities of H.264/AVC HP@L4.0 or H.265/HEVC MP10@L4.1.
		Decoding up to 4 bitstreams, each not exceeding the capabilities of H.264/AVC HP@L5.1 or H.265/HEVC MP10@L5.1.

5.3 VR Video Profiles

The VR profiles for streaming services are defined in TS 26.118 [6], specifying the coded representation and media profile of 360 VR distribution signals. Table 5.3-1 provides an overview of the 360 VR relevant formats considered in the context of 3GPP VR Profiles.

For restrictions on source formats such as resolution and frame rates, content generation and encoding guidelines, refer to TS 26.118 [6], Annex A.

Table 5.3-1: High-level Summary of Operation Points

Operation Point name	Decoder	Bit depth	Typical Original Spatial Resolution	Frame Rate	Colour space format	Transfer Characteristics	Projection	Rotation	RWP	Stereo
Basic H.264/AVC	H.264/AVC HP@L5.1	8	Up to 4k	Up to 60 Hz	BT.709	BT.709	ERP w/o padding	No	No	No
Main H.265/HEVC	H.265/HEVC MP10@L5.1	8, 10	Up to 6k in mono and 3k in stereo	Up to 60 Hz	BT.709 BT.2020	BT.709	ERP w/o padding	No	Yes	Yes
Flexible H.265/HEVC	H.265/HEVC MP10@L5.1	8, 10	Up to 8k in mono and 3k in stereo	Up to 120 Hz	BT.709 BT.2020	BT.709, BT.2100 PQ, BT.2100 HLG	ERP w/o padding CMP	No	Yes	Yes
Main 8K H.265/HEVC	H.265/HEVC MP10@L6.1	10	Up to 8k in mono and 6k in stereo	Up to 60 Hz for 8K and 120 Hz for 4k	BT.709 BT.2020	BT.709, BT.2100 PQ, BT.2100 HLG	ERP w/o padding	No	Yes, but restricted to coverage	Yes

Table 5.3-2 summarizes the video operation point, sample entry, and DASH integration associated with each video media profiles defined in clause 5.2 of TS 26.118 [6].

Table 5.3-2 Video Media Profiles

Media Profile	Operation Point	Sample Entry	DASH Integration
Basic Video	Basic H.264/AVC	resv avc1	Single Adaptation Set Single Representation streaming
Main Video	Main H.265/HEVC or Main 8K H.265/HEVC	resv hvc1	Single or Multiple independent Adaptation Sets offered Single Representation streaming
Advanced Video	Flexible H.265/HEVC	resv hvc1, hvc2	Single or Multiple dependent Adaptation Sets offered Single or Multiple representation streaming

5.4 Messaging Services

3GPP TS 26.143 [8] specifies the media types, formats, codecs capabilities and profiles for the messaging applications used over the 5G System. The document extends to codecs for speech, audio, video, still images, bitmap graphics, 3D scenes and assets, and other media in general, as well as scene description.

Specifically, the 2D video capabilities defined in TS 26.143 [8] clause 6.2 are fully aligned with 5G Media Streaming in 3GPP TS 26.511 [9]:

- **AVC with HD and Full-HD resolutions**
- **HEVC with HD, Full-HD and UHD resolutions**

For Beyond 2D video capabilities, as HEVC simulcast and HEVC frame packing already been included in SA4 specifications and given the coding benefits MV-HEVC provides compared to these solutions, the support for stereoscopic MV-HEVC for low delay applications of stereoscopic 3D video was recommended by TR 26.966 [10]. This aspect is being addressed in a Rel-19 work TS 26.265 [11].

6 Evaluation and Characterization Framework

6.1 Overview

Generally, the test and characterization framework as documented in TR 26.955, clause 5 also applies to this document. This clause only documents differences and extensions that are needed for beyond 2D Evaluation and characterization framework.

The overview of the evaluation framework for the B2D messaging is presented in Figure 6.1-1. Representative reference sequences are collected and stored in a well-defined B2D format. For a video encoder, the configuration is provided that matches the application constraints. The resulting video streams are “pseudo”-packaged in order to determine the file size/bitrate. The data is then unpackaged, and a B2D video decoder is used to reconstruct data in the B2D format again. The data is stored. The original sequence and the recovered sequence are used determine metrics. The sequences may also be inspected subjectively.

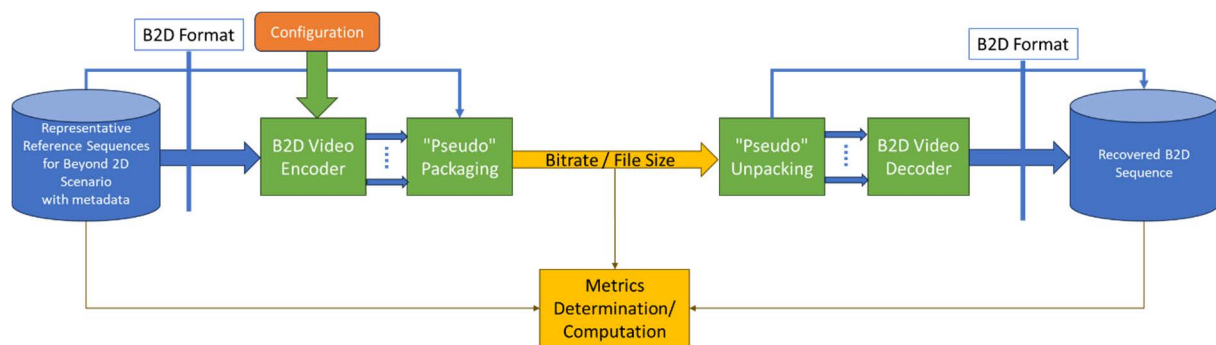


Figure 6.1-1 B2D Evaluation framework

6.2 Reference Sequences

This document provides reference sequences that are used to generate anchors and are also made available in order to generate test bitstreams for other codecs. Reference sequences are selected to be representative for a scenario.

Reference sequences are described in Annex C of this document along with their properties and their licenses. A format for raw reference sequences based on a JSON schema is defined in clause B.2.

Annex D describes how to upload new proposed reference sequences and how to download the reference sequences.

6.3 Reference Software Tools

The reference software tools for each scenario are described in details in clause 7.

6.4 Metrics

The performance metrics for each scenario are defined in clause 7

6.5 Encoding Constraints

The encoding constraint definition in clause 5.6 of TR 26.955 also apply for this report.

In addition, the following is defined:

- **Equal Quality Views:** equal quality views refers to the encoding such that each view when decoded has the same quality target, typically applying the same QP.

7 Considered Scenarios

7.1 Introduction

This clause collects relevant scenarios and corresponding workflows for beyond 2D video services, based on the template defined in Annex A. For each scenario, the following information is provided:

- **Motivation:** This provides context for the scenario. What is the market relevance of the proposed scenario over the next few years, supported by key indicators?
- **Description of the scenario:** This provides a description of the corresponding beyond 2D end-to-end workflows, which includes identifying and defining beyond 2D formats being used in the context and representation technologies to deliver these formats.
- **Source format properties:** This defines a clear range of the considered and relevant source formats, including the signal properties, but also the characteristics of the content.
- **Encoding and decoding constraints and settings:** This provides details on encoding settings and constraints for the codecs mapped to the scenarios.
- **Performance metrics and requirements:** Documents the relevant performance metrics, and/or the main KPIs of the scenario.
- **Interoperability Considerations:** Documents relevant interoperability requirements.
- **Referenced Sequences:** Defines the reference sequences that are selected for this scenario in order to do the evaluation. A justification is provided, why this sequence is selected.
- **Test Conditions:** Provides detailed test condition, for example based on a reference software together with the sequences and configuration parameters.
- **External Performance:** Provides performance data available externally.
- **Additional Information:** Provides additional information, including industry activities, implementation constraints, and innovations.

7.2 Scenario 1: UE-to-UE Stereoscopic Video Live Streaming

7.2.1 Motivation

Live Streaming services can be deployed across various platforms, including social media platforms like YouTube Live™, Facebook Live™, and TikTok™, as well as through e-commerce platforms such as eBay™ and Taobao™ [97]. It significantly impacts marketing by providing a dynamic and interactive channel to directly connect markets and their target audiences in real time. To continue captivating users, it's essential to explore a more immersive live streaming experience by incorporating beyond 2D video.

Most of the current Beyond 2D streaming services provided by network operators, services providers, and device manufacturers on the market are based on the stereoscopic video format, as defined in clause 4.3.2. In terms of distribution, existing stereoscopic 3D video formats, such as frame-compatible side-by-side and 2D video plus depth. Particular emphasis is given to the DVB systems [99] and IP transport, focusing HTTP/TCP streaming, adaptive HTTP streaming, RTP/UDP streaming, P2P Networks, and Information-Centric Networking-ICN. Hybrid transport technologies, combining broadcast and broadband networks for video delivery are also addressed. The most important standards are MPEG-2 systems, which is used for digital broadcast and storage on Blu-ray discs, real-time transport protocol (RTP), which is used for real-time transmissions over the Internet, and the ISO base media file format, which

can be used for progressive download in video-on-demand applications [100]. In clause 6.2 of TR 26.905 [101], it provides a DASH-based streaming solution for streaming Stereoscopic 3D video.

7.2.2 Description of the Anticipated Application

7.2.2.1 Overall Description

3GPP until now has very restricted set of services but based on the considerations in clause 7.2.1, the following encoding benchmark capabilities are considered for decoding:

- The capability of supporting up to two (N=2) concurrent decoder instances with the aggregate capabilities of H.265 (HEVC) YUV 4:2:0, 10 bit, Max Resolution 4096 x 2048.

The considered scenarios is low-latency streaming. Important aspects that are expected to be considered when evaluating a codec in the context of this UE-to-UE Stereoscopic Video Live Streaming scenario are:

- Quality and Coding Efficiency:
 - High and uninterrupted visual quality, taking into account the services constraints.
 - The ability to compress 2 or more B2D streams in real-time to minimize latency requirements.
 - Any savings can provide significant benefits due to the expected large volume of the traffic either in quality or network utilization.
- Considered settings for encoding:
 - Low-latency settings
- Encoding in this scenario is typically done as:
 - Live and On-Demand distribution and encoding
 - Sever and Cloud-based Encoding

7.2.2.2 Capturing and processing

The existing and emerging capture methods include:

- **Stereoscopic camera:** a dual-lens camera which can directly capture stereo 3D video. For example, the SpatialLabs Eyes™, a stereoscopic camera capable of capturing at up to 8-MP (aka 4K) per eye at 30 fps or 2K per eye at 60 fps (<https://www.tomshardware.com/cameras/3d-call-me-maybe-acers-new-spatiallabs-camera-live-streams-impressive-3d-video-in-8k-but-few-can-view-it>). Another example is the ZTE Nubia Pad 3D II™, which can capture stereo 3D video at up to 13-MP per eye at 30 fps with the rear camera and 8-MP per eye at 30 fps with the selfie camera.
- **3D Camera Rig:** The cameras setup is shown in the figure below, which consists of two identical HD camcorders (Canon HG-20™) and an adjustable stereo mount. The mount ensures that optical axes of the cameras are parallel and supports the continuous adjustment of the camera distance in the range 7-50 cm. To ensure matching of the focal length the wide angle end of the zoom lens with a focal length of 43 mm has been used. In order to match the cameras with each other the focal length, white balance and shutter speed have been set manually. The synchronized operation of the two camcorder is ensured through the use of a single remote control. The camcorders support the capture of images with a resolution of 1920×1080 pixels and store them as high quality JPEG files.



Figure 7.2.2.2-1 Camera Rig for stereoscopic video capture (AI generated Image)

- **AI Based 2D-to-Stereo3D Conversion:** The AI-based conversion leverages deep neural networks to perform real-time, end-to-end conversion of 2D videos and images into stereoscopic 3D format [98]. This technology is proving commercially viable and meets the growing demand for high-quality stereoscopic images, as demonstrated by commercial services.

For UE capable of directly capturing stereoscopic video on the device, it pre-processes the captured video frames into a well-defined B2D format and sends them to the encoder as input. The encoded B2D video streams are then streamed to the streaming server within the network, where the server may transcode them into different bitrates and distribute them to various audiences. The receiving end decodes B2D video streams and perform post-processing to adapt to the rendering system.

For UE limited to capturing only 2D video (e.g., UE with a monocular camera), the UE initially encodes the regular 2D video and streams it to a cloud server capable of real-time 2D-to-beyond 2D transcoding (a generic pipeline for this transcoding process is described in Figure 7.2.2.2-2). The cloud server then encodes the transcoded B2D video and streams it to the streaming server.

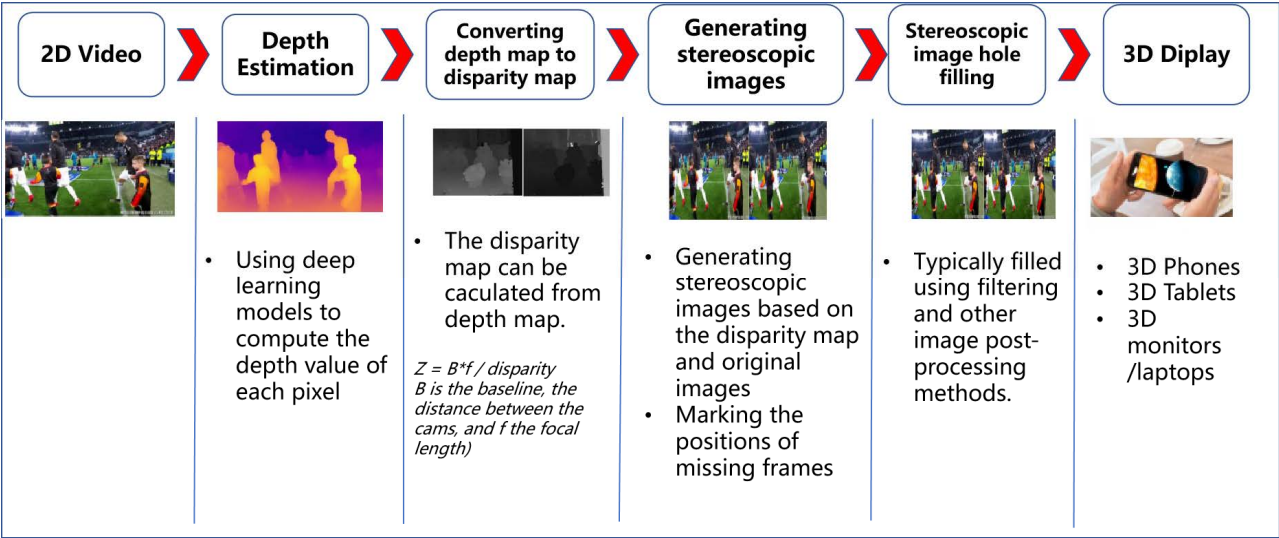


Figure 7.2.2.2-2 Pipeline for 2D-to-Stereo3D Conversion

7.2.2.3 Encoding

The following solutions can be used to realize this scenario:

- Concurrent H.265/HEVC
- MV-HEVC

7.2.2.4 Packing and Delivery

The content can be delivered using regular ISO BMFF based distribution, including streaming with DASH/HLS/CMAF.

7.2.2.5 Decoding

The following solutions can be used to realize this scenario:

- Concurrent H.265/HEVC decoding capabilities
- MV-HEVC

7.2.2.6 Rendering

Rendering can be on:

- Backward-compatible to 2D presentation, e.g., a mobile phone, but the stereoscopic effect is lost in this case.
- A device for 3D presentation, e.g., autostereoscopic displays, VR headset, and AR glasses, these devices can track the viewer's eye position and adjusts the 3D effect in real-time for single viewer applications (parallax adjustment) and rendering.

7.2.3 Source Format Properties

Table 7.2.3-1 provides an overview of the different source signal properties for UE-to-UE Stereoscopic Video Live Streaming. This information is used to select proper test sequences.

Table 7.2.3-1 UE-to-UE Stereoscopic Video Live Streaming Source Properties

Source format properties	B2D Live Streaming
Number of views	2
Spatial resolution for each view	For each view: 1920 x 1080 2560 x 1600
Chroma format	Y'CbCr, RGB
Chroma subsampling	4:2:0
Picture aspect ratio	32:9 16:9 16:10
Frame rates	25, 30, 60, 90, 120 Hz
Bit depth	8, 10

7.2.4 Encoding and Decoding Constraints

Table 7.2.4-1 provides an overview of encoding and decoding constraints for UE-to-UE Stereoscopic Video Live Streaming scenario using H.265/HEVC and MV-HEVC. This information supports the definition of detailed anchor conditions.

Table 7.2.4-1 Encoding and Decoding Constraints

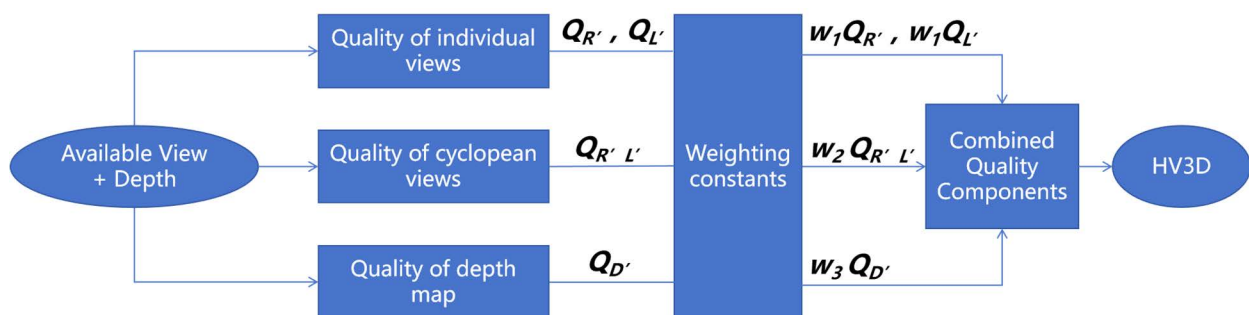
Encoding and Decoding Constraints	H.265/HEVC	MV-HEVC
Relevant Codec and Codec Profile/Levels	H.265/HEVC Main 10 Profile Level 4.1, 5.1	Multiview Main or Multiview Main10 profile Level 4, 5.1 and higher
Random access frequency	1 second	1 second
Bit rates and quality configuration	Fixed QP: [17~37] CBR Half Width/Height: 5-8Mbps Full Width/Height: 8-16Mbps Capped-VBR	Fixed QP: [17~ 37]
Bit rate parameters (CBR, VBR, CAE, HRD parameters)	Covering a range of relevant bitrates and qualities	Covering a range of relevant bitrates and qualities
Latency requirements and specific encoding settings	Low latency requirements	Low latency requirements
Encoding complexity context	Real-time encoding, Cloud-based encoding	Real-time encoding, Cloud-based encoding
Required decoding capabilities	H.265/HEVC Main 10 Profile Level 4.1, 5.1	Multiview Main or Multiview Main10 profile Level 4, 5.1 and higher

7.2.5 Performance Metrics

7.2.5.1 Objective Metrics for Captured Stereoscopic Video

Objective evaluation of stereoscopic video can be conducted by applying 2D video quality metrics (e.g., PSNR, SSIM, SNR...) to each source view position and computing their average.

A full-referenced human visual-system-based quality metric for stereoscopic videos called HV3D had been proposed [102, 103]. As shown in Figure 7.2.5.1-1, HV3D metric takes into account the quality of individual views, the quality of the cyclopean view (fusion of the right and left view, what the viewer perceives), as well as the quality of the depth information. The HV3D quality metric is taking values between 0 and 1 (because it is optimized to be correlated with MOS/10), and higher than 1 in case quality is improved.

**Figure 7.2.5.1-1 HV3D Flowchart**

7.2.5.1.1 Quality of individual views

The metrics are computed for the quality of individual views that form the stereo pair. The quality of the distorted right view with respect to its matching reference view is calculated as followings. The quality of the left view is calculated in the same fashion.

$$w_1 Q_{R'} = w_1 VIF(Y_{R'}, Y_{R'}) + w_4 VIF(U_{R'}, U_{R'}) + w_4 VIF(V_{R'}, V_{R'})$$

Where Y_R and $Y_{R'}$ are luma information of the reference and distorted right views respectively, U_R and V_R are the chroma information of the reference right-view, $U_{R'}$ and $V_{R'}$ are the chroma information of the distorted right-view, w_l and w_4 are weighting constants.

7.2.5.1.2 Quality of cyclopean view

A 3D-DCT transform is then applied to each pair of matching blocks, generating two 16×16 DCT-blocks containing the DCT coefficients of the fused blocks. Given the human visual system's sensitivity to contrast, a 16×16 Contrast Sensitivity Function (CSF) modeling mask is applied to the 16×16 DCT block, assigning greater weights to frequencies that are more perceptually significant. This process is illustrated as follows:

$$XC = \sum_{i=1}^{16} \sum_{j=1}^{16} C_{i,j} X_{i,j}$$

Where XC is the cyclopean-view model for a pair of matching blocks in the right and left views, $X_{i,j}$ are the low frequency 3D-DCT coefficients of the fused view, i and j are the horizontal and vertical indices of coefficients, and $C_{i,j}$ is the CSF modeling mask.

Once the cyclopean-view model for all the blocks within the distorted and reference stereoscopic views is obtained, the quality of the cyclopean view is calculated as follows:

$$Q_{R'L'} = VIF(D, D')^\beta \sum_{i=1}^N \frac{SSIM(IDCT(XC_i), IDCT(XC'_i))}{N}$$

7.2.5.1.3 Quality of depth maps

The quality of depth information plays an important role in the perceptual of the stereoscopic video content. In HV3D, the quality of depth map is chosen to be variance-dependent, and it is formulated as follows:

$$Q_{D'} = VIF(D, D')^\beta \sum_{i=1}^N \frac{\sigma_{d_i}^2}{N \cdot \max(\sigma_{d_i}^2 | i = 1, 2, \dots, N)}$$

Where σ_{d_i} is the variance of block i in the depth map of the 3D reference view, β is a constant = 0.7 and N is the total number of blocks. The local disparity variance, $\sigma_{d_i}^2$, is defined as follows:

$$\sigma_{d_i}^2 = \frac{1}{64 \times 64 - 1} \sum_{k,l=1}^{64} (M_d - R_{k,l})^2$$

Where M_d is the mean of the depth value of each 64×64 block in the normalized reference depth map. The reference depth map is normalized with respect to its maximum value per frame, ranging between 0 and 1. $R_{k,l}$ is the depth value of pixel (k, l) in the outer 64×64 block within the normalized reference depth map.

7.2.5.1.4 Weighting constants

Weighting constants are found using least mean square technique such that the difference between the HV3D metric values and the MOS values was minimized for the training video set.

$$\min_{w_{i,i=1,2,3,4}} \{ || HV3D - MOS ||^2 \}$$

In study [102], and experiment has been conducted to to determine the best values for the weighting constants w_1 , w_2 , w_3 and w_4 which result in the minimum mean of square errors between HV3D index and the MOS. Table 7.2.5.1.4-1 shows the resulting values for these constants.

Table 7.2.5.1.4-1 Weighting Constants

w_1	w_2	w_3	w_4
-------	-------	-------	-------

0.14	0.1208	0.05	0.1353
------	--------	------	--------

7.2.5.2 Objective Metrics for Generated Stereoscopic Video

Current AI-based stereoscopic video generation methods often experience artifacts such as edge sharpness mismatch, cardboarding effects and crosstalk. These artifacts are particularly prominent along foreground objective edges [18], as shown in Figure 7.2.5.2-1. Therefore the objective metric for generated stereoscopic video emphasize edge region evaluation.

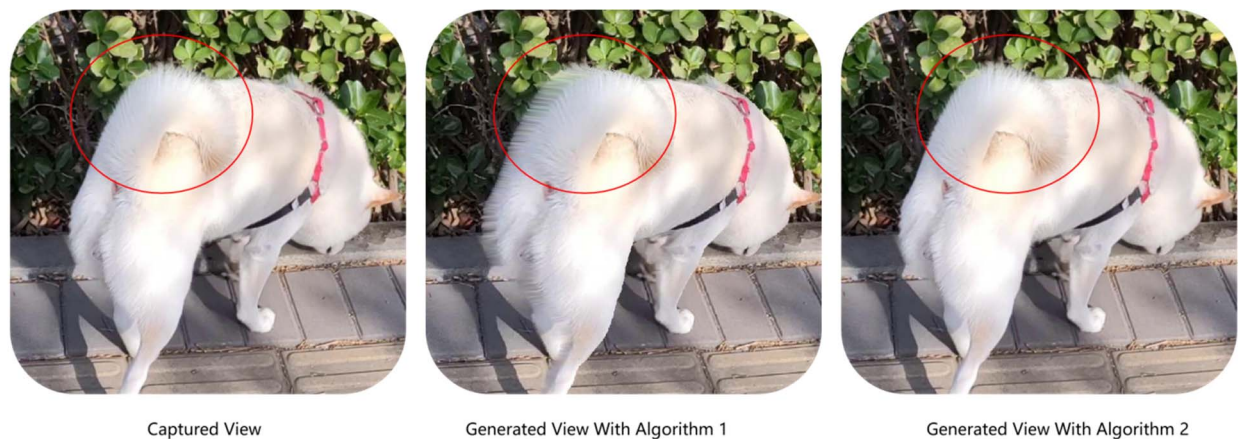


Figure 7.2.5.2-1 Example of AI-generated stereoscopic video

The objective evaluation of generated stereoscopic video sequences generally involves the following procedures:

Predicted Image and Depth Map Acquisition:

- It first takes a source monocular image as input into a stereo video generation model (e.g., AI-based algorithm) to generate the predicted image. For example, if the source image is a left-view image, the model generates the right-view image, and vice versa.
- After generation, a depth estimation algorithm to obtain the depth map of the predicted image.

Depth Map Processing and Edge Expansion:

- The depth values in the predicted image's depth map are divided into N ranges (where N is a positive integer)
- Edge detection and pre-processing such as filtering are the applied to extract vertical edge lines from the depth map (As shown in Figure 7.2.5.2-2).

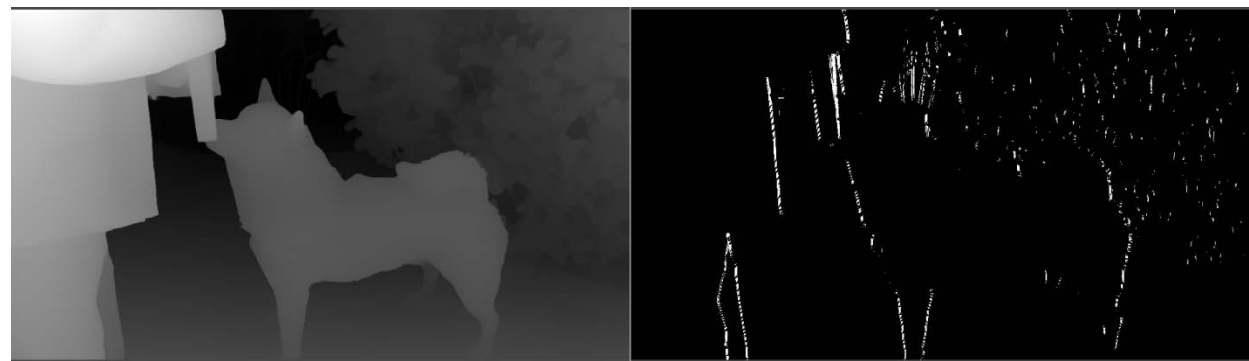


Figure 7.2.5.2-2 Example of Edge detection in depth maps

- The expansion ratio for each vertical edge line is determined based on the depth region it belongs to. Regions with greater depth values (i.e., farther from the viewer) are assigned smaller expansion ratios, while regions closer to the viewer has larger deformation, reflecting the greater disparity typically found in close objects.

- Expand the vertical edge lines horizontally according to their respective expansion ratios, and finally obtaining the segmentation mask (Figure 7.2.5.2-3) of the edge region in the predicted image's depth map.

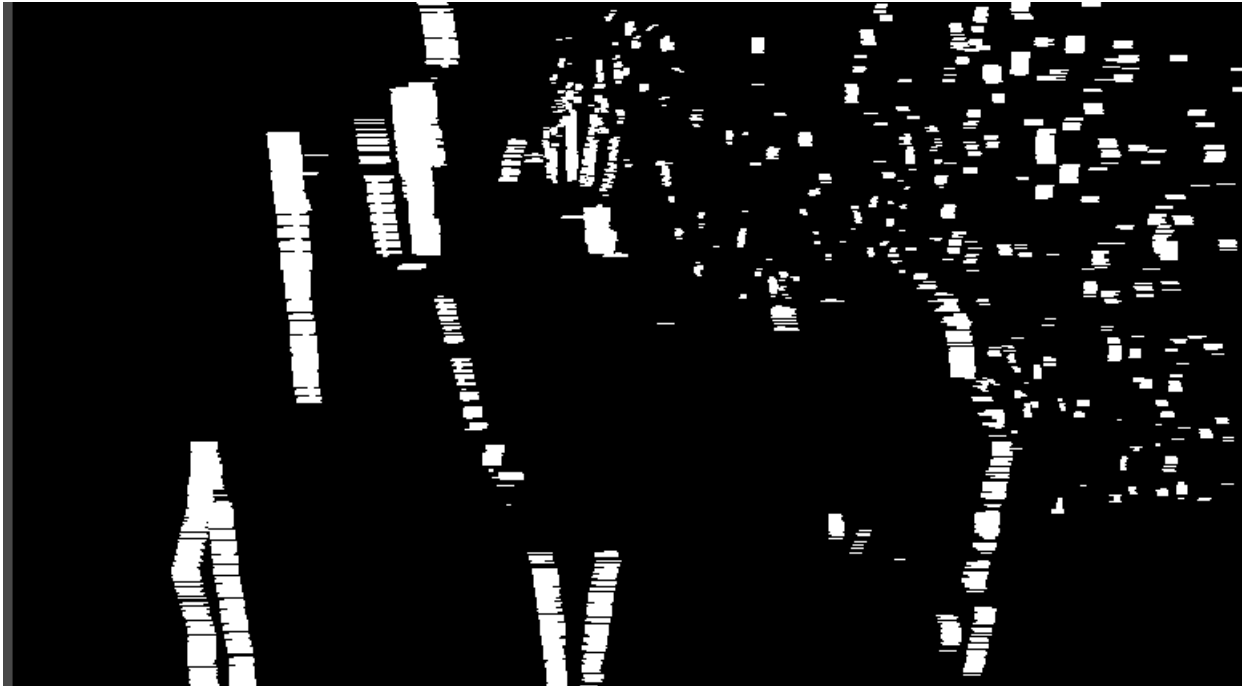


Figure 7.2.5.2-3 Example of segmentation mask

Evaluation Function Calculation

- Obtain the edge region in the predict image based on the segmentation mask.
- Calculate the first sub-evaluation function by comparing the edge region of the predicted image with the edge region of the Ground Truth Image.
- Calculate the second sub-evaluation function by comparing the non-edge regions of the predicted image (i.e., regions excluding the second edge region) with the non-edge regions of the Ground Truth Image.

The final evaluation function Q is calculated through the following weighted summation formula:

$$Q = \alpha \cdot \text{subQ1} + \beta \cdot \text{subQ2}$$

Here, α and β represent the weights assigned to the first and second sub-evaluation functions, respectively. Since artifacts are significantly more pronounced in edge regions compared to non-edge regions, α is significantly smaller than β . For example, α and β may be assigned values of 0.2 and 0.8, respectively.

7.2.5.3 Subjective Evaluation

According to [103], the viewing conditions for subjective tests were set according to the ITU-R Recommendation BT.500-13 [104]. The evaluation was performed using a 46" Full HD Hyundai 3D TV (Model: S465D) with passive glasses. The peak luminance of the screen was set at 120 cd/m2 and the color temperature was set at 6500K according to MPEG recommendations for the subjective evaluation of the proposals submitted in response to the 3D Video Coding Call for Proposals [109]. The wall behind the monitor was illuminated with a uniform light source (not directly hitting the viewers) with the light level less than 5 % of the monitor peak luminance.

A total of 88 subjects participated in the subjective test sessions, ranging from 21 to 32 years old. All subjects had none to marginal 3D image and video viewing experience. They were all screened for color blindness (using Ishihara chart), visual acuity (using Snellen charts), and stereovision acuity (via Randot test– graded circle test 100 seconds of arc). Subjective evaluations were performed on both training and validation data sets.

Test session started after a short training session, where subjects became familiar with video distortions, the ranking scheme, and test procedure. Test sessions were set up using the single stimulus (SS) method where videos with different qualities were shown to the subjects in random order (and in a different random sequence for each observer). Each test

video was 10 seconds long and a four-second gray interval was provided between test videos to allow the viewers to rate the perceptual quality of the content and relax their eyes before watching the next video. There were 11 discrete quality levels (0-10) for ranking the videos, where score 10 indicated the highest quality and 0 indicated the lowest quality. Here, the perceptual quality reflects whether the displayed scene looks pleasant in general. In particular, subjects were asked to rate a combination of “naturalness”, “depth impression” and “comfort” as suggested by [110]. After collecting the experimental results, we removed the outliers from the experiments (there were seven outliers) and then the mean opinion scores (MOS) from the remaining viewers were calculated. Outlier detection was performed in accordance to ITU-R BT.500-13, Annex 2 [110].

7.2.5.4 Correlation between the objective and subjective metrics

The performance of the objective metric, HV3D, described in section 7.2.5.1 is validated by subjective test described in section 7.2.5.2, using 88 subjects, following the ITU-R BT.500-13 recommendation. Performance evaluation results showed that HV3D quality metric quantifies quality degradation caused by several representative types of distortions very accurately, with Pearson correlation coefficient of 90.8 % [109].

7.2.6 Interoperability Consideration

For UE-to-UE Stereoscopic Live Streaming, DASH-based solutions are expected.

7.2.7 Reference Sequences

7.2.7.1 Candidate Source Stereoscopic 3D Video Sequences

This section introduces candidate source stereoscopic 3D video sequences available for testing. Some sequences are freely available under a license agreement, while others are self-generated using AI-based 2D-to-3D conversion algorithms. Additionally, some sequences are self-captured using commercially available mobile devices.

7.2.7.1.1 Public Datasets

InStereo2K [111], an indoor real scene stereo dataset. It contains 2000 pairs of images with high accuracy disparity maps. The author kindly grants the permission to use this dataset under the license in 3GPP.

Middlebury 2021 mobile datasets [112], 24 datasets obtained with a mobile device on a robot arm, using the technique described in [112]. The authors kindly grant permission to use and publish all images and disparity maps on their website. However, they request to cite the appropriate paper [112] for using 2021 datasets.

Middlebury 2014 stereo datasets [112], 33 datasets obtained using the technique described in [112]. The authors kindly grant permission to use and publish all images and disparity maps on their website. However, they request to cite the appropriate paper [112] for using 2014 datasets.

The table below summarizes the characteristics of the sequences in the public datasets.

Table 7.2.7.1.1-1 Test material datasets

Name					Color Space	
InStereo2K	Natural	Y	Y	1080 × 860	BT.709 SDR	https://github.com/YuhuaXu/StereoDataset
Middlebury 2021 Mobile stereo datasets	Natural	Y	Y	1920 × 1080	BT.709 SDR	https://vision.middlebury.edu/stereo/data/scenes2021/
Middlebury 2014 stereo datasets	Natural	Y	Y	2964 × 1988	BT.709 SDR	https://vision.middlebury.edu

						u/stereo/data/scenes2014/
--	--	--	--	--	--	---

7.2.7.1.2 Self-Converted Sequences

The AI-based conversion of existing 2D images and Video to stereo3D is proving commercially viable and fulfills the growing need for high quality stereoscopic images. This approach is particularly effective when creating content for the new generation of autostereoscopic displays that require multiple stereo images. Various open-source algorithms and platforms use deep neural networks to perform real-time end-to-end conversion of 2D videos and images to stereoscopic 3D video format.

As 2D-to-Stereo3D conversion algorithms usually take RGB video format, the Python scripts can be found in Annex D.2.2 to convert between YUV and RGB formats.

The test sequences use the left view of the stereoscopic videos collected in Section 7.2.7.1.3 as input, and generate the right view through AI algorithms to synthesize side-by-side stereoscopic videos. The sequences can be found in Annex C.3.5, C.3.6, and C.3.7.

7.2.7.1.3 Self-Captured Sequences

A dual-lens camera can be used to directly capture stereo 3D video. There are many mobile devices on the market with this capability. For example, SpatialLabs Eyes™ provided by Acer is a stereoscopic camera capable of capturing up to 8-MP (4K) per eye at 30 fps or 2K per eye at 60 fps. Or the ZTE Nubia Pad 3D II™ can capture stereo 3D video with the specifications in Table 7.2.7.1.3-1:

The main camera setup is the dual-camera systems includes two identical 13 MP lenses. These cameras capture slightly different perspectives of the same scene, mimicking the way human eyes perceive depth. The AI then processes these images to produced a coherent 3D representation. The selfie camera setup features two lenses positioned near the center of the top bezel when the tablet is oriented horizontally (with the longer side on the top).

Table 7.2.7.1.3-1 Specification of Capturing Device

Number of Cameras	2 (Dual)
Resolution	13 MP (wide); 13 MP (wide)
Autofocus	AF, AF
Video Recording	1200 @ 30 fps
Others	LED Flash, panorama, HDR, Stereoscopic AI-powered 3D capture
Number of Cameras	2 (Dual)
Resolution	8 MP (ultra wide); 8 MP (ultra wide)
Aperture	f/2.2, f/2.2
Field of View	105°, 105°
Video Recording	1200 @ 30 fps

The captured videos (in mp4 file) need further processing (e.g., reading the right/left views and concatenate them into one video frame) the scripts can be found in Annex D.2.3. An FFmpeg command described in Annex D.2.4 can be used to save each frame into a proper test sequence,

The test sequences captured from ZTE Nubia Pad 3D II™ main camera and post-processed by the above tools can be found in Annex C.3.2, C.3.3, and C.3.4.

7.2.8 Test Condition

7.2.8.1 Test model and configuration files

The encoder configuration settings for both encodings are outlined below:

- Inter-view coding structure
 - 2 view case: left-right (in coding order)
 - I-P inter-view prediction MV-HEVC
- Temporal prediction structure: GOP 8, intra every 24 frames (random access at ~1sec)
- Full resolution texture coding
- Codec software: HTM v16.3 for Simulcast HEVC and MV-HEVC

The following configuration files are provided in [113]:

- /HTM-16.3-fixed/cfg/MV-HEVC/baseCfg_2view.cfg: Used to configure input/output filenames and encoder parameters (I-frame interval, number of B-frames, etc.)
- /HTM-16.3-fixed/cfg/MV-HEVC/qpCfg_QP25.cfg: : Used to configure the encoding QP
- /HTM-16.3-fixed/cfg/MV-HEVC/seqCfg_Shark.cfg: Contains the source sequence parameters (resolution, frame count, frame rate, etc.)

For each selected test sequence, configuration files containing information needed for HTM-16.3 configuration will be provided.

7.2.8.2 Rate points and test conditions

Fixed QP was used to evaluate and compare the performance of MV-HEVC and Simulcast HEVC is compared and evaluated in terms of PSNR (dB) and bitrate (kbps) over the set of QP values [17, 22, 27, 32, 37].

7.2.8.3 Profiles

MV-HEVC Main Profile is used.

7.2.8.4 Bitstream Generation

The HTM v16.3 is used to encode and decode test sequences as described in clause 7.2.8.1.

Below are examples of command lines for encoding and decoding test sequence:

- Before compilation, navigate to `source/Lib/TLibCommon/TypeDef.h` and modify the following parameter to configure the software as an MV-HEVC encoder:

```

/** \file      TypeDef.h
    \brief      Define macros, basic types, new types and enumerations
 */
(...)
/* HEVC_EXT might be defined by compiler/makefile options.
   Linux makefiles support the following settings:
   make          -> HEVC_EXT not defined
   make HEVC_EXT=0 -> NH_MV=0 H_3D=0    --> plain HM
   make HEVC_EXT=1 -> NH_MV=1 H_3D=0    --> MV only
   make HEVC_EXT=2 -> NH_MV=1 H_3D=1    --> full 3D
 */
#ifndef HEVC_EXT
#define HEVC_EXT 1

```

```
#endif
(...)
```

- to encode a test sequence:

```
./TAppEncoder.exe -c baseCfg_2view.cfg -c seqCfg_Shark.cfg -c qpCfg_QP25.cfg
```

- to decode a test sequence:

```
./TAppDecoder.exe -b stream.bit -o shark_qp25.yuv -w 1
```

7.2.9 External Performance data

The verification test report for MV-HEVC can be downloaded from the JCT-3V website [114]

The VQEG 3DTV Group conducted an evaluation of video quality models for stereoscopic 3D television content. In collaboration with the DVB, they further assessed the visual impact of Side-by-Side, Top-Bottom, and Tile formats on Quality of Experience (QoE) for Full-HD television transmissions across varying bitrates [115]. The findings from this evaluation played a key role in the development of ITU-T Recommendations P.914 [116], P.915 [117], and P.916 [118].

Two recent research papers [119][120] evaluated end-to-end stereoscopic 3D live streaming using the iPhone 15TM and the Apple Vision ProTM XR headset, the key observations are summarized below:

- **Bandwidth Requirements**
 - Apple TV+TM's immersive 180° stereoscopic video demands ~99.81 Mbps at peak quality (4320×4320, 90fps), challenging LTE/5G uplink.
 - Vision ProTM's spatial video requires 31 Mbps, while iPhone 15 ProTM uses 16.89 Mbps, comparable to YouTubeTM's 4K (35-45 Mbps) and 2K (16 Mbps) streams.
 - iPhone 15 ProTM's spatial videos consume about twice the bitrate of conventional iPhone 11 ProTM videos.
- **Impact of network artifacts on depth perception:**
 - According to [119], lower bitrates degrade disparity map quality, with a 3.5 dB PSNR drop when reducing bitrates from 300 Mbps to 1 Mbps, along with noticeable object dislocation in the scene, creating unnatural depth conflicts under poor network conditions.
- **Content-Related Impacts on User Experience**
 - Stereoscopic videos generally provided a better viewing experience than monoscopic videos at the same bitrate. However, some exceptions occurred: close-up scenes sometimes caused visual discomfort, and high-motion scenes were rated lower across both formats due to motion sickness. Scenes with strong depth cues and occlusions were particularly well-suited to stereoscopic presentation, receiving higher user ratings.

7.2.10 Additional information

The industry has increasingly embraced stereoscopic 3D video, as demonstrated by recent advancements in application support:

- Dolby VisionTM Profile 20 extends Dolby Vision's quality enhancements to MV-HEVC and immersive content: https://professionalsupport.dolby.com/s/article/Dolby-Vision-Profile-20-FAQ?language=en_US
- Apple Vision ProTM Spatial Video Formats (a stereo MV-HEVC video track, plus spatial metadata): <https://developer.apple.com/av-foundation/HEVC-Stereo-Video-Profile.pdf>
- NVIDIATM's Video Codec SDK 13.0 introduces an MV-HEVC encoder, further supporting stereoscopic 3D video: <https://developer.nvidia.com/blog/enabling-stereoscopic-and-3d-views-using-mv-hevc-in-nvidia-video-codec-sdk-13-0/>

7.3 Scenario 2: Streaming of professionally produced Volumetric Video with single asset containing people

7.3.1 Scenario name

Streaming of professionally produced Volumetric Video with single asset containing people.

7.3.2 Motivation for the scenario

The scenario addresses on-demand streaming of post-produced volumetric video with single asset containing people, providing experiences beyond what is achievable with 2D content. Particularly in AR application, the user can watch the volumetric video asset from all directions as if the asset were naturally present. In both AR and VR applications, the user can move smoothly around the asset, change its size or make the asset rotate.

The content is in the form of volumetric video, which is a frame-based immersive experience whereby each frame represents a volumetric region in 3D space in which any point is either non-occupied or has a colour that may depend on the viewing direction. Volumetric video has the potential to provide a more immersive and interactive experience than 2D content.

Several use cases for on-demand volumetric video streaming can be envisioned related to various domains including but not limited to entertainment, education and industrial monitoring. For example, in the entertainment domain users can stream a performance from their favorite singer or band to their living room and experience greater immersion potentially combined with spatial audio. Another example is the education/training domain, where a produced volumetric video of a fitness instructor shows how to perform an exercise that helps a student to better understand how the exercise is done and thus replicate it in a correct way. Yet another example in the education domain would be a mechanic giving a tutorial on how to assemble a mountain bike. A trainee can watch the movements of the mechanic from different angles and get an improved understanding of the different steps due to depth perception and different viewpoints.

For first implementations of relevant use cases the content can be quite simple without hindering the purpose, consisting of a camera captured 6 DoF person as asset and a 3D graphics background or an AR camera background coming from the rendering device. As an alternative a person could be captured with an object (e.g. a ball) or two persons together. Important is that everything is in a single asset.

A. *Technology evaluation on the market*

KDDI experimented transmission of produced volumetric video over mobile networks including real time encoding and decoding [121].

Volucap, based in Potsdam Germany, developed several showcases and tested these on the market. The number of showcases is growing and the current list can be consulted here: <https://volucap.com/showcases/>.

The following showcases illustrate particularly the proposed scenario:

- Sports training: Volucap and Deutsche Telekom produced a clip to learn cool dribbles and precise throws from the former basketball star Josh Mayo: <https://volucap.com/portfolio-items/meeting-josh/>
- Music Group: Volucap and the music group “Boss Hoss” prototyped the your favorite band in your living room: <https://volucap.com/portfolio-items/the-bosshoss-augmented-reality/>
- Tagesschau: Volucap and the German news broadcaster Tagesschau collaborated to capture a volumetric representation of a news: <https://volucap.com/portfolio-items/tagesschau-2025/>
- Book enhanced with AR: Volucap enhanced a children song book with AR content on a smartphone: <https://volucap.com/portfolio-items/rolf-zuckowski/>
- XR Fashion show: Volucap and Lana Mueller, a Berlin based designer, produced a fashion presentation in XR: <https://volucap.com/portfolio-items/lana-mueller-fashion/>
- Charité medical VR training: This immersive simulation uses advanced [volumetric video technology](#) to bring users into the heart of a realistic surgical environment, ideal for [virtual reality \(VR\)](#) training scenarios: <https://volucap.com/portfolio-items/charite-medical-vr-training/>

B. *Industry activities*

Several industry activities regarding on-demand volumetric video streaming have been experimented between various mobile network operators, volumetric capture studios and technology providers. Some of these industry activities are listed here:

- Volograms, based in Dublin, Ireland
 - Provides professional volumetric content creation services to feed AR use cases such as augmented museum, training or fashion experiences: <https://www.volograms.com/made-with-volograms>
 - The company has also developed an AI based solution to enable AR volumetric content from 2D single photo or video: <https://www.volograms.com/>
- 8i, Mantis Vision, Metastage, Volograms, XD Productions, etc. present volumetric capturing projects on their websites, similar to Volucap
- XD Productions and Volograms content (both professional and AI-based) has been showcased in public trade shows and conferences by InterDigital as part of MPEG-I V3C platform demonstration with the V-PCC player
- Zerospace and Canon are collaborating to open a volumetric video capturing studio. With over 100 Canon Cinema EOS cameras, it claims to offer unmatched capabilities. The website illustrates capture of sports content (e.g. basketball, Karate): <https://www.zerospace.co/studios/canon-volumetric-capture>
- Brazilian SBTVD Forum has adopted volumetric video for inclusion in their [TV 3.0 standards](#) (support is not be mandatory in all receivers; focus is on content distribution over the Internet and consumption on smartphones and HMDs). TV 3.0 services are planned to be launched in 2025 [122]
- The Volumetric Format Association has been formed to ensure interoperability of volumetric video capturing, processing, compression and playback [123]

C. *Production tools/companies*

The following is a non-exhaustive list of companies providing tools, equipment or services to produce volumetric video content:

- 4D People: Renderpeople is a company that provides large libraries of different kinds of scanned people in 3D and 4D, where 4D includes the dimension of time and corresponds to Volumetric Video: <https://4dpeople.com/>
- 4D Views: 4D Views is a company that provides a volumetric capturing system named Holosys including HW, SW, support and editing software. A volumetric player is provided to review volumetric demo sequences: <https://www.4dviews.com/>
- 8i: Provides a volumetric capturing system including HW, SW and a solution to stream content to devices, browsers and HR/VR headsets: <https://8i.com/>
- Arcturus: Provides volumetric capturing services and commercializes a volumetric capturing system named HoloCapture and a volumetric video post-production and streaming solution named HoloSuite. A number of volumetric production studios are installed around the world such as Dimension in London, Metastage in Los Angeles, ifland Studio in Seoul (SK Telecom), Nikon Creates Corporation in Tokyo and at ETH in Zurich. See more information: <https://arcturus.studio/>
- CIVIT: <https://civit.fi/volumetric-capture-studio/>
- Dimension Studio: <https://dimensionstudio.co/>
- Evercoast : <https://evercoast.com/>
- Mantis Vision : <https://mantis-vision.com/>
- Metastage : <https://metastage.com/>
- Volucap: Volucap claims to have the world's highest resolution for capturing volumetric video with 700 megapixels for each shot and a unique lighting system: <https://volucap.com/>
- XD Prod : <https://www.xdprod.com/>

Capturing systems typically integrate with 3D editing tools such as Maya, Blender for post producing content.

D. Delivery solutions

Arcturus provides an on-demand adaptive streaming solution for volumetric video [124]

8i provides a solution to stream volumetric video produced with their capturing system [125]

An end-to-end implementation of a MPEG V3C standardized platform for packaging and delivery of volumetric video over content delivery network (CDN) is available [126]

E. Content decoding and rendering

Arcturus provides a play-back solution with broad delivery capabilities including support for Unreal, Unity, iOS and Android [127].

8i provides a solution to decode and render volumetric video on a browser running on CE devices and AR/VR headsets [125].

5G-MAG hosts a V3C Immersive Platform [128]. It provides a Unity package to decode, render and play V3C content in Unity using the V3C Immersive Platform – Decoder Plugin.

7.3.3 Description of the scenario

This scenario covers Streaming of professionally produced Volumetric Video with single asset containing people.

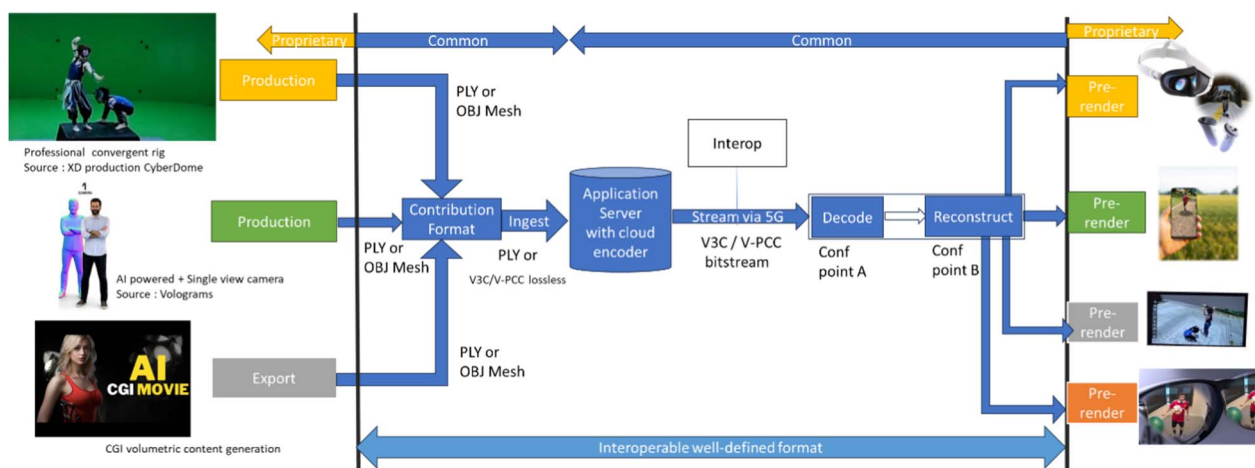


Figure 7.3.3-1: Streaming of professionally produced Volumetric Video with single asset containing people (content courtesy XD Productions)

A. Capturing and processing

Capturing of high-quality 6 DoF assets as a volumetric video is typically done with a rig of cameras aligned on a circle around the asset(s) to be captured. Depending on the rig, there can be one or more layers of cameras at different height positions. The number of cameras per layer depend again on the designer of the rig and in the year 2024 there are typically between 30 and 100 cameras per layer. Cameras can be equipped with depth sensors. Hardware such as cameras and depth sensors are mostly off the shelf equipment, but the assembly in the rig is vendor dependent and proprietary.

The various camera and depth sensor signals are fed into a production pipeline that produces the volumetric video. Production includes stitching the various signals, filling holes, correcting occlusions, etc. Persons or physical objects (e.g. a ball or an instrument) can be combined in an asset or separate assets can be used for each person or object. For simplification and not hindering the purpose, the use case described in this document is limited to a single asset. The representation format of a produced asset is typically a dense dynamic point cloud or a dynamic mesh.

As an example, in the following the production pipeline of the company XD Productions is illustrated.

The figures below show the XD Productions CYBERDOME capture rig and associated real time viewing to control acquisition.

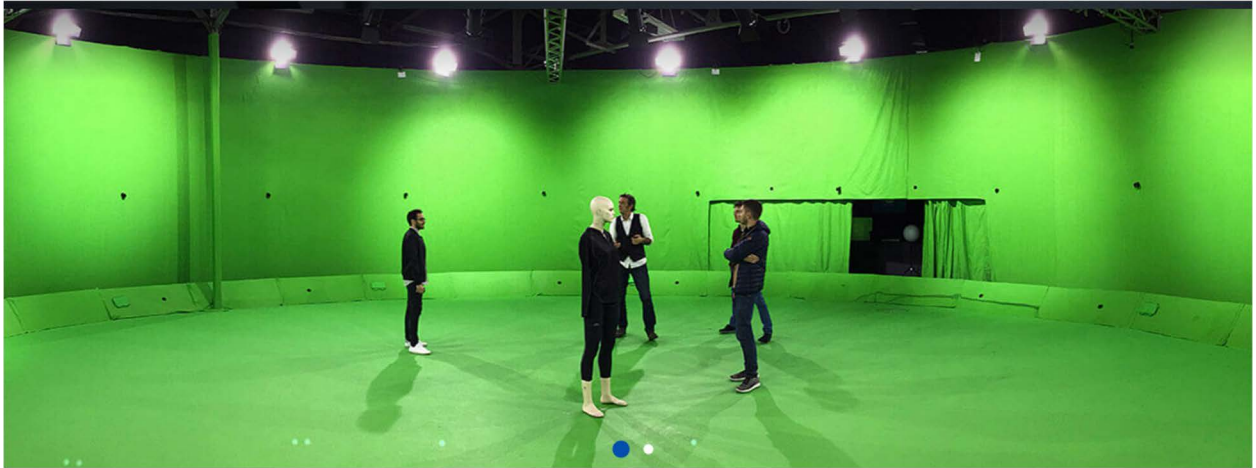


Figure 7.3.3-2: XD Productions capture rig (<https://www.xdprod.com/services/studio/studio-virtuel/>)



Figure 7.3.3-3: XD Production real time virtual production
(<https://www.xdprod.com/services/studio/studio-virtuel/>)

The figures below show CYBERDOME acquisitions covering single or multiple characters in dynamic scenes.



Figure 7.3.3-4: XD Productions contents screenshots, from top right to left: Acrobat01, Soccer Blue, Soccer Red, Dancer01 and Acrobat Duo

The acquisition processing pipeline includes a rig of about sixty 4K cameras, arranged in hemispheres around the scene to be captured. The set is 15-meter in diameter for a 7-meter diameter capture area. Two types of lenses are simultaneously used, with variable focal lengths, which allows to adapt the size of the capture area, and to mix wide shots and close-ups on the same captures to improve the quality of the textures. Each content item is then converted into point cloud frames. The processing output is provided in the PLY format.

Another example on how single asset volumetric video is produced is shown in a [video](https://www.youtube.com/watch?v=xX4SJTE3hmQ) (<https://www.youtube.com/watch?v=xX4SJTE3hmQ>) by Metastage.

There are several companies that provide volumetric video capturing technology or entire volumetric video capturing studios. More detailed information can be found in chapter 5.4.4 of the [DVB Study Mission report S101 on Volumetric Video](#). [129]

The Volumetric Format Association [123] describes an end-to-end volumetric chain including capturing, processing, encoding, decoding and rendering. After capturing the processing generates a point cloud that can be encoded and delivered as dense dynamic point cloud and reconstructed either as a point cloud or as a mesh for rendering.

The Ultra Video Group of Tampere University describes the process of generating volumetric video content with the Mantis Vision system in their paper [130]. In a first step from the signals from multiple camera units on the rig a raw point cloud of the person or object is generated. In a second step a mesh is generated by using surface reconstruction algorithms. In a third step the mesh is sampled to generate a point cloud, which can then be voxelized on a regular 3D grid and normal can be calculated if needed.

It can be concluded that dense dynamic point clouds and dynamic meshes are used by industry to represent persons or objects as volumetric video.

In the following it is described how the scenario can be implemented with MPEG V-PCC [131] by using dense dynamic point clouds and with MPEG V-DMC [55] by using dynamic meshes for representing the single asset.

The following figure shows the V-DMC workflow:

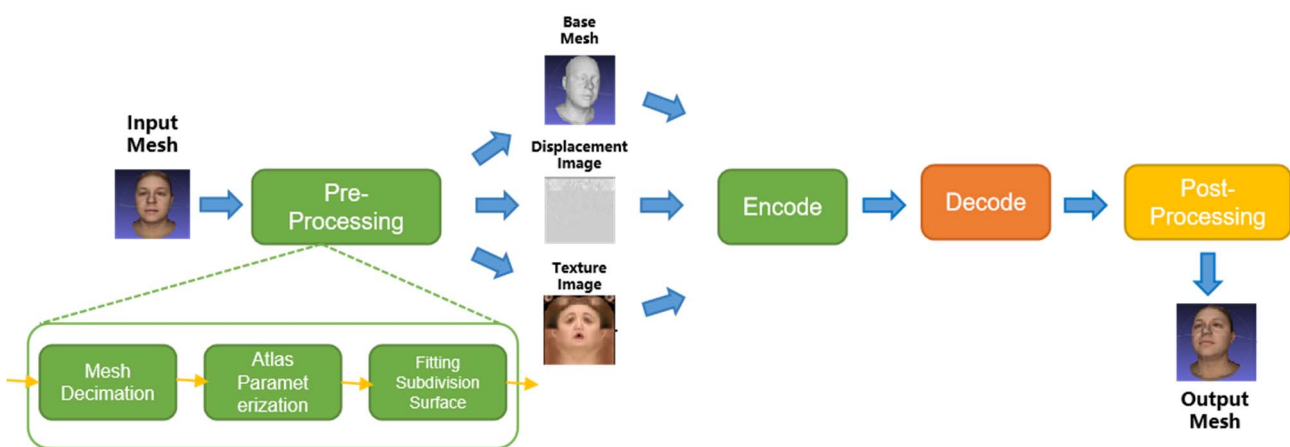


Figure 7.3.3-5 V-DMC Workflow

The pre-processing stage consists of three main processes: a decimation step, an atlas parameterization step, and a subdivision step.

B. Encoding

Volumetric video can be represented in the representation format dense dynamic point clouds. MPEG has developed a specification named V-PCC for compressing and delivering the representation format dense dynamic point clouds at bitrates enabling consumer applications. V-PCC is standardized in ISO/IEC 23090-5 Visual Volumetric Video-based Coding (V3C) and Video-based Point Cloud Compression (V-PCC) [131].

During its experimentation with V-PCC KDDI implemented a real time V-PCC encoder [121].

The following figure shows the V-PCC encoder main steps.

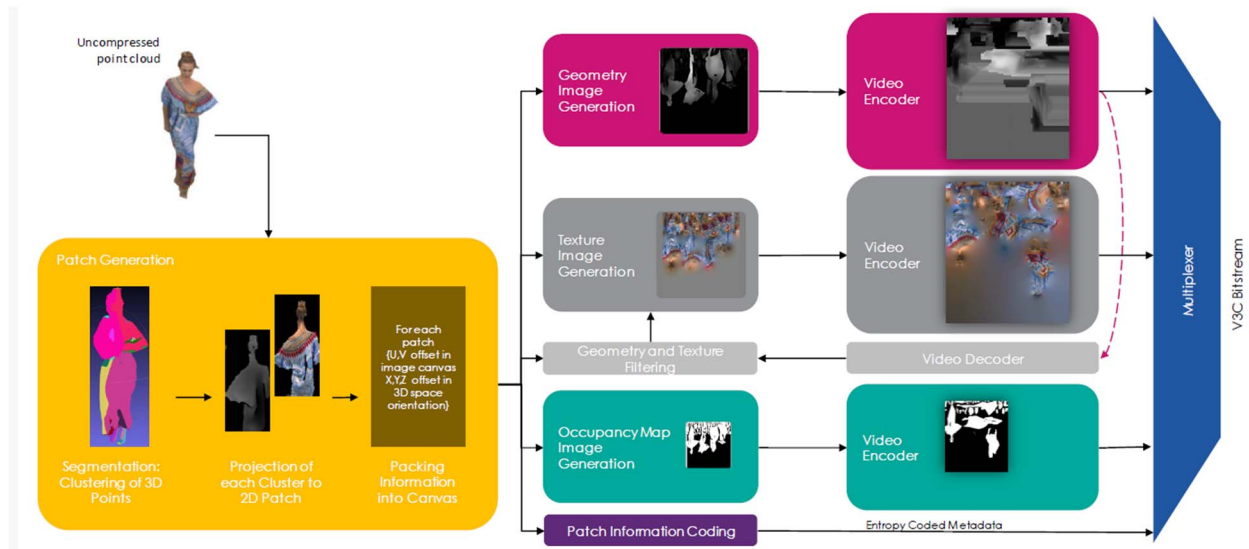


Figure 7.3.3-6 V-PCC encoder main steps (Content courtesy 8i)

For encoding of geometry, texture and occupancy map V-PCC relies on 2D video codecs. Due to its efficiency and market penetration HEVC aka H.265 is the choice of 2D video codec for the presented scenario.

Volumetric video can also be represented in the representation format dynamic mesh. MPEG has developed a specification named V-DMC for compressing and delivering the representation format dynamic at bitrates enabling consumer applications. V-DMC [55] is standardized in ISO/IEC 23090-29 Video-based dynamic mesh coding (V-DMC). As V-PCC, V-DMC relies on (HW-accelerated) video codecs for the bulk of the data (attribute maps, etc).

The following figure shows the V-DMC encoder main steps.

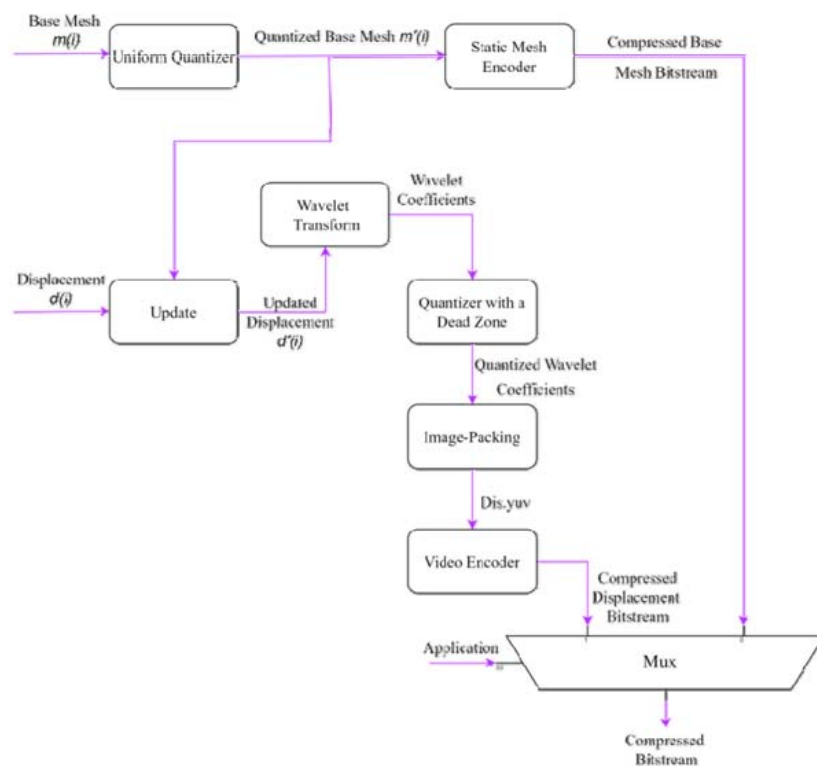


Figure 7.3.3-7 V-DMC encoder main steps [156]

C. Packaging and delivery

MPEG has developed a specification addressing storage and delivery V-PCC coded data published as ISO/IEC 23090-10 Carriage of visual volumetric video-based coding data [132].

As of July 2025, MPEG is working on the second edition of ISO/IEC 23090-10 that will cover V-DMC storage in ISOBMFF format and delivery utilizing DASH.

An overview of MPEG standards for storage and transport of V3C can be found in [154]

D. Decoding

For the dense dynamic point cloud representation format, decoding can rely on the V-PCC specification and for dynamic mesh it can rely on the V-DMC specification. In both cases no dedicated hardware is required for V-PCC real-time decoding on consumer devices. If the encoder uses HEVC for the 2D video encodes, the decoder in the device can make use of integrated hardware HEVC video decoder capabilities for all pixel data, and a small number of metadata is decoded by a CPU.

As an example, the following figure shows the architecture of a V-PCC decoder.

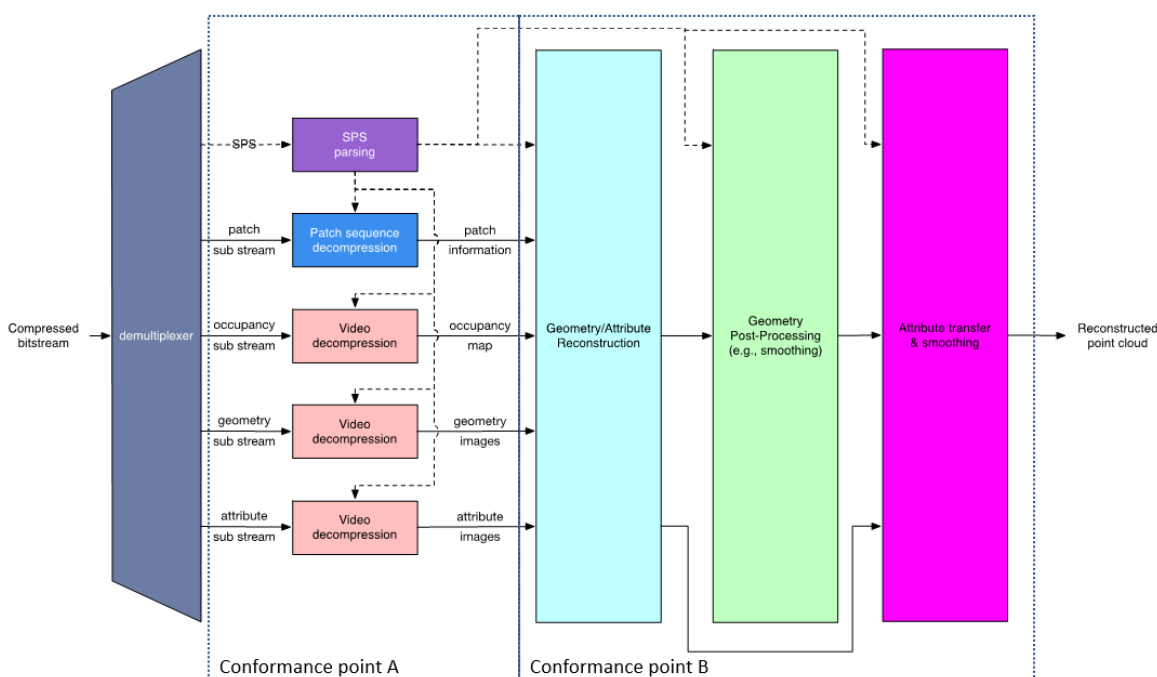


Figure 7.3-8 V-PCC decoder main steps

Decoding of the dense point cloud is terminated at the output of the video decoders, but these images are just intermediate results and do not represent a useable image for the human eye. Additional stages are needed to reconstruct the dense point cloud in 3D space and render it to the display of a consumer device. It can be rendered to 2D displays of e.g. smartphones and tablets, but also on head mounted devices or other 3D displays.

An example of point cloud data decoding processing has been described in [133].

During its experimentations with V-PCC and V-DMC, KDDI implemented a real time V-PCC encoder/decoder [121] and a V-DMC encoder/decoder [153].

Futuresource estimates that by the end of 2023 there are 4.1 billion Smartphones globally in the field with the capacity to decode HEVC video [134].

E. *Post-processing

Post-processing is implementation dependent, but it is typically performed on a GPU without dedicated V-PCC/V-DMC hardware.

F. Rendering

Rendering is implementation dependent, but it is typically performed on a GPU without dedicated V-PCC/V-DMC hardware. More information can be found in section 4.3.3.3 for dense dynamic point cloud and in section 4.3.5.3 for dynamic mesh.

G. General constraints on latency, bandwidth, reliability and complexity

For delivery, the volumetric video frames are organized using a random-access reference frame structure.

All decoder and renderer processes are real-time and may have a latency in the order of a few frames.

7.3.4 Source format properties

As source format for the scenario the representation format dense dynamic point clouds and dynamic mesh are considered. Section 4.3.3 describes the dense dynamic point cloud format and section 4.3.5 describes the dynamic mesh format. The present section provides more detail in direct relation with the scenario.

The following table includes signal properties that are typically used in the near/mid term to represent people or objects.

Table 7.3.4-1 Signal properties for dense point cloud format

Source format properties	Volumetric Video with single asset
Number of points /Spatial Resolution	Up to 2 million points per frame
Chroma format	RGB
Chroma subsampling	Not Applicable
Picture aspect ratio	Not Applicable
Frame rates	25, 30 Hz
Bit depth	8 and 10
Colour space formats	RGB 444 nonlinear, BT.709
Transfer characteristics	BT.709
Viewpoints	All assets can be viewed from all directions and different distances

Table 7.3.4-2 Signal properties for dynamic mesh format

Source format properties	Volumetric Video with single asset
Number of Polygons /Texture Map Resolution	30k polygons with 4k texture per frame
Chroma format	RGB
Chroma subsampling	Not Applicable
Picture aspect ratio	Not Applicable
Frame rates	25, 30 Hz
Bit depth	8 and 10
Colour space formats	RGB 444 nonlinear, BT.709
Transfer characteristics	BT.709
Viewpoints	All assets can be viewed from all directions and different distances

For highly dynamic sports sequences higher frame rates such as 50 or 60 fps may be useful and this is left for the future.

In the following the quality of the dense point cloud representation format for representing people is further investigated.

7.3.4.1 Conversion and quantization for dense dynamic point cloud format

If volumetric video source sequences are delivered in production quality, then bandwidth for consumer delivery may be too high and a conversion/quantization is necessary to reduce the number of points and use the fixed-point format. The result of that conversion is that the sequence is in a bounding box of 10, 11 or 12 bit which is named in the following as Vox10, Vox11 and Vox12.

- Vox 12 creates a quality that is close to the quality that comes out of the production system for most provided sequences. Emotional facial expressions are clearly visible and tissue structure of cloths is visible. See figures 7.3.4.5.1-1 and 7.3.4.5.2-1 below.

- Vox 11 creates a quality that allows viewing the sequence from a wider distance and it allows to zoom closer. Emotional facial expressions and patterns on cloths is visible. See figures 7.3.4.5.1-2 and 7.3.4.5.2-2 below.
- Vox 10 creates a quality allowing to view the sequence from a wider distance, but getting too close is less recommended. See figures 7.3.4.5.1-3 and 7.3.4.5.2-3 below.

7.3.4.2 Impact of rendering for dense dynamic point cloud format

It is referred to section 4.3.3.3 that covers rendering and display systems. Rendering is typically not covered by standards and allows manufacturers to differentiate. The referred section discusses hole filling, but also techniques such as local meshing could be applied.

7.3.4.3 Impact of the background

To avoid interference between the background and the test sequence, it is recommended to use a neutral background colour for codec evaluation. If a VR or AR background is used instead of a neutral background, potential artefacts in the sequence are less visible, as the human eye concentrates on the overall picture and not only on the sequence.

This means also that in a real service where a background is used, potential artefacts in a sequence are less visible.

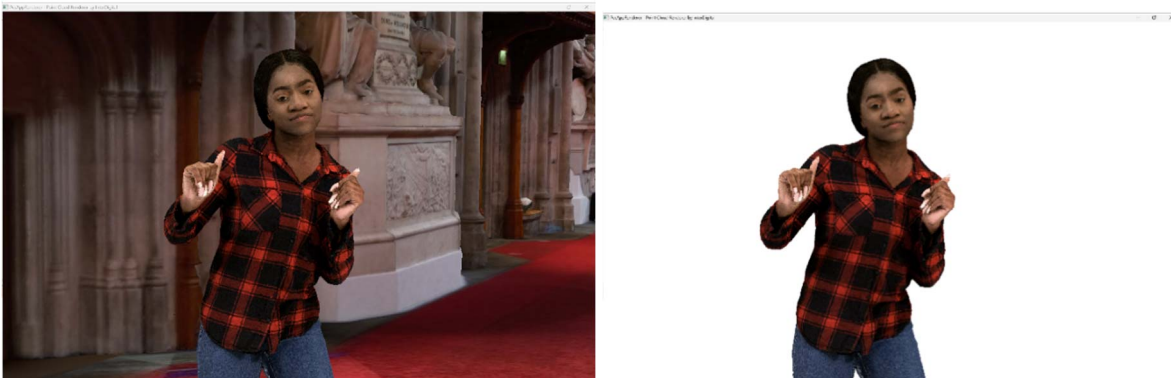


Figure 7.3.4.3-1 Impact of background on perceived quality (content courtesy RenderPeople [135])

7.3.4.5 Visual quality examples sequences in dense dynamic point cloud format

In the following the impact of number of points in a frame and type of renderer on the visual quality of the dense point cloud representation format is demonstrated. In this section still pictures are used to judge the preserved detail. In a volumetric video sequence some artefacts in the snapshots may not be visible.

It can be observed that sequences represented as dense point clouds of around 1M points/frame allow to watch a sequence with a person in AR from a wider distance (e.g. from 3m*) and point clouds of around 2M points/frame allow to get closer (e.g. to around 1.5m distance) at good quality for the target scenario. In the latter emotional facial expressions and tissue structure of cloths is visible. More points per frame improve the details, but this may not be required for the target scenario. But if a scenario would require it, a high-end volumetric video production system is able to capture details from e.g. skin or finer details of tissue and it can be represented with the point cloud representation format.

*A typical demonstration scenario would be to use e.g. a smartphone or tablet running a volumetric video application showing a real person of e.g. 3m distance on the screen captured by the camera and rendering at the same time a second person rendered from a point cloud next to the first person.

In the following the sequence Volucap_T003_ThomasScenic-03 from Volucap [136] is used.

7.3.4.5.1 Thomas near with representative renderer (splat blend mode) and neutral background



Figure 7.3.4.5.1-1 Vox 12 with 9.5M points per frame (content courtesy by Volucap [136])



Figure 7.3.4.5.1-2 Vox 11 with 2.3M points per frame (content courtesy by Volucap [136])



Figure 7.3.4.5.1-3 Vox 10 with 600K points per frame (content courtesy by Volucap [136])

7.3.4.5.2 Thomas near with representative renderer (cube mode) and neutral background



Figure 7.3.4.5.2-1 Vox 12 with 9.5M points per frame (content courtesy by Volucap [136])



Figure 7.3.4.5.2-2 Vox 11 with 2.3M points per frame (content courtesy by Volucap [136])



Figure 7.3.4.5.2-3 Vox 10 with 600K points per frame (content courtesy by Volucap [136])

7.3.5 Encoding and decoding constraints and settings

The following table provides an overview of encoding and decoding constraints for V-PCC with H.265/HEVC as codec for the Volumetric Video with single asset streaming scenario. Contribution aspects are not considered in this table.

Table 7.3.5-1 Encoding and decoding constraints

Encoding and Decoding Constraints	V-PCC with H.265/HEVC
Relevant Codec and Codec Profile/Levels	H.265/HEVC Main 10 Profile Level 4.1, 5.1 Metadata stream parsing
Random access frequency	1 seconds
Bit rates and quality configuration	Fixed QP Geometry see table D.3.4.1.1-1 Fixed QP Texture see table D.3.4.1.1-1 bitrates [1;50 Mbps]
Bit rate parameters (CBR, VBR, CAE, HRD parameters)	Covering a range of relevant bitrates and qualities
Latency requirements and specific encoding settings	No specific latency requirement
Encoding complexity context	Cloud-based encoding, offline encoding
Required decoding capabilities	3 decoder instantiations of H.265/HEVC Main 10 Profile Level 4.1, 5.1 for (occupancy, geometry and color) One synchronized metadata bitstream (Atlas)

7.3.6 Performance Metrics and Requirements

7.3.6.1 Anchors

There is no specification in 3GPP that references a volumetric video anchor codec suitable for the streaming single asset scenario. MPEG V-PCC [131] is the first codec supporting the dense dynamic point cloud representation format with inter coding and therefore no anchor codec for the format can be selected. Similar to V-PCC, MPEG V-DMC [153] is the first codec supporting the dynamic mesh representation format and therefore no anchor codec for that format can be selected.

7.3.6.2 Objective tests

Objective tests for dense dynamic point cloud codecs and dynamic mesh codecs follow the principles as defined in TR 26.955, besides that there is no anchor:

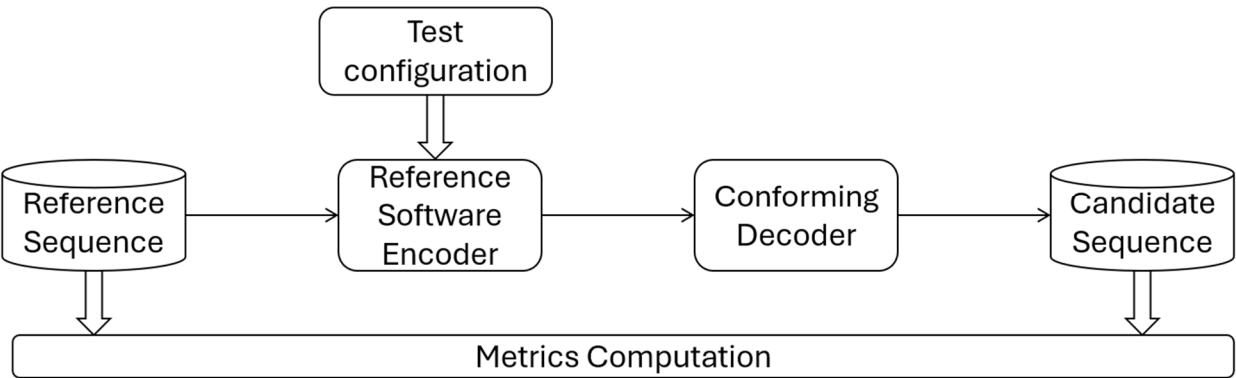


Figure 7.3.6.2-1 Test architecture for objective test

MPEG WG7 specified an objective metric that allows to characterize point cloud codecs. This objective metric is described in annex B of the Call for Proposals for Point Cloud Compression [136]. MPEG WG7 used this “point-based” metric to develop codecs for point cloud compression [137]. The “point-based” metric operates in 3D space and provides information on geometry and color distortion.

From 2D video objective testing it is known that a single objective metric is limiting, therefore “PCQM” [138] is added as a second objective metric in addition to the “point-based” metric.

Both objective metrics, the “point-based” metric and PCQM will be reported for all rate points and for all reference test sequences.

MPEG WG7 also specified objective metrics that allow to characterize dynamic mesh codecs. These objective metrics are described in annex B of the Call for Proposals for Dynamic Mesh Coding [154]. It includes in annex B.1 the “point-based” metric as primary metric and in B.2 the “image-based metric” as informative metric.

To compute metrics, MPEG provides a public version of mpeg-pcc-mmetric [139]. The metric software implements the “point-based” metric, the “PCQM” metric and the “image-based” metric as referred in this section.

Five rate points that cover a range from low quality to high quality for codecs to be characterized will be selected.

A spreadsheet will be provided to 3GPP SA4 that can be used to visualize the objective results.

Note that objective results obtained with the selected metrics can only be used for comparison with codec for the same format. There is no known objective metric that allows to compare different volumetric video codecs (e.g. dense point cloud against mesh) in a fair manner.

7.3.6.3 Subjective tests

Persons or objects in the “Streaming of professionally produced Volumetric Video with single asset containing people” scenario can be viewed from any angle and distances. To cope with this free viewpoint, MPEG WG7 developed a special procedure how to subjectively evaluate such volumetric sequences: The selected reference sequences are rendered with a point cloud renderer by following a pre-defined camera path into a 2D video. The procedure is done with the source reference sequence and with the coded/decoded reference sequence. Both 2D videos can then be evaluated with well-known 2D video evaluation methods.

The software tool that implements rendering of 2D videos from dense dynamic point clouds and dynamic meshes by following a camera path is provided by MPEG [140] and is named in the following representative renderer.

Due to specifics related point cloud rendering described in 4.3.3., this representative renderer supports two methods how to render voxels, one method is like cubes of a fixed size, the second method is splat blend. The latter draws a camera facing semitransparent splat of radius `PointSize` centered on the point position. The transparency (or alpha) varies with the distance from the center from fully opaque at the center to nearly transparent at the edge, to provide blending between points and reduce aliasing. In order to correctly use alpha blending, the points are sorted relative to the camera plane. Two blending modes are available, gaussian and linear, and the alpha falloff speed is customizable using the “pointFocus” setting.

For objective compression performance tests against the source reference, the cube method is recommended. For tests of the dense dynamic point cloud format against other volumetric representation formats such as mesh, the splat blend rendering is recommended as it fills holes between voxels. The representative renderer allows to activate a 3D background model or a neutral background. It also allows to activate a floor so that the test sequences look grounded.

A characteristic camera path for each volumetric test sequence is pre-defined to get a length of the output video as close as possible to 10s. A camera path that includes a full person view and a closer view at typical distance to the face is a good example.

The representative renderer produces uncompressed RGB 2D videos that can then be compressed with high bitrate HEVC. Resolution of the videos is 1080p, the frame rate and color space are aligned with the input point cloud.

The produced compressed 2D videos are made available to SA4 members so that the quality of a codec to be characterized can be judged.

There is no plan to engage an independent subjective test lab.

7.3.6.4 Correlation between the objective and subjective metrics

The objective metrics described in section 7.3.6.2 and the subjective comparison using the representative renderer in cube mode described in section 7.3.6.3 are used to evaluate coding distortions.

From MPEG there is no information available on the correlation between objective and subjective results and this may be investigated on a best effort basis.

A potential way of obtaining such correlation is described: the 4 licensed test sequences used in the MPEG subjective verification test report for V-PCC [141] can be obtained from the rights holders, the test sequences can be prepared as used in the verification test and objective metrics described in section 7.3.6.2 can be generated. The objective results can then be compared with the subjective results from the verification test and the correlation can be analyzed.

NOTE: For V-DMC the verification test is not available in the timeframe of this TR and no information will be provided.

7.3.6.5 Verification and crosscheck

All produced bitstreams, metric results and produced videos shall be crosschecked by at least one other 3GPP SA4 member to ensure that results are correct.

7.3.7 Interoperability Considerations for the application

MPEG-DASH is used with ISO/IEC 23090-10 Carriage of visual volumetric video-based coding data [142]

As of April 2025, MPEG is working on the second edition of ISO/IEC 23090-10 that will cover V-DMC storage in ISOBMFF format and delivery utilizing DASH.

RTP is not proposed for this scenario.

7.3.8 Test Sequences

NOTE: The content of this scenario relates to moving persons.

7.3.8.1 Candidate source dense point cloud sequences

Collected candidate raw dense point cloud sequences that are available for testing are presented in Annex C.2

7.3.8.2 Selected source dense point cloud sequences

This section lists 5 raw point cloud sequences that have been selected for objective and subjective testing. The following sequences have been selected for performing objective and subjective tests.

Test sequences have been selected based on visual quality and that these have not been used during codec development in MPEG. The following table lists the selected test sequences:

Table 7.3.8.2-1 Selected source point cloud sequences

Sequence	Content Provider	FPS	#Frames	Duration (s)	mean #point / frame	Attributes	Normals	geometry precision	attribute precision	Normal precision
Allyah	Render People	30	300	10	1.835.544	R,G,B	yes	11	8	float
Henry	Render People	30	300	10	1.818.531	R,G,B	yes	11	8	float
Nathalie	Volucap	30	300	10	1.640.033	R,G,B	yes	11	8	float
Mitch	Volucap	25	250	10	1.788.147	R,G,B	yes	11	8	float
Juggle Soccer	XD Productions	25	125	5	1.883.637	R,G,B	yes	11	8	float

The following thumbnails illustrate the selected test sequences:



Figure 7.3.8.2-1 Aliya, Henry, Nathalie, Mitch and Juggle Soccer (content courtesy by Renderpeople, Volucap and XD Productions)

The selected licensed source point cloud sequences can be grouped as follows:

- Group 1 - Freely available to 3GPP members: Nathalie, Mitch and Juggle Soccer
- Group 2 - Publicly purchasable: Henry
- Group 3 - Publicly free available: Aliyah

Sequences of group 1 have already been converted to pointclouds of around 2 million points per frame and maximum 10s length and are provided on the server. Sequences of group 2 and group 3 need to be downloaded and converted by those doing the test.

7.3.8.3 Metadata for source dense point cloud sequences

7.3.8.3.1 Overview

For a raw dense point cloud sequence used in the context of this Technical Report, the following metadata is proposed:

- Name of the sequence
- Scenario
- Sequence key as provided in TR 26.956
- URI to zip file containing PLY files (URL to weblink where the original sequence is available)
- Name Format of files
- Frame count
- Start frame number
- Frame rate
- Geometry precision
- Color format
- Peak Value (bounding box resolution e.g. 1023, 2047)
- Copyright statement
- Contact Person

Optionally, the following metadata can be added:

- Background
- TR 26.956 reference
- Thumbnail preview
- Preview MP4
- Average point number per frame
- Duration
- MD5 of the zip file containing all point cloud frames

Size in bytes of the zip file containing all point cloud frames.

A JSON scheme is defined in Annex B.2.3 for this matter. An example is provided in clause 7.3.8.3.3

7.3.8.3.2 JSON Scheme

JSON schema for the raw format can be found in Annex B.2.3.

7.3.8.3.3 Example

An example is attached to the zip file:

```
{
  "Sequence": {
    "Name": "Exemple",
    "Background": "This is a B2DV format example",
    "Scenario": "Streaming of Beyond 2D Produced VoD Content",
    "Key": "S01",
    "TR26.956": "Annex X.Y.Z"
  },
  "Properties": {
    "URI": "https://dash-large-
files.akamaized.net/WAVE/3GPP/Beyond2D/ReferenceSequences/file.zip",
    "thumbnail": "https://dash-large-
files.akamaized.net/WAVE/3GPP/Beyond2D/ReferenceSequences/file.png",
    "preview": "https://dash-large-
files.akamaized.net/WAVE/3GPP/Beyond2D/ReferenceSequences/file.mp4",
    "NameFormat": "exemple%04d.ply",
    "frameCount": 320,
    "startFrame": 0,
    "frameRate": 30,
    "pointsCountMean": 1000000,
    "geometryPrecision": 10,
    "colorformat": "rgb",
    "peak": 2047,
```

```
        "duration": 10.0,
        "md5": "d055a94f35f7594776186fc5d09a9fa4",
        "size": 11510343938
    },
    "CopyRight": "Conditions that are suitable for this study",
    "Contact": {
        "Name": "Celine Guede",
        "Company": "InterDigital",
        "e-mail": "celine.guede@interdigital.com",
        "generation": "provided by contact"
    }
}
```

7.3.9 Detailed test conditions

7.3.9.1 V-PCC test model and configuration files

The public version of the MPEG V-PCC test model named tmc2 in master branch is used to encode and decode dense dynamic point clouds [147].

For using tmc2 in Random Access (RA) mode, MPEG provides a configuration file [148].

- cfg/common/ctc-common.cfg
- cfg/condition/ctc-random-access.cfg
- cfg/hdrconvert/yuv420toyuv444_16bit.cfg

For each selected test sequence, a configuration file containing information needed for tmc2 configuration will be provided.

7.3.9.2 Rate points and test conditions

In line with the V-PCC verification test [149], 5 rate points R1 to R5 for Random Access (RA) are used for each test sequence. Fixed rate points are used to enable an indicative subjective comparison of V-PCC with potential future other codecs for scenario 2, including codecs supporting another potential representation format (e.g. dynamic mesh with V-DMC).

For test sequences with 11-bit geometry precision (vox11) with approximately 2M points/frame the following target bitrates in kbps are used:

- R1: 5000
- R2: 10000
- R3: 20000
- R4: 30000
- R5: 50000

Target bitrates are obtained by selecting values for the V-PCC codec parameters Occupancy Precision, QP Geometry and QP Texture. The values are selected per test sequence and are included in a JSON file that is used in the scripts for encoding. More information on encoding can be found in Annex D.3. The values have been selected by doing encodes

with varying value combinations and selecting those combinations which come close to the target bitrate and where a monotonic curve for objective metrics is obtained. There was no optimization in the sense of finding the closed bitrate match with the best objective and subjective performance.

In addition to the three codec parameters, a configuration file per test sequence is provided which is used by the encoding scripts.

7.3.9.3 Profiles

The V-PCC verification test [149] tested various V-PCC profiles such as HEVC Main10 V-PCC Basic Rec2, HEVC Main10 V-PCC Extended Rec2 and VVC Main10 V-PCC Extended Rec2 using test sequences with 10 bit geometry precision. To align with V-PCC prototype implementations at release time of this technical report, focus is on testing of the HEVC Main10 V-PCC Basic Rec0 profile using test sequences with 11-bit geometry precision.

7.3.9.4 Bitstream Generation, output

The MPEG V-PCC test model is used to encode and decode test sequences as described previously [147].

To compute metrics, the tool `mpeg-pcc-mmetric` [139] is used.

Scripts are provided to be able to:

- Encode each sequence for each condition, rate and profile
- Decode the corresponding sequence
- Compute the metrics
- Generate tables and graphs

Below are examples of command lines for the test profile for a `vox11` sequence:

- to encode a test sequence:

```
mpeg-pcc-tmc2/release-v25.0/bin/PccAppEncoder \
--config=mpeg-pcc-tmc2/release-v25.0/cfg/common/ctc-common.cfg \
--config=mpeg-pcc-tmc2/release-v25.0/cfg/condition/ctc-random-access.cfg \
--config=mpeg-pcc-tmc2/release-v25.0/cfg/sequence/${test_sequence}.cfg \
--configurationFolder=mpeg-pcc-tmc2/release-v25.0/cfg/ \
--uncompressedDataFolder=${source_sequence}/ \
--compressedStreamPath=${test_sequence}.bin \
--normalDataPath=${source_sequence}/${source_sequence}_%04d.ply \
--frameCount=32 \
--resolution=2047 \
--geometryQP=11 \
--attributeQP=28 \
--occupancyPrecision=2 \
--profileToolsetIdc=0 \
--profileReconstructionIdc=0 \
--mapCountMinus1=0
```

- to decode a test sequence:

```
mpeg-pcc-tmc2/release-v25.0/bin/PccAppDecoder \
--startFrameNumber=0 \
--compressedStreamPath=${test_sequence}.bin \
--reconstructedDataPath=${test_sequence}_dec_%04d.ply \
--inverseColorSpaceConversionConfig=mpeg-pcc-tmc2/release-
v25.0/cfg/hdrconvert/yuv420toyuv444_16bit.cfg
```

- to compute objective metrics of a test sequence:

```

mpeg-pcc-mmetric/1_1_7/build/Release/bin/mm \
--firstFrame 0 \
--lastFrame 31 END \
compare --mode pcc \
--inputModelA source_sequence_%04d.ply \
--inputModelB test_sequence_dec_%04d.ply END \
compare --mode pcqm \
--inputModelA source_sequence_%04d.ply \
--inputModelB test_sequence_dec_%04d.ply

```

For each test, outputs are:

- Bitstream file
- Log files containing metrics information
 - Encoder log output
 - Decoder log output
 - Metric log output

A CSV file containing concatenated metrics information for each condition and selected profile is generated for all sequences and rates.

The following information is stored:

- SeqId: identifier of the sequence
- CondId: tested condition (RA)
- RateId: tested rate number [R1..R5]
- nbFrame: number of tested frames
- NbInputPoints: number of points in the source sequence
- NbOutputPoints: number of points in the candidate test sequence
- MeanOutputPoints: mean number of points in the candidate test sequence
- MeanDuplicatePoints: mean number of duplicated points (with same geometry) in the candidate test sequence
- TotalBitstreamBits: size of the bistream in bits
- geometryBits: size of the geometry stream in bits
- metadataBits: size of the metadata stream in bits
- attributeBits: size of the attribute stream in bits
- D1Mean: mseF,PSNR (p2point)
- D2Mean: mseF,PSNR (p2plane)
- LumaMean: c[0],PSNRF
- CbMean: c[1],PSNRF
- CrMean: c[2],PSNRF
- PCQM: PCQM PSNR
- SelfEncoderRuntime: encoder time for current process
- ChildEncoderRuntime: encoder time for child processes
- SelfDecoderRuntime: decoder time for current process

- ChildDecoderRuntime: decoder time for child processes

From this CSV file, an excel spreadsheet is generated to get tables and graphs for interpretation of the results.

Annex D.3 explains the installation of the scripts and the test sequences preparation and annex D.4 explains the bitstream and objective metric generation. In addition to the instructions provided in these annexes, there is documentation in the doc folder of the installed repository

7.3.9.5 Videos Generation for subjective tests

The representative renderer [140] is used to generate videos for the subjective evaluation.

Videos are generated for the 5 selected test sequences and for the 5 rate points each. To evaluate the impact of rendering, each of the 25 videos is rendered in 3 modes as follows:

- Cube mode, neutral background with a color including a floor for the asset making the rendered scene more realistic
- Splat Blend mode, neutral background with a color including a floor for the asset making the rendered scene more realistic
- Splat Blend mode, 3D background model fitting with the test sequence

A camera path and blend parameter/render options adapted to each sequence are provided [151]. Backgrounds are presented in annex.E and annex C.2 links the background with the selected test sequences.

A script is provided to generate the videos with the chosen camera path and blend parameter/render options adapted to each sequence and the output is stored as high-quality video sequence of a length as close as possible to 10s.

The generation is done into two steps: first the generation of the RBG raw file with the camera path and then the conversion into YUV or MP4 files.

The video sequences are generated with the following video parameters:

- Video resolution: progressive uncompressed full-range HD format (1920x1080). Note that upsampling by the TV set should be avoided
- Frame rate: The frame rate will be aligned with the frame rate of the test sequence
- Color space: ITU-R BT.709
- Sub-sampling: 4:2:0 YUV 10 bits or x265 fast preset lossless

Annex D.5 explains the generation of videos and where to find these after successful execution of the scripts.

7.3.9.6 Verification / crosschecks

All produced bitstreams, metric results and produced videos will be crosschecked by at least one other SA member to ensure that results are correct.

7.3.10 External Performance data

The subjective verification test report for V-PCC can be downloaded from the public MPEG website [141]

The Brazilian SBTVD Forum performed objective tests with V-PCC. Full results are available in chapter 6.10 (Candidate Technology I), 6.10.3.2 and 6.10.4 of the following document [152]

7.3.11 Additional information

Sequences can be decoded and visualized in real time using a 3D background or in Augmented Reality on a smartphone, tablet, head-mounted display using DASH streaming mode or local file system.

Nokia's real-time V-PCC decoder implementation that was released as open source: <https://github.com/nokiatech/vpcc>

A simple scene description could be added to enable the placement of the asset in the scene (position, orientation, scale and etc) but is outside the scope of this document, which is focused on the format and codec evaluation.

InterDigital recently made a public release of a platform for evaluation and demonstration of real time decoding and rendering of V-PCC. A streaming server provides pre-encoded volumetric video content with V-PCC in real time following a user request. The content has been previously segmented thanks to a V3C DASH Packager able to handle V3C bitstreams. The decoder platform is composed of a native decoder plugin in charge of decoding the content and a simple host application in charge of rendering. The platform has been released via 5G-MAG [128].

7.4 Scenario 3: Streaming of Multi-view plus depth Produced Content

7.4.1 Motivation for the scenario

This scenario handles the streaming of produced multi-view plus depth content that provides experiences beyond what is achievable with 2D content. The scenario allows for the evaluation of the streaming of high-quality, professionally captured and produced multi-view plus depth video content.

In this scenario, multi-view plus depth content is played back on phones and tablets for which one viewport is shown. UI elements and/or the tilt sensor can be used to change the virtual viewport. This experience is more immersive than 2D video because the (subtle) pose changes provide motion parallax which is a strong perceptual depth cue. This effect is already achieved using a small number of cameras (3-4) and a limited viewing space (the size of a person's head). More cameras (10-20) are needed for free-viewpoint functionality.

In this scenario, it is also considered, but not evaluated, that the same content can be played back on more advanced devices including head-mounted displays and eye-tracked autostereoscopic displays. For these classes of devices, two virtual viewports are rendered, thereby providing a stronger 3D effect due to the combination of motion parallax and stereopsis depth cues. While relevant, the expectation is that for the foreseeable future the majority of the UE's will be 2D phones and tablets.

The main benefits of using the multi-view plus depth representation for on-demand streaming, is that 1) the difference in appearance of objects between cameras is preserved, making the experience more like video and less like graphics, 2) less processing steps are needed to construct the representation as compared to the derived representations, 3) transmission is possible by the combination of 2D video plus metadata.

As of start of 2025, no commercial deployment of multi-view plus depth *delivery* to mobile devices has been identified. Multi-view plus depth video can offer an experience on top of 2D video. First services can provide stills or short clips that enable a viewer to look around and observe some actions from different viewpoints. This can be a stepping stone towards live streaming of multi-view plus depth video.

This scenario is based on the multi-view plus depth video representation format that is defined in clause 4.3.4. Capturing setups and production software are available as described in the related representation format definitions. Contribution, compression and storage formats for multi-view plus depth video are available, see clause 7.4.3. It is expected that segmented media delivery will be used based on DASH and ISO/BMFF. Carriage of coded media using ISO/BMFF has been specified for MIV [166] in [160] and MV-HEVC [167] in [165]. Other codecs may be considered. Hardware video decoder capabilities can be used for all pixel data. Rendering and display systems for multi-view plus depth video are described in clause 4.3.4.3.

7.4.2 Description of the scenario

This scenario considers on-demand streaming of multi-view plus depth produced content to a UE (Figure 7.4.2-1). All or the most relevant parts of the content are produced using a camera array that observes a scene. The array may include 2D cameras and/or range-sensing cameras. In some cases, part of a scene may be created or inpainted using AI or CGI to reduce the number of physical cameras. This scenario does not consider use of AI/CGI production without a physical camera array.

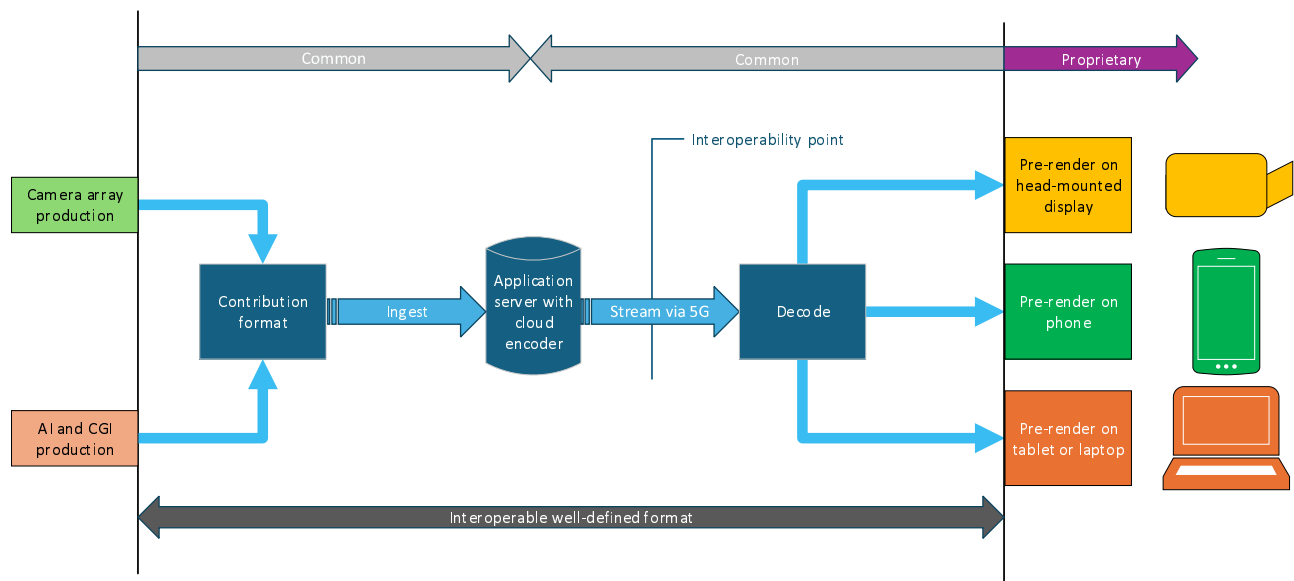


Figure 7.4.2-1: On-demand streaming of B2D produced content to a UE

Capture setup, production tools and workflows for multi-view plus depth video capture systems and production tools are described in clause 4.3.4.2. Contribution, compression and storage formats are linked to the multi-view video representation format. Well-defined contribution formats exist that carry the raw texture/depth images and camera parameters, e.g. as described in clause 4.3.4.2. Compression formats for multi-view plus depth video are described in clause 4.3.4.4. One codec that can be used to realize this scenario is MPEG Immersive Video (MIV) [166]. Another option is the Multiview extension of high efficiency video coding (HEVC) standard [167], commonly referred to as MV-HEVC, as well as the 3D extensions of HEVC (3D-HEVC). MV-HEVC enables the encoding of multiple views and depth data in HEVC by allowing the presence of additional layers in an HEVC bitstream, each corresponding to either a different view or depth information. Such support is enabled through only high level syntax modifications in the original design of the HEVC standard making it easy to repurpose multiple HEVC encoders or decoders that might be available in an existing implementation. 3D-HEVC, however, introduces additional low level coding tools intended for the improved compression of depth information and might not be available in most implementations. Other codecs may also be considered. Below one possible workflow with MIV is described.

The multiple camera views and depth maps are encoded to create a unified representation. An example could be MIV constrained to one or more atlases and packed video data. The single video sub-bitstream per atlas would be encoded with the HEVC Main10 profile. The bitstream contains all camera parameters that are necessary for 6DoF rendering. Each atlas is independently renderable. Another example can be MV-HEVC using auxiliary layers for depth maps and SEI messages for camera parameters. Figure 7.4.2-2 provides an example of an MIV encoder flow.

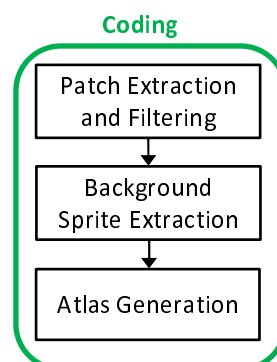


Figure 7.4.2-2: MIV encoder example

- Patch Extraction and Filtering: extraction of regions from the texture and depth map for the purpose of pixel-rate reduction and allowing object interactivity at the client.
- Background View Extraction: The ground surface and far-away background can be represented by a single background texture with depth. This greatly reduces the required pixel space.

- Atlas Generation: The patches and sprite are packed in an atlas such that both the pixel area is optimally used and the temporal correlation is retained to guarantee an acceptable bitrate.

An example of multi-view plus depth video encoding has been described in the paper [161].

The encoded bitstream is encapsulated to ISOBMFF according to the rules of the used codec.

For example, an MIV bitstream may be packaged in one track, or multiple tracks where the packed video data is one track, common atlas data is one track, and atlas data is another track. ISO/IEC 23090-10 [160] specifies how to map MIV (V3C) onto ISOBMFF, file format and DASH.

When a scene is represented by multiple atlases, only one of them may be decoded based on the viewing position. This is called atlas-level sub-bitstream access. In the case of DASH, switching atlas would amount to changing tracks.

The decoder(s) will make use of hardware video decoder capabilities for all pixel data, and metadata describing information needed for rendering is decoded/parsed by a CPU.

Rendering and display systems for multi-view plus depth video are described in clause 4.3.4.3.

7.4.3 Source format properties

The source format that is commonly used and recognized is a set of video sources in combination with a format for camera parameters. Texture is typically available in 8 or 10 bit unsigned integer formats and geometry is typically available in 16 bit unsigned integer or 32-bit float formats. The Colmap format [165] is most commonly used in the literature. There is also a JSON format that is used within multiple MPEG activities, and a converter between the two camera parameter formats exists [164].

For this scenario, the multi-view plus depth video source format has 3 to 20 views. It is expected that most or all test data will have perspective projection (PSP), but test data with equirectangular projection (ERP) may be included.

Each view has the following components:

- Texture (color)
- Depth coded as normalized disparity

Depth information can be used in rendering e.g. by shaders for surface normal estimation.

NOTE: Further details on depth processing is FFS.

All views have view parameters: camera ID, camera intrinsics, camera extrinsics (pose) and depth quantization parameters (optional).

Views may be undistorted, otherwise distortion parameters have to be provided.

The signal properties defined in clause 4.3.4.1 apply with no further constraints.

7.4.4 Encoding and decoding constraints and settings

Some constraints and settings below are given for MIV:

Codec profiles/levels:

- HEVC Main 10 MIV Main
 - MIV level 2.0 or 2.5.
- HEVC Main 10 MIV Extended
 - MIV level 2.0, 2.5 or 3.0 whereby the level 3.0 is only allowed if there is a single video sub-bitstream.

Support for multi-plane image (MPI) through the MIV Extended Restricted Geometry sub-profile may be relevant for this scenario, but it is not considered for this study for practical reasons: it requires an additional conversion from multi-view plus depth to multi-plane image.

Some constraints and settings below are given for MV-HEVC:

For content with N views plus depth, there will be the following layers:

- First (reference) texture layer: Main or Main 10 profile, level 4.1 or 5
- $N - 1$ dependent texture layers: Multiview Extended or Multiview Extended 10 profile, level 4.1 or 5
- First (reference) depth layer (AuxId = AUX_DEPTH): Multiview Monochrome or Multiview Monochrome 10 profile, level 4.1 or 5
- $N - 1$ dependent depth layers: Multiview Monochrome or Multiview Monochrome 10 profile, level 4.1 or 5

The presence of the following two SEI messages is required for virtual view synthesis:

- Depth representation info SEI
- Multiview acquisition info SEI

Note that the coding of the depth layers could be independent of the coding of the texture layers.

In general, a random-access frequency of 32 frames can be considered. It is up to the service provider to define the exact random access frequency.

Transmission systems need to be prepared to resend data in case of data loss. If data loss still occurs or retransmitted data does not reach the receiver device in time for rendering, previous immersive frames may be re-rendered with updated viewing poses. In case one or more of the sub-bitstreams is lost, it is up to the application to determine an optimal method for hiding the missing information.

Typically, bitrates between 5 and 50 Mbit/s may be considered.

Bitrate parameters related to video sub-bitstreams need to be configured by the streaming service provider. Transfer characteristics are signalled in the video sub-bitstreams.

There are no special requirements regarding ABR. Configuration is left for the service provider to determine.

Latencies between 500ms to several seconds are considered. Random access interval or segment duration are configured according to the latency requirements.

Encoding is performed by a content provider. This scenario assumes professional setting for recording and processing the content, so no real-time or encoder hardware or architecture requirements are provided.

It is expected that devices support HW accelerated video decoding.

Decoding requirements for MIV:

- HEVC Main 10
- HEVC levels are determined according to the maximum HEVC Level that is needed for a video sub-bitstream decoder to fulfill the MIV level.
 - HEVC level 5.1 for MIV level 2.0
 - HEVC level 5.2 for MIV level 2.5
 - HEVC level 6.1 for MIV level 3.0
- Video sub-bitstreams need to be independently decodable. This helps implementations on various platforms that may have only high-level APIs. For instance, geometry needs to be full range.

Samples in the sub-bitstreams should be temporally aligned.

Decoding requirements for MV-HEVC:

While decoding and rendering all views may result in a higher quality, this is not a requirement. At a minimum, a client needs to be able to select the two nearest views for decoding and rendering. This requires the decoding of the following six sub-bitstreams:

- Reference texture: Main or Main 10, level 4.1 or 5
- Reference depth: Multiview Monochrome or Multiview Monochrome 10, level 4.1 or 5
- Two dependent textures: Multiview Extended or Multiview Extended 10, level 4.1 or 5
- Two dependent depths: Multiview Monochrome or Multiview Monochrome 10, level 4.1 or 5

The view selection can change each intra period.

7.4.5 Performance Metrics and Requirements

The tests are run for a chosen level as described in clause 7.4.6. Bitstreams are provided. Camera calibration, and depth estimation, and encoding are not evaluated.

The test will have four rate points and QP values are selected for each sequence to approximately match the 5 to 50 Mbps range. When saturation occurs before 50 Mbps a lower value may be chosen in consultation. When there are multiple video components or packed regions then the other QP values need to be directly derived from the texture QP using an equation or a look-up table. (They cannot depend on the sequence.)

The QMIV tool [35], available at <https://gitlab.com/mpeg-i-visual/qmiv>, is available to compute full-reference objective metrics:

- PNSR,
- Weighted sphere PSNR (WS-PSNR) [171],
- Immersive video PSNR (IV-PSNR) [34],
- Immersive video SSIM (IV-SSIM) [172].

All source views that were used for encoding are provided. Each source view is reconstructed by decoding and rendering (view synthesis). The QMIV tool is then run on all source views and the score is averaged over all views.

Depending on bit rate, quality of depth maps and rendering, either the video codec or view synthesis is the limiting factor. BD-PSNR is calculated for both metrics because the metric behaves more predictably than BD-rate.

There is experience in testing of multi-view plus depth video in MPEG context. The test conditions as described are a simplification and evolution of the common test conditions for MIV defined in [162].

The main challenge with testing of multi-view plus depth video is that codecs are asymmetric. The input is a number of source views (with depth maps), and the output of the decoder + renderer can be any viewport within a spatial region around those source views. In the mentioned CTC two tests are used:

- Objective evaluation at source view positions
- Subjective evaluation of pose trace videos (dynamic viewports)

This has resulted in a lack of correlation between objective and subjective results, but despite that it is the best-known approach. Alternatives that have been tried and dismissed (for now):

- Objective evaluation at dynamic viewports: It includes view synthesis in the reference condition and this skews the results towards a specific renderer. It prevents an A/B comparison of different renderers.
- Subjective evaluation at source view positions: This is not how the end-user will interact with the content, and it does not evaluate artifacts due to viewport dynamics.

For this test, because the aim is to prove feasibility of a scenario, objective evaluation may be sufficient, especially when supplemented with (informal) real-time demonstration of the same bitstreams that were used for objective evaluation.

7.4.6 Interoperability Considerations for the application

The multi-view plus depth video bitstream needs to be carried over DASH for this scenario. It is not necessary to prove this as part of the feasibility test, if written evidence can be provided.

In the example of using MIV as a codec, there are implementations for DASH [5G-MAG] and RTP + SDP [uvgRTP]. It is possible to subset MIV to always transmit all pixel data in a single packed video track plus a timed metadata track.

7.4.7 Test Sequences

The evaluation has been performed on the sequences listed in Table 7.4.7-1. Their full description is presented in Annex C.4..

Table 7.4.7-1: Test sequences for the evaluation of the scenario

Sequence	Provider	Frames	Resolution	Bit depth	Color format
Breakfast	InterDigital	97 @ 30 Hz 3.2 s	1920 x 1080 5 x 3 views	texture: 10b depth: 16b	texture: 4:2:0 BT.709 depth: 4:2:0 full range linear
Bartender	ETRI	300 @ 30 Hz 10.0 s	1920 x 1080 21 views	texture: 10b depth: 16b	texture: 4:2:0 BT.709 depth: 4:2:0 full range linear
DanceMoves	Philips	449 @ 15 Hz 29.9 s	1920 x 1080 6 views	texture: 10b depth: 16b	texture: 4:2:0 BT.709 depth: 4:2:0 full range linear

7.4.8 External Performance data

For MIV the performance data is available from the verification test report [163].

NOTE: This performance data was based on different source view properties and the results may not translate to this study.

7.4.9 Additional Information

The Metaverse Standards Forum (MSF) has established a Volumetric Media Interoperability working group which aims to build a better understanding of volumetric media, including multi-view plus depth video, to identify relevant areas of applications and compatibility requirements, and to establish common requirements for different systems. See here the WG description: <https://metaverse-standards.org/domain-groups/volumetric-media-interoperability/>

The technology is expected to be highly scalable since it uses well-established transport technologies like DASH and 2D video coding techniques.

Regarding complexity, rendering and decoding frame rates for MIV content were measured for Windows and Android platforms in [161]. The results show that the developed platform can decode V3C content in real time on both Windows and Android. Evaluation of battery consumption (power levels) is FFS.

Streaming of multi-view plus depth content has the potential to disrupt several markets including entertainment/media, education/training, retail/shopping.

Several use cases can be envisioned related to these domains. For example, in an education/training scenario, a pre-recorded video of a fitness instructor showing how to perform an exercise can help the student to better understand how the exercise is done and thus replicate in a correct way. Another example in education domain would be a mechanic giving a tutorial on how to assemble a mountain bike. The viewer can watch the movements of the mechanic from different angles and get an improved understanding of the different steps due to depth perception and different viewpoints. In the entertainment domain, users can stream a performance from their favorite band to their living room and experience greater immersion potentially together with spatial audio..

8 Common Evaluation Features

There is no common evaluation features (e.g., metrics, software...) identified.

9 Evaluation of Selected Scenarios

9.1 Introduction

This clause defines test conditions and parameters, KPIs, Metrics, test sequences, agreed reference signals per scenario, and also provides the evaluation results.

9.2 Scenario 1: UE-to-UE Stereoscopic Video Live Streaming

9.2.1 Evaluation Overview

This section presents an overview of the evaluation process for the UE-to-UE stereoscopic video live streaming scenario. MV-HEVC and simulcast HEVC are detailed as candidate solutions. It includes objective evaluation results.

9.2.2 Reference Sequences

The evaluation has been performed on the sequences presented in Annex C.3.2, C.3.3, C.3.4, C.3.5, C.3.6, and C.3.7. Only the first 150 frames were used of each test sequence.

9.2.3 Performance Metrics

Only Peak-Signal to Noise Ratio $PSNR(Y)$ is provided.

9.2.4 Candidate Solutions

9.2.4.1 Solution 1: Simulcast HEVC

9.2.4.1.1 Introduction

This clause provides the evaluation process for Scenario 1 using simulcast HEVC, based on the test conditions defined in Clause 7.2.8.

9.2.4.1.2 Reference Software

The reference software for HEVC is called HM (HEVC Test Model). The source code can be downloaded from the GitLab: <https://vcgit.hhi.fraunhofer.de/jvet/HM>.

After compiling the source code, the following executable files were generated:

- **TAppEncoderStatic / TAppEncoderStaticd:** The encoder application. It converts raw video data (e.g., YUV) into an HEVC bitstream. It implements all encoding tools as defined by the HEVC standard, configurable via command-line arguments.
- **TAppDecoderStatic / TAppDecoderStaticd:** The decoder application. It decodes an HEVC bitstream back into raw video data (YUV). It is primarily used for verifying the correctness of the encoded bitstream.
- **TAppDecoderAnalyserStatic / TAppDecoderAnalyserStaticd:** An enhanced decoder with analysis capabilities. In addition to decoding, it provides detailed statistics on the decoding process, such as information per CU/PU/TU, distribution of motion vectors, prediction modes, and quality metrics (e.g., PSNR, SSIM).
- **TAppExtractorStatic / TAppExtractorStaticd:** A tool for extracting specific information from an HEVC bitstream, such as NAL unit information, parameter sets (VPS, SPS, PPS), or data for specific frames/slices.
- **TAppRendererStatic / TAppRendererStaticd:** A tool to render decoded raw video data directly to the screen for visual inspection, often using graphics APIs to convert YUV to RGB for display.

- **annexBbytecountStatic / annexBbytecountStaticd:** A utility to analyze and count bytes of NAL units within an Annex B formatted HEVC bitstream. Useful for validating stream structure and gathering NAL unit size statistics.
- **convert_NtoMbit_YCbCrStatic / convert_NtoMbit_YCbCrStaticd:** A utility for converting the bit-depth of YCbCr data (e.g., from 8-bit to 10-bit) or for converting between chroma subsampling formats (e.g., YUV420 to YUV422).

The following Python scripts were used to calculating objective metrics:

- The script for calculating MV-HEVC bitrate and Y-PSNR is under `./metrics/analyze_mvhevc.py`
- The script for calculating Simulcast HEVC bitrate and Y-PSNR is under `./metrics/analyze_hm_logs.py`
- The script for calculating BD-Rate is under `./metrics/compare.py`

9.2.4.1.3 Parameter Settings

Run HM TAppEncoderstatic to encode each YUV file using the fixed QP values that are defined in clause 7.2.8.2, the same QP values are used for all sequences and all encoder conditions.

All sequences have been encoded using the configurations in `./cfg/HM/encoder_lowdelay_main_rext.cfg`.

Encoding was performed by running the shell script with appropriate parameters. The shell scripts are in the attached sh files. For example:

```
echo "streetView_captured_left: QP=17"
./TAppEncoderStatic -c ~/HTM-16.3/cfg/HM/encoder_lowdelay_main_rext.cfg -i
~/yuvfiles/streetView_captured_left.yuv -o output0703.bin --SourceWidth=1920 --
SourceHeight=1080 --FrameRate=30 --FramesToBeEncoded=150 --QP=17

echo "streetView_captured_right: QP=17"
./TAppEncoderStatic -c ~/HTM-16.3/cfg/HM/encoder_lowdelay_main_rext.cfg -i
~/yuvfiles/streetView_captured_right.yuv -o output0703.bin --SourceWidth=1920 --
SourceHeight=1080 --FrameRate=30 --FramesToBeEncoded=150 --QP=17
```

9.2.4.1.4 Evaluation Results

The evaluation results are provided in the attached logs files, under `./logs/HEVC`, and summarize in Table in clause 9.2.5.

9.2.4.1.5 Network Requirements

Simulcast HVEC serves as a baseline for performance, the evaluation results indicate that to achieve very high objective quality (PSNR > 41 dB), simulcast HEVC requires significant high bandwidth. For example, the complex scenes such as Street View and Cute Dog, the bitrate demand ranges from 30,000 kbps to 83,000 kbps. Even at lower quality settings (QP 27, 32, 37), Simulcast HEVC still requires notable bandwidth demands for complex scenes, ranging from 1,500 kbps to 14,500 kbps.

9.2.4.2 Solution 2: MV-HEVC

9.2.4.2.1 Introduction

This clause provides the evaluation process for Scenario 1 using MV-HEVC, based on the test conditions defined in Clause 7.2.8.

9.2.4.2.2 Reference Software

The reference software for Multiview High Efficiency Video Coding (MV-HEVC) is an extension of the HEVC (H.265) standard that supports multiple views. The latest version of the MV-HEVC reference software (HTM) is available on: https://hevc.hhi.fraunhofer.de/svn/svn_3DVCSoftware/branches/HTM-16.3-fixes/.

9.2.4.2.3 Parameter Settings

Run HM TAppEncoderStatic to encode each YUV file using the fixed QP values that are defined in clause 7.2.8.2, the same QP values are used for all sequences and all encoder conditions.

All sequences have been encoded using the configurations in ./cfg/MV-HEVC/baseCfg_2view. The configuration files are in the attached cfg files.

Encoding was performed by running the shell script with appropriate parameters. The shell scripts are in the attached sh files. For example:

```
echo "dog_ai_generated: QP=17"
./TAppEncoderStatic -c ./cfg/baseCfg_2view_movegirl_ai_generated.cfg --QP=17 -
wdt 1920 -hgt 1080 -fr 30 -f 150
echo "dog_ai_generated: QP=22"
./TAppEncoderStatic -c ./cfg/baseCfg_2view_movegirl_ai_generated.cfg --QP=22 -
wdt 1920 -hgt 1080 -fr 30 -f 150
echo "dog_ai_generated: QP=27"
./TAppEncoderStatic -c ./cfg/baseCfg_2view_movegirl_ai_generated.cfg --QP=27 -
wdt 1920 -hgt 1080 -fr 30 -f 150
echo "dog_ai_generated: QP=32"
./TAppEncoderStatic -c ./cfg/baseCfg_2view_movegirl_ai_generated.cfg --QP=32 -
wdt 1920 -hgt 1080 -fr 30 -f 150
echo "dog_ai_generated: QP=37"
./TAppEncoderStatic -c ./cfg/baseCfg_2view_movegirl_ai_generated.cfg --QP=37 -
wdt 1920 -hgt 1080 -fr 30 -f 150
```

9.2.4.2.4 Evaluation Results

The evaluation results are provided in the attached logs files, under ./logs/MV-HEVC, and summarize in Table in clause 9.2.5.

9.2.4.2.5 Network Requirements

MV-HEVC is designed to improve coding efficiency by leveraging redundancy between different views. The evaluation results show significant lower network bandwidth requirements compared to a Simulcast HEVC approach for stereoscopic video. The bandwidth savings are substantial, ranging from 20% to over 50% depending on the content and quality level. For example, the "Cute Dog - Generated" sequence at R2 (QP 22), MV-HEVC achieved a bitrate of 16.5 Mbps, representing a 50% reduction compared to Simulcast HEVC's 32.8 Mbps.

9.2.5 Summary of Evaluation

The evaluation results are summarized in Table 9.2.5-1.

Table 9.2.5-1: Evaluation Results with Simulcast HEVC and MV-HEVC for Scenario 1

Test Sequence	Rate points	Simulcast HEVC		MV-HEVC	
		Bitrate [kbps]	PSNR [dB]	Bitrate [kbps]	PSNR [dB]
Street View - captured	R1 (17)	82597.98	45.46	[77934.46]	[42.79]
	R2 (22)	40906.65	41.19	[29811.63]	[37.89]
	R3 (27)	12828.06	36.93	[9759.37]	[35.08]
	R4 (32)	4591.49	34.35	[3882.06]	[32.86]
	R5 (37)	1655.49	31.69	[1568.76]	[30.38]
Street View - Generated	R1 (17)	79587.76	45.59	[63793.59]	[42.05]
	R2 (22)	37996.06	41.31	[22728.87]	[38.38]
	R3 (27)	11355.45	37.19	[7199.21]	[35.87]
	R4 (32)	3988.24	34.69	[3105.62]	[33.83]
	R5 (37)	1506.74	32.03	[1387.66]	[31.37]
Cute Dog - Captured	R1 (17)	60282.85	46.58	50309.66	43.54
	R2 (22)	31296.69	42.86	21219.06	40.21
	R3 (27)	13854.47	39.02	8549.64	37.45
	R4 (32)	7691.15	35.89	4030.95	34.97
	R5 (37)	4361.23	32.99	2146.72	32.38
Cute Dog - Generated	R1 (17)	64623.85	46.53	40674.14	43.26
	R2 (22)	32782.40	42.69	16506.89	40.33
	R3 (27)	14509.13	38.79	6686.76	37.67
	R4 (32)	7981.78	35.69	3274.28	35.22
	R5 (37)	4504.21	32.82	1771.73	32.65
Moving Girl - Captured	R1 (17)	9100.29	51.28	8944.59	49.49
	R2 (22)	3319.19	48.92	2661.69	47.36
	R3 (27)	1485.50	46.75	947.79	45.72
	R4 (32)	818.86	44.34	475.70	43.77
	R5 (37)	480.74	41.72	250.63	41.41
Moving Girl - Generated	R1 (17)	9788.79	51.19	7,297.06	49.41
	R2 (22)	3630.03	48.73	2300.78	47.26
	R3 (27)	1523.01	46.49	822.27	45.59
	R4 (32)	825.61	44.14	422.32	43.75
	R5 (37)	478.95	41.56	250.63	41.58

Table 9.2.5-2 shows the Bjøntegaard Delta (BD) bitrates reductions obtained by MV-HEVC in comparison to Simulcast HEVC.

Table 9.2.5-2: BD-rate reduction of MV-HEVC relative to Simulcast HEVC for Scenario 1

Test Sequence	BD-rate reduction of MV-HEVC [%] relative to Simulcast HEVC
Street View - captured	[61.580523]
Street View - Generated	[22.565416]
Cute Dog - Captured	-8.714002
Cute Dog - Generated	-37.240038
Moving Girl - Captured	-2.813581
Moving Girl - Generated	-20.596453

NOTE: The evaluation results for Street View—both captured and generated require further verification.

9.3 Scenario 2: Streaming of professionally produced Volumetric Video with single asset containing people

9.3.1 Evaluation Overview

Clause 7 identifies the dense dynamic point cloud and dynamic mesh representation formats as the dominant formats used for provision of services based on scenario 2.

To ensure good quality for scenario 2, around 2 million points per frame for a dense dynamic point cloud and around 30k triangles and 4K texture per frame for dynamic mesh have been identified as appropriate. More details can be found in table 7.3.4-1 and table 7.3.4-2.

At time of writing of this technical report the MPEG V-DMC [110] test model was not publicly available, so for the evaluation of MPEG V-DMC is referred to clause 11.

The remaining part of clause 9.3 concentrates on the evaluation of dense dynamic point clouds as representation format and MPEG V-PCC [131] as the codec.

Here follows a quick summary of the selected constraints of the representation format and encoding/decoding constraints which allow good quality including proven real time decoding/rendering on off-the-shelf consumer devices such as smartphones, tablets and VR headsets:

- Representation format dense dynamic point cloud: up to 2 million points / frame, 11-bit bounding box, more details in table 7.3.4-1
- Encoding/decoding: HEVC Main10 V-PCC Basic Rec0 profile with bitrates of up to 50Mbit/s, more details in table 7.3.5

The envisaged interoperability point is the one between the application server/playout system in the cloud and the consumer device, see figure 7.3.3-1, so focus is delivery volumetric video with single asset to end consumers.

As performance metrics, two objective metrics are provided and videos are provided for self-conducting subjective viewing.

9.3.2 Reference Sequences

Clause 7.3.8.1 lists all available candidate raw dense point cloud sequences and clause 7.3.8.2 selects from these 5 sequences for objective and subjective testing. Criteria for the selection were quality and diversity of content providers.

9.3.3 Performance Metrics

Clause 7.3.6.2 describes the “point-based” metric [136] and the “PCQM” metric [138]. Both objective metrics, the “point-based” metric and “PCQM” are reported for all rate points and for all test sequences. MPEG V-PCC [131] is the first codec supporting the dense dynamic point cloud representation format with inter coding and therefore no anchor codec for the format has been selected.

9.3.4 Candidate Solutions

9.3.4.1 Solution 1: MPEG V-PCC profile HEVC Main10 V-PCC Basic Rec0

9.3.4.1.1 Introduction

The generation of objective metrics and generation of 2D videos for subjective viewing for scenario 2 is supported by a software package that widely automates the whole process. The principal stages are test sequence preparation, bitstream and objective metric generation and video generation.

9.3.4.1.2 Reference Software

Clause 7.3.9.4 describes the stages for sequence preparation and bitstream and objective metric generation. For encoding and decoding the software package uses the MPEG V-PCC test model [147].

Clause 7.3.9.5 describes the stage video generation.

9.3.4.1.3 Parameter Settings

Clause 7.3.9.1 describes the principal configuration files for the V-PCC test model [147], where e.g. the random-access mode is selected.

Clause 7.3.9.2 describes additional configuration information to obtain fixed target bitrates. Target bitrates are obtained by selecting values for the V-PCC codec parameters Occupancy Precision, QP Geometry and QP Texture per sequence. The fixed bitrate is not fully fixed, it is rather an average over the sequence length, which is 5s or 10s depending on the sequence. Such fixed bitrates have been selected to enable an indicative subjective comparison of V-PCC with potential future other codecs for scenario 2, including codecs supporting another potential representation format (e.g. dynamic mesh with V-DMC).

9.3.4.1.4 Distribution

The performed evaluation is on the V3C bitstream level does not include packaging and delivery of V-PCC based on ISO/IEC 23090-10. So potential overhead of packaging and delivery is not included in the evaluation.

9.3.4.1.5 Evaluation Results

9.3.4.1.5.1 Objective evaluation

Below the graphs are plotted for the point-based metric (PSNRs for D1, D2, Cb, Cr and Luma) and for the PCQM metric. These results can be used as an anchor for comparing in future with other point-cloud based codecs for volumetric video.

Bitstreams are provided for those sequences that are provided as reference sequences on Aspera, i.e. Mitch, Nathalie and JuggleSoccer. Bitstreams can be accessed as follows:

- Log into Aspera: <https://aspera.pub/I4tSQ8k>

- 3GPP members can request credentials by sending a request per email to:
3GPP_B2D_Datasets@interdigital.com
- Go to directory Bitstreams/Scenario-2/V-PCC. In the directories mitch, nathalie and jugglefootrouge there are zip files containing the bitstreams for the 5 rate points. The file md5sum_bin.txt contains the md5 checksums that should be used to check if the download was correct.

For sequences coming from Renderpeople like Henry and Aliyah the bitstreams cannot be shared due to the license. Here again the file md5sum_bin.txt contains the md5 checksums that should be used to check if it was possible to generate the bitstream from scratch.

The spreadsheet with full objective results can be downloaded from Aspera by using the same credentials and by going to the directory Bitstreams/Scenario-2/V-PCC/Metrics. To access the spreadsheet, open the file FiDx0_Basic_C2RA_3gpp_test_configuration.xlsm. The spreadsheet has one tab named “C2 lossy RA” with detailed information how bits are spent between geometry, occupancy and color, PSNR information for both metrics and information on encoding/decoding time. The tab “Graphs” shows the plots for both metrics which are included below in this document. The sequence can be selected on the upper left corner when clicking on the sequence name.

9.3.4.1.5.1.1 Objective results of sequence Mitch

The following 5 figures present the point-based metric results.

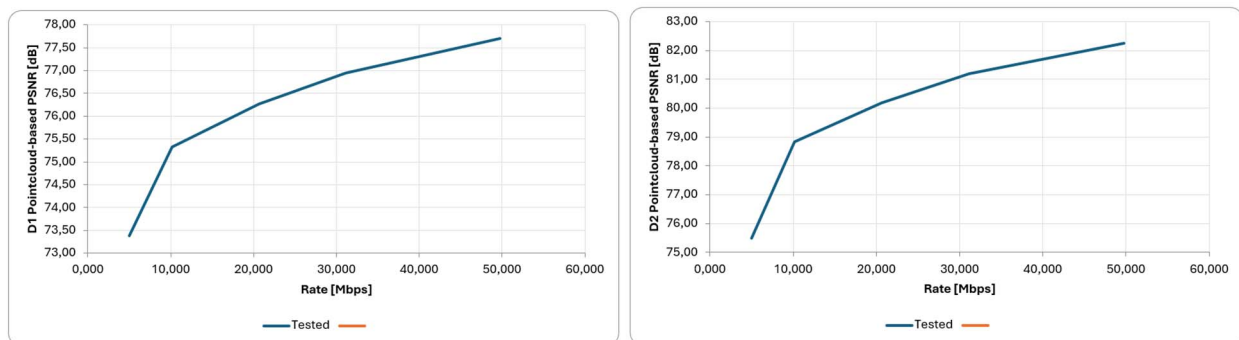


Figure 9.3.4.1.5.1.1-1: D1 and D2 metrics

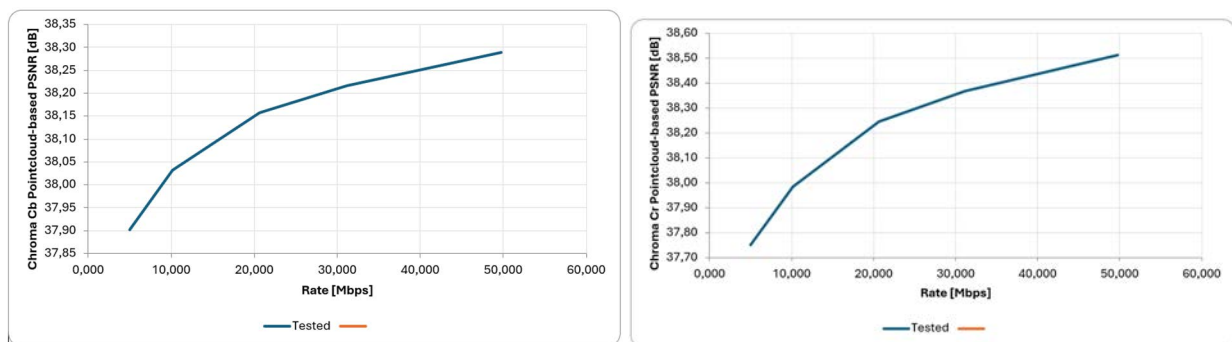


Figure 9.3.4.1.5.1.1-2: Cb and Cr metrics

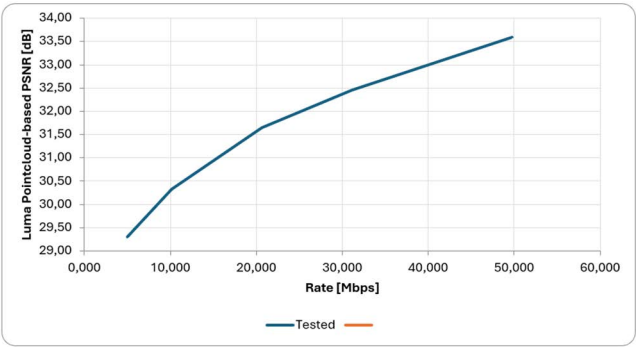


Figure 9.3.4.1.5.1.1-3: Luma metric

The following figure presents the PCQM metric results.

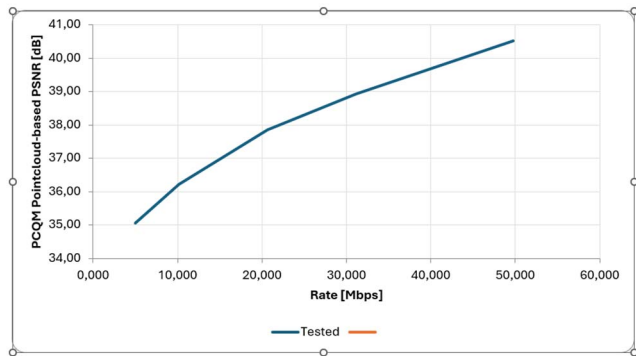


Figure 9.3.4.1.5.1.1-4: PCQM metric

9.3.4.1.5.1.2 Objective results of sequence JuggleSoccer

The following 5 figures present the point-based metric results.

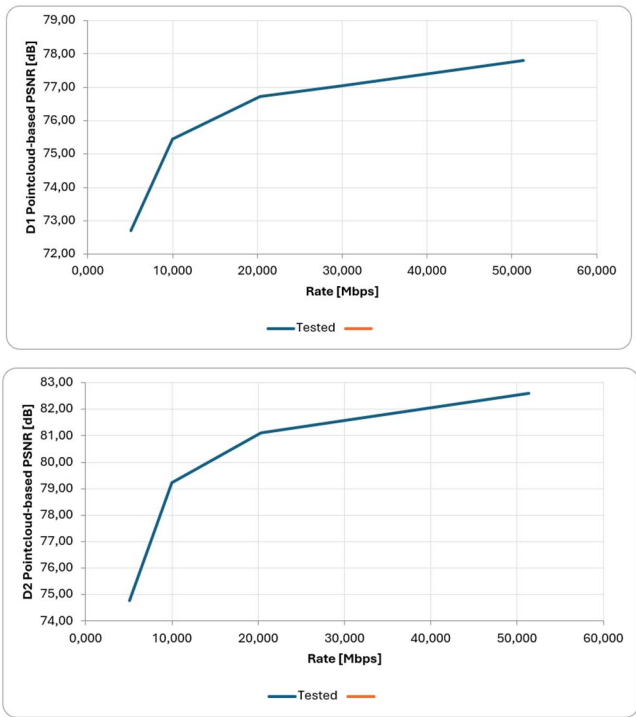


Figure 9.3.4.1.5.1.2-1: D1 and D2 metrics

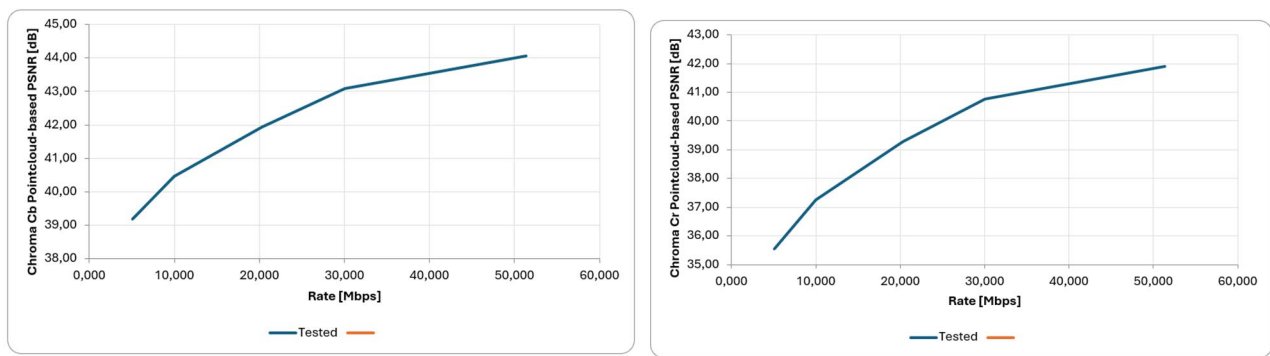


Figure 9.3.4.1.5.1.2-2: Cb and Cr metrics

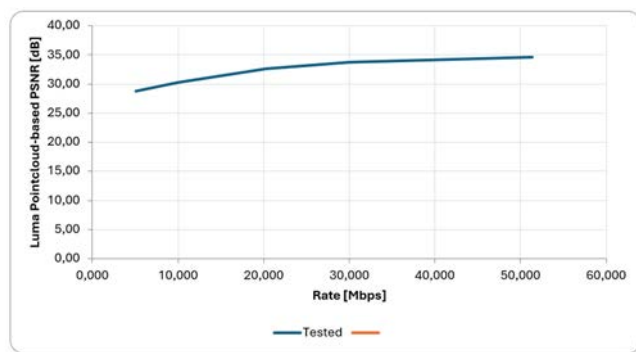


Figure 9.3.4.1.5.1.2-3: Luma metric

The following figure presents the PQCM metric results.

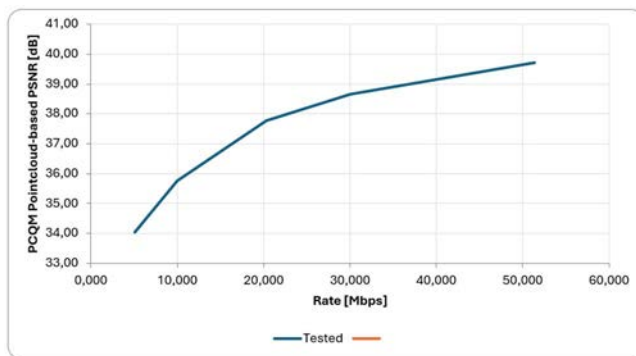


Figure 9.3.4.1.5.1.2-4: PCQM metric

9.3.4.1.5.1.3 Objective results of sequence Henry

The following 5 figures present the point-based metric results.

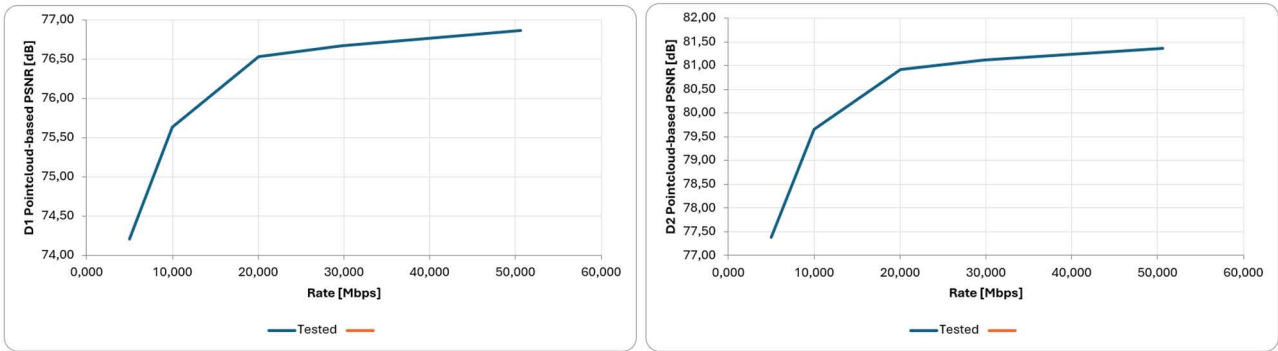


Figure 9.3.4.1.5.1.3-1: D1 and D2 metrics

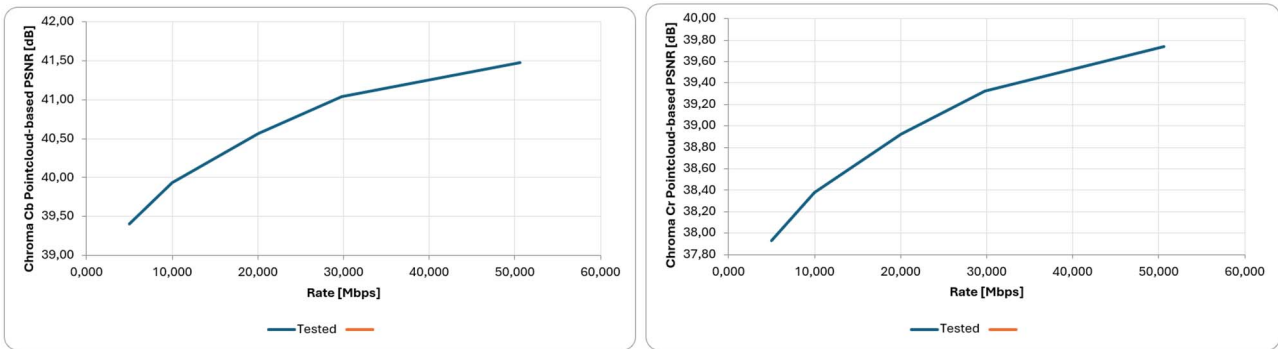


Figure 9.3.4.1.5.1.3-2: Cb and Cr metrics

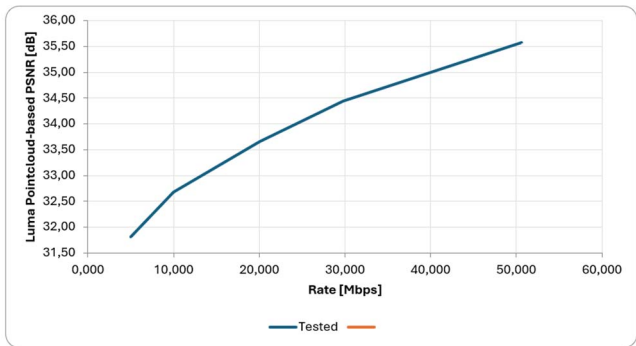


Figure 9.3.4.1.5.1.3-3: Luma metric

The following figure presents the PQCM metric results.

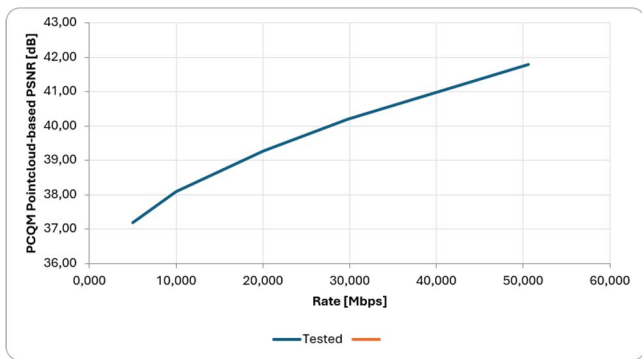


Figure 9.3.4.1.5.1.3-4: PCQM metric

9.3.4.1.5.1.4 Objective results of sequence Nathalie

The following 5 figures present the point-based metric results.

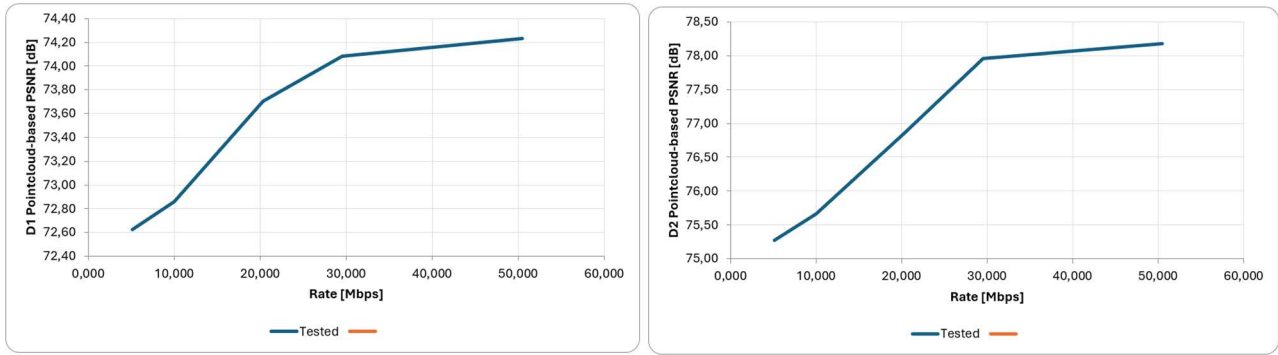


Figure 9.3.4.1.5.1.4-1: D1 and D2 metrics

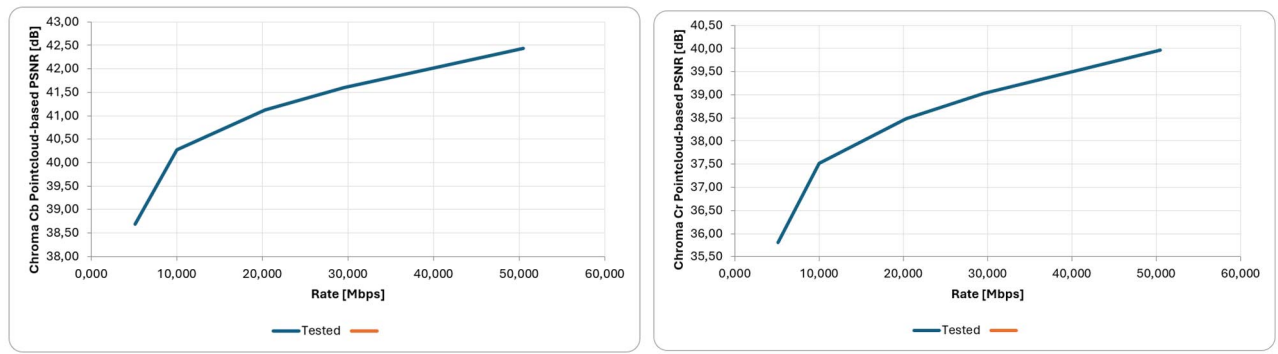


Figure 9.3.4.1.5.1.4-2: Cb and Cr metrics

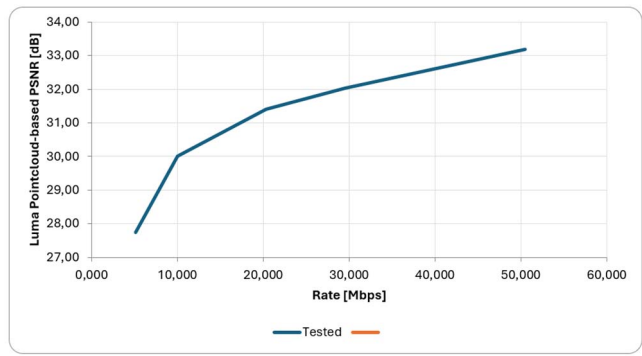


Figure 9.3.4.1.5.1.4-3: Luma metric

The following figure presents the PQCM metric results.

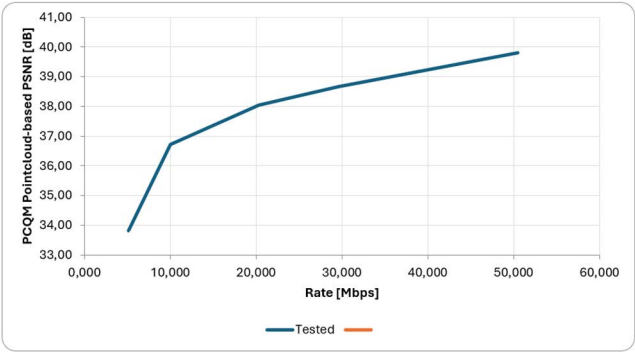


Figure 9.3.4.1.5.1.4-4: PCQM metric

9.3.4.1.5.1.5 Objective results of sequence Aliyah

The following 5 figures present the point-based metric results.

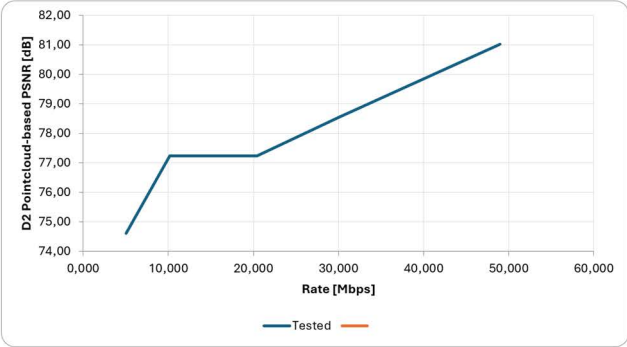
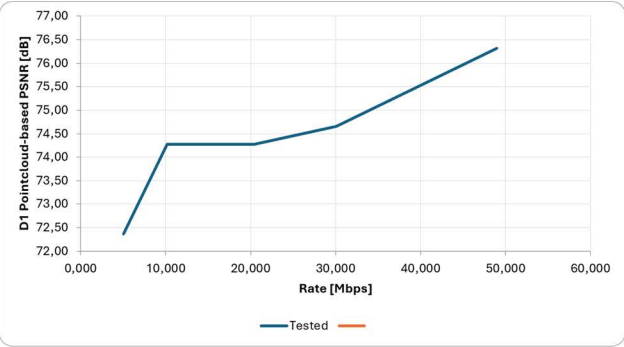


Figure 9.3.4.1.5.1.5-1: D1 and D2 metrics

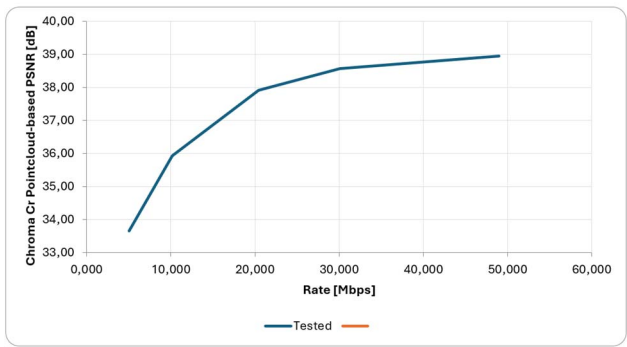
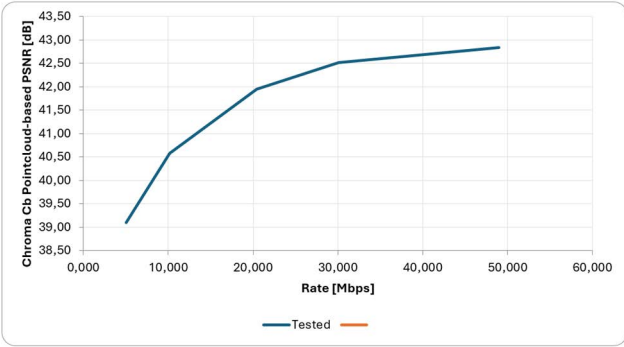


Figure 9.3.4.1.5.1.5-2: Cb and Cr metrics

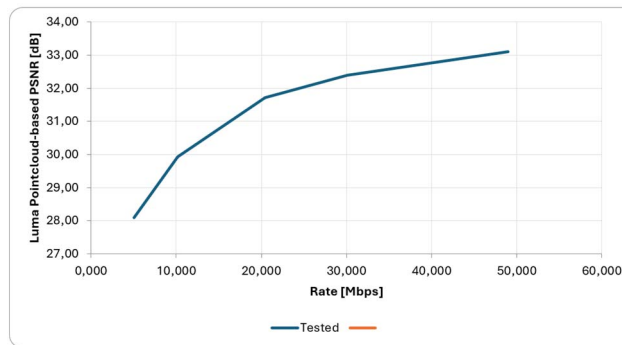


Figure 9.3.4.1.5.1.4-3: Luma metric

The following figure presents the PQCM metric results.

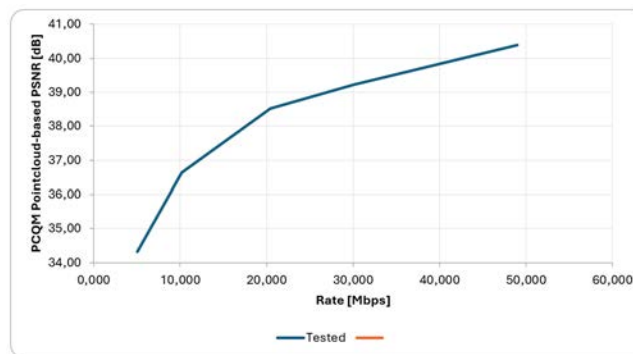


Figure 9.3.4.1.5.1.5-4: PCQM metric

9.3.4.1.5.1.6 Bitstream crosschecks

A spreadsheet summarizing the sequences, along with the corresponding originators and cross-checkers, is available in the directory Bitstreams/Scenario-2/V-PCC/Metrics.

9.3.4.1.5.2 Subjective evaluation

Videos are provided for all sequences and can be accessed as follows:

- Log into Aspera: <https://aspera.pub/I4tSQ8k>
- 3GPP members can request credentials by sending a request per email to: 3GPP_B2D_Datasets@interdigital.com
- Go to directory Bitstreams/Scenario-2/V-PCC/Videos. In the directory there is a zip file with the name of each sequence. It includes 18 videos for each sequence, coming from the 5 rate points plus one for the reference sequence and each of these 6 is rendered in 3 modes as described in clause 7.3.9.5. In the same directory is a file md5sum.txt which contains the md5 checksums.
- A dedicated camera path has been selected for each sequence which can be found in the directory Bitstreams/Scenario-2/V-PCC/camerapath

The videos can be visualized on e.g. a TV set and can be used for a viewing test at 3GPP member's premise.

9.3.4.1.5.3 External evaluation

9.3.4.1.5.3.1 External reports

For completeness the subjective verification test report for V-PCC organized by MPEG [141] is mentioned and the objective tests of V-PCC conducted by the Brazilian SBTVD Forum [152] is mentioned. For details see clause 7.3.10.

These reports use dense dynamic point clouds with lower resolution and higher V-PCC profiles than tests conducted in this report and the information is complementary.

9.3.4.1.5.3.2 Evaluation platform

5G-MAG hosts a V3C Immersive Platform [128]. It provides a Unity package to decode, render and play V-PCC encoded content in real time supporting off-the-shelf Android and Windows based consumer devices.

9.3.4.1.6 Network Requirements

NOTE: Documents required bitrates as well as possibly other aspects.

The performed evaluation did not analyze network requirements besides bitrates. It is referred to clause 11.

9.4 Scenario 3: Streaming of Multi-view plus depth Produced Content

9.4.1 Evaluation Overview

This section provides an overview of the evaluation process for the Streaming of Multi-view plus depth Produced Content scenario. Candidate solution HEVC Main10 MIV Main is presented in detail. Objective results are presented, as well as the commands for creating the pose trace videos.

9.4.2 Reference Sequences

The evaluation has been performed on the sequences presented in Annex C.4. Only the first 65 frames were used of each test sequence.

9.4.3 Performance Metrics

The performance metrics that were introduced in section 7.4.5 were selected.

9.4.4 Candidate Solutions

9.4.4.1 Solution 1: HEVC Main10 MIV Main

9.4.4.1.1 Introduction

The evaluation framework is available in the mvd/ folder of <https://github.com/5G-MAG/rt-beyond2d-evaluation-framework>, tag v0.1.0, under the 5G-MAG public license.

9.4.4.1.2 Reference Software

The software that has been used for the evaluation of the scenario is listed in Table 9.4.4.1.2-1. All software has been built from source using Python 3.12, LLVM 18.1.8 with help of the install.py script of TMIV, as follows:

```
# environment with python, clang and clang++ on the path
git clone https://gitlab.com/mpeg-i-visual/tmiv.git
cd tmiv
python -m venv venv
. venv/bin/activate
python -m pip install --upgrade pip
pip install -r requirements.txt
scripts/install.py clang-release
```

Table 9.4.4.1.2-1: Software used for the evaluation of the scenario

Software	URL	Version
Test model for MPEG immersive video (TMIV)	https://gitlab.com/mpeg-i-visual/tmiv	24.0
HEVC test model (HM)	https://vcgit.hhi.fraunhofer.de/jvet/HM	18.0
Quality metrics for immersive video (QMIV)	https://gitlab.com/mpeg-i-visual/qmiv	2.0

HM 18.0 and Kvazaar 2.3.1 have been compared in MPEG context for the coding of MIV video sub-bitstreams [170]. HM 18.0 was selected for this study because it has a better rate-distortion characteristic in general. However, because HM lacks support for delta QP maps, packed video support was disabled in TMIV.

9.4.4.1.3 Parameter Settings

For this study, content was encoded using TMIV and HM. Encoding of MIV bitstreams using TMIV and HM involves three steps:

1. Run the TMIV encoder to output a raw YUV video file for each video sub-bitstream, and a partial MIV bitstream with patch parameters and video parameters. The main work of the TMIV encoder is to prune pixels, patch patches, and generate atlas frames.
2. Run HM TAppEncoder to encode each YUV file.
3. Run the TMIV multiplexer to combine the partial MIV bitstream and the coded video sub-bitstream into a full MIV bitstream (a V3C sample stream).

All sequences have been encoded using the configurations in Table 9.4.4.1.3-1. The purpose of having multiple configurations is to illustrate the impact of pixel rate on rate-distortion characteristics. Because this is a new representation there is no anchor.

- The *full views* (FV) condition codes the texture and geometry video component of each view as a separate HEVC Main 10 video sub-bitstream. This condition gives the highest quality, but also the largest pixel rate and bit rate. It serves as an upperbound of what can be achieved with the current test sequences and software if pixel rate is not a concern. The MIV level depends on the input.
- The *MPEG MIV main* (A) condition is part of the MIV CTC anchor, defined in ISO/IEC JTC 1/SC 29/WG 04 N 0659. It results in two atlases, each with a texture and geometry component, thus resulting in four video sub-bitstreams. It causes TMIV to select a number of source views based on an available pixel budget. The resulting bitstreams have MIV level 3.5. Some source views are selected to be basic views and they are fully coded. Some other views are selected as semi-basic views and they are placed in full in the atlas, but then some patches can be placed on top. Finally there are additional views from which only patches are taken (Figure 9.4.4.1.5.1-1).
- The *Synthesize center view* (SCV) condition was designed for this study because the pixel rate of the MIV CTC may be too high for mobile devices. The atlas has a single synthesized center view plus patches of the source views. The aim of this condition is to provide a MIV level 2.5 result by lowering the pixel rate compared to the A condition (Figure 9.4.4.1.5.1-2).

Table 9.4.4.1.3-1: Encoder conditions

Condition	Profile	Level	Abbreviation	Directory name
Full views	HEVC Main 10	-	FV	config/full_views
MIV main anchor	HEVC Main 10 MIV 2 (FDIS 23090-12:—)	3.5	A	config/miv_main_anchor
Synthesize center view	HEVC Main 10 MIV Extended (23090-12:2023)	2.5	SCV	config/synthesize_center_view

Encoding was performed by running the encode.py script of TMIV with appropriate parameters. For all sequences the first 65 frames were encoded. It executes the TMIV Encoder, HM, and the TMIV Multiplexer with appropriate parameters. For example:

```
TMIV_DIR/bin/encode.py -i INPUT_DIR -o out -s D02 -n 65 \
-r RP0 -f 0 -v HM -j 4 -t TMIV_DIR \
--config-dir share/config \
```

```
-c config/synthesize_center_view/SCV_1_TMIV_encode.json \
-m config/synthesize_center_view/SCV_3_TMIV_mux.json \
-C share/config/hm/encoder_randomaccess_main10.cfg
```

The only substantial difference between the encoder conditions is the TMIV encoder configuration because the TMIV multiplexer configuration is trivial and the HM configuration is kept to the same random-access configuration for all conditions.

The rate point RP0 is a result without coding of the video sub-bitstreams that can be used to determine how much quality is lost by the pixel pruning prior to video coding. Rates RP1 .. RP4 correspond to the following QP values in Table 9.4.4.1.3-2. The same QP values are used for all sequences and all encoder conditions. The geometry QP is derived from the texture QP as done for the MIV CTC [162]. Hence, virtually no QP tuning has been performed at all.

Table 9.4.4.1.3-2: QP values for all sequences and encoder conditions

Rate point	Texture	Geometry
RP1	20	2
RP2	30	10
RP3	40	18
RP4	50	26

9.4.4.1.4 Distribution

9.4.4.1.5 Evaluation Results

9.4.4.1.5.1 Example atlas frames

The full views (FV) condition encodes each component of each view separately, e.g. resulting in 30 separate 1920 x 1080 videos for the Breakfast sequence. Figure 9.4.4.1.5.1-1 and Figure 9.4.4.1.5.1-2 provide examples of atlas frames for the MIV main anchor (A) and synthesize center view (SCV) conditions. A comparison of pixel rates is provided in Tab 9.4.4.1.5.2-1. Note that the size of each atlas depends on the sequence and on the encoding condition. This is because TMIV calculates the atlas frame size based on a number of inputs.

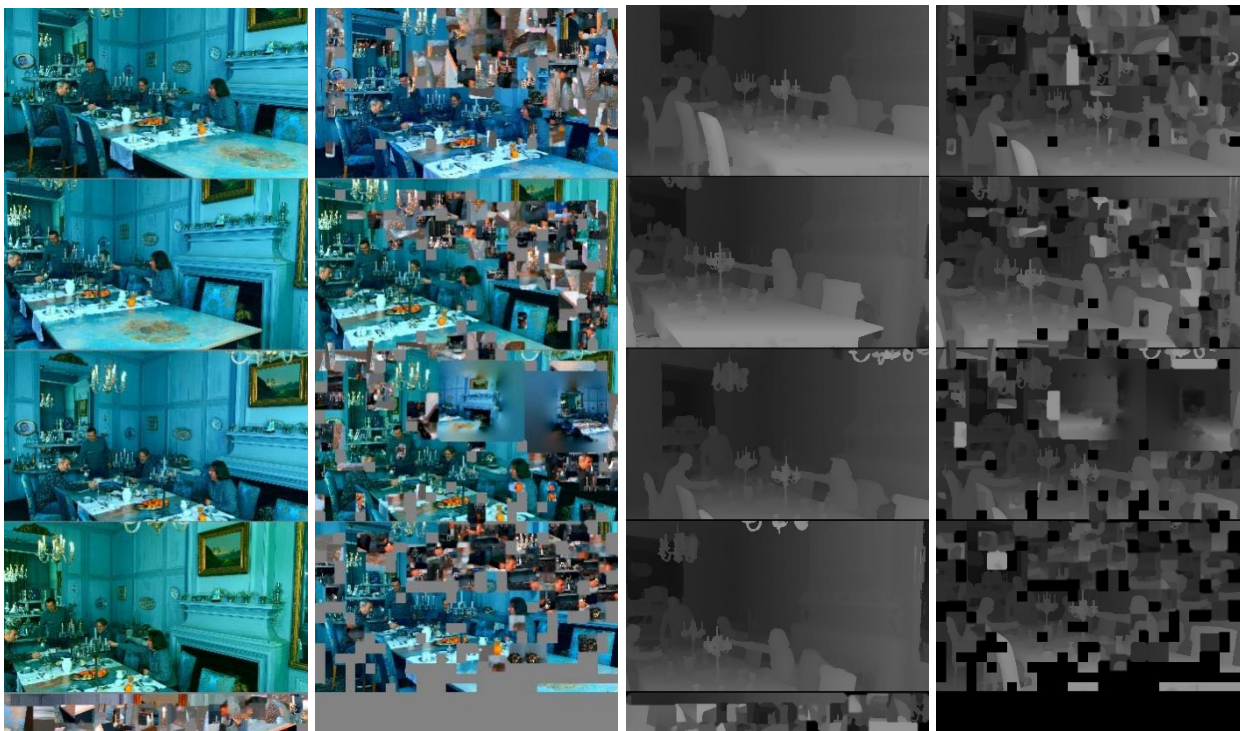


Figure 9.4.4.1.5.1-1: Video components of condition A with left to right: texture for atlas 0 and 1, geometry for atlas 0 and 1



Figure 9.4.4.1.5.1-2: Video components of condition SCV with left texture and right geometry

9.4.4.1.5.2 Pixel rate and MIV levels

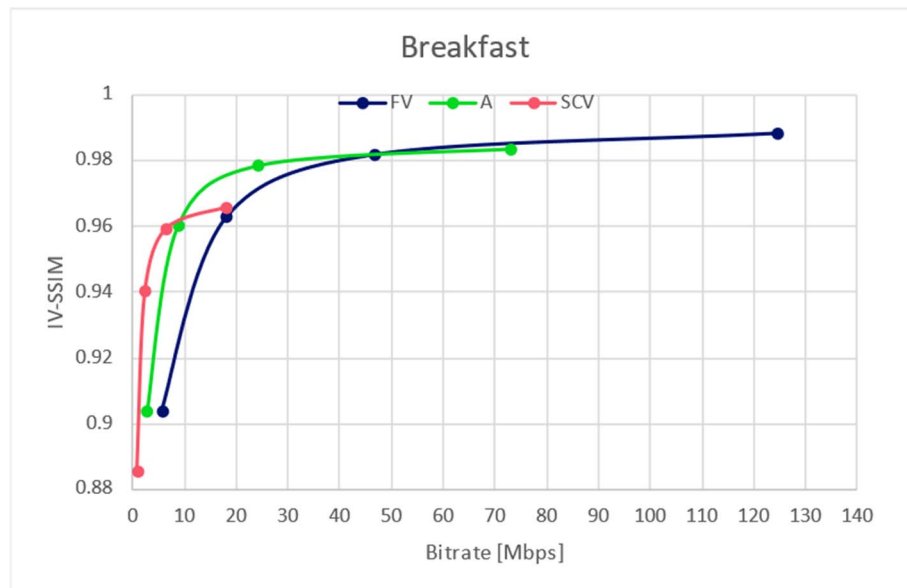
The pixel rates per video sub-bitstreams and the aggregate pixel rate are depicted in Table 9.4.4.1.5.2-1. The MIV level is based on the luma picture size and aggregate luma sample rate level limits as provided in ISO/IEC FDIS 23090-12:—Table A.7. Note that the MIV level for the FV condition is determined mainly by the aggregate luma sample rate because the luma picture size is only 1920 x 1080 but there are many video sub-bitstreams. Note that the coding of DanceMoves for condition A is inefficient because not all space in the atlases is used.

Table 9.4.4.1.5.2-1: Pixel rates for all sequences and conditions:

Condition	Sequence	Components	Sizes	Aggregate size (# luma samples)	Aggregate luma sample rate	MIV level
FV	Breakfast	15 x texture 15 x depth	1920 x 1080 1920 x 1080	62.2 M	1.87 G/s	3.0
FV	Bartender	21 x texture 21 x depth	1920 x 1080 1920 x 1080	87.1 M	2.61 G/s	3.5
FV	DanceMoves	6 x texture 6 x depth	1920 x 1080 1920 x 1080	24.9 M	0.373 G/s	2.0
A	Breakfast	2 x texture 2 x geometry	1920 x 4608 960 x 2304	22.1 M	0.664 G/s	2.5
A	Bartender	2 x texture 2 x geometry	1920 x 4608 960 x 2304	22.1 M	0.664 G/s	2.5
A	DanceMoves	2 x texture 2 x geometry	1920 x 4608 960 x 2304	22.1 M	0.664 G/s	2.5
SCV	Breakfast	1 x texture 1 x geometry	2880 x 2432 1440 x 1216	8.76 M	0.263 G/s	2.0
SCV	Bartender	1 x texture 1 x geometry	2944 x 2368 1472 x 1184	8.71 M	0.261 G/s	2.0
SCV	DanceMoves	1 x texture 1 x geometry	2048 x 3456 1024 x 1728	8.85 M	0.133 G/s	2.0

9.4.4.1.5.3

Rate-distortion characteristics

**Figure 9.4.4.1.5.3-1: Rate distortion curves for Breakfast for all three coding conditions**

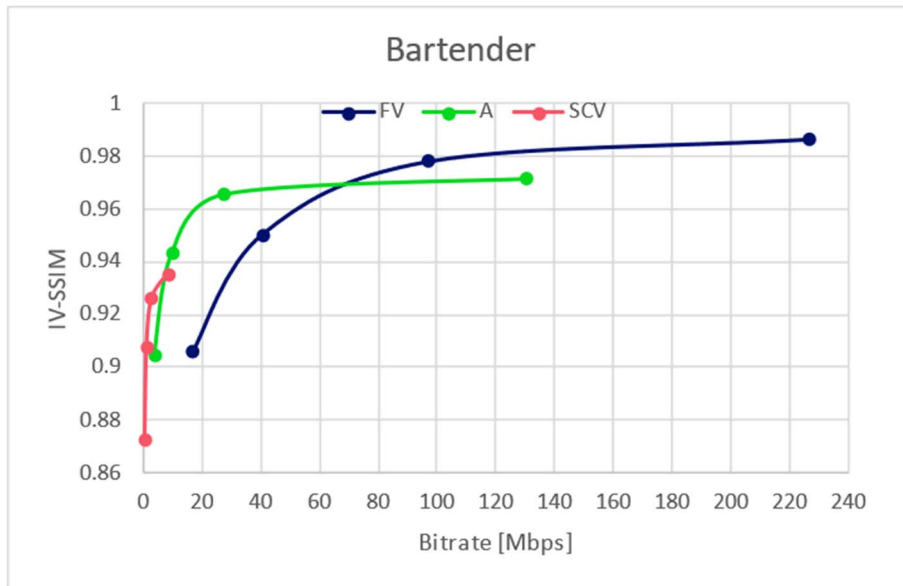


Figure 9.4.4.1.5.3-2: Rate distortion curves for Bartender for all three coding conditions

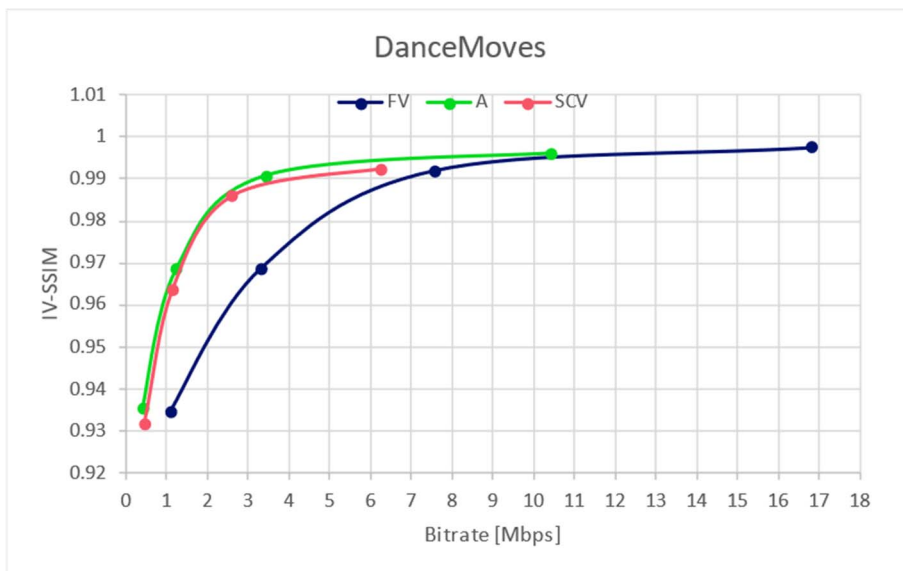


Figure 9.4.4.1.5.3-3: Rate distortion curves for DanceMoves for all three coding conditions

9.4.4.1.5.4 Pose trace videos

For each bitstream, that is for each sequence for each encoder condition and for each rate RP0 .. RP4, three pose trace videos have been rendered. A bitstream can be decoded and rendered using a command like this:

```
TMIV_DIR/bin/TmivDecoder -j 1 -n 32 -N 128 -s D02 -r RP3 -P p01 \
  -c config/synthesize_center_view/SCV_4_TMIV_decode.json \
  -p inputDirectory out -p outputDirectory out \
  -p configDirectory share/config
```

The decoder configurations differ only in path formats: there is no out-of-band information for RP1 .. RP4.

9.4.4.1.5.5 Availability of test data

The source video data (texture and depth), camera parameters, pose trace definitions, bitstreams and pose trace videos are available. For Breakfast, the information is hosted by InterDigital. For Bartender and DanceMoves, the information is hosted by Philips. Access will be provided to participants on request.

10 Gaps and Optimization Potential

10.1 Identified Gaps and Deficiencies with Video Capabilities

The Technical Report does not explicitly identify gaps or deficiencies with existing video capabilities in 3GPP standards needing immediate standardization. The focus of this study is primarily on evaluating existing and emerging Beyond 2D video representation formats where the principal findings are:

- Stereoscopic Video (see clause 4.3.2): A comprehensive evaluation has been performed revealing that:
 - Stereoscopic Video can be supported by existing content production workflows and captured by emerging mobile devices on the market. It also can be generated from monocular video using AI-based algorithms, decoding and rendering can be performed on the device. It has been observed that the industry is beginning to adopt this format.
 - The AI-based stereoscopic video generation methods often experience artifacts such as edge sharpness mismatch, cardboarding effects and crosstalk. These artifacts are especially noticeable along the contours of foreground objects in a scene. Therefore the objective metric for generated stereoscopic video emphasizing edge region evaluation has been described in clause 7.2.5.2.
 - Both Simulcast HEVC and MV-HEVC codecs have been tested for stereoscopic video.
 - Identified gap: While 3GPP specification TS 26.265 provides support for representation format and codec to support streaming of stereoscopic video content, the extension for stereoscopic representation formats and compression options is worth to be studied in future work.
- Dense Dynamic Point Clouds (see clause 4.3.3) and Dynamic Meshes (see clause 4.3.5): A comprehensive evaluation has been performed revealing that:
 - Dense dynamic point clouds and dynamic meshes are currently the primary representation formats to produce and deliver camera captured volumetric video. However, due to rather complex production systems (multiple or many calibrated cameras are needed), no widespread adoption of these workflows have yet happened, and the number of produced content items is limited. Work on MPEG Video-based Dynamic Mesh Coding V-DMC [55] for coding dynamic meshes was not finished at the closure date for the first version of the Technical Report and therefore the evaluation has been concentrated on dense dynamic point clouds with MPEG V-PCC [131] as the codec.
 - Dense dynamic point clouds with around 2 million points per frame encoded with MPEG V-PCC at 20 to 30 Mbit/s using the HEVC video codec provides a satisfying quality for the described scenario. See visual quality examples in clause 7.3.4.5 and videos in clause 9.X.4.1.5.2 subjective evaluation.
 - Rendering of dense dynamic point clouds can produce holes and needs special care. The effect of very simple cube based rendering and simple splat blend-based rendering is investigated, where the latter mitigates the problem. While rendering is considered an implementation aspect and is typically not normatively defined in coding standards, for proper user experience, sufficiently good rendering is essential. See impact of rendering in clause 4.3.3.3. Based on this, if consistent end-to-end quality is of concern for a service provider, potential requirements or recommendations on rendering performance may need to be defined.
 - Real-time decoder implementation of the MPEG V-PCC profile "HEVC Main10 V-PCC Basic Rec0" supporting up to 2-million-point points per frame on off the shelf consumer device has been demonstrated [128] re-using hardware-implemented HEVC decoders, but performance (CPU, power consumption) of such a hybrid SW/HW solution is for further study.
 - 3GPP provided a study on 6G use cases and services requirements in TR 22.870 [157]. Clause 9.12 of this report describes a use case on personalized interactive immersive guided tour, where assets represented as volumetric video are part of the scene.
 - Identified gap: There is no support in 3GPP specifications for representation formats and codecs to support streaming of professionally produced single asset volumetric video. However, services and applications

potentially wanting to use such representation formats may be implemented based on the hybrid HW/SW solution following the demonstration, and the existing HEVC-capabilities in 3GPP specifications would be sufficient to support the use case.

- Multi-view plus depth (see clause 4.3.4): A comprehensive evaluation has been performed revealing that:
 - Multi-view video enables the viewer to interact with the content by seamlessly moving and reorienting a virtual viewport. This serves two goals from the user perspective: it is possible to look around objects, and it is possible to freely choose a viewpoint. The latter is arguably less immersive, but enables the viewer to observe an action in more detail, i.e. by "being the director".
 - There is at least one codec (MIV) including publicly available encoding, decoding and rendering software that is capable of representing multi-view plus depth content. A real-time decoder and renderer for MIV is publicly available in 5G-MAG. A second codec (MV-HEVC) has been identified but has not been tested in this study.
 - By design multi-view plus depth content is not required to capture a scene from all angles. For instance, all cameras may be on one side of a scene, like in a studio for 2D video. A user has the freedom to move within a constrained volume (the viewing space).
 - There are no commercial end-to-end content production pipelines for this representation format (including tools for camera calibration and depth estimation), which hampers the adoption of the multi-view plus depth representation for practical applications.
 - Coding of realistic scenes needs a high luma sample rate to achieve an acceptable quality, resulting in higher HEVC levels or more HEVC decoders as compared to 2D video. The luma sample rate depends on the number of cameras and their separation, with specific examples provided in this report.
 - Identified gap: There is no support in 3GPP specifications for representation formats and codecs to support streaming of multi-view plus depth content.

10.2 Potential Requirements for New Video Capabilities

From the collecting scenario, future 3GPP standards may need to consider the following new video capability requirements:

- **Extensions for Stereoscopic Video:** The collected scenario one indicates a need for enhanced support for stereoscopic video formats to enable more immersive Beyond 2D experiences. This aspect has been addressed in TS 26.265 [11] and its potential next phase
- **Monitoring Market Adoption of New Beyond 2D Formats:** A comprehensive evaluation of emerging Beyond 2D formats, including point clouds, multi-view has been performed. The focus remains on continuous monitoring the market traction of these technologies, especially in content generation. Due to time constraints, Dynamic Mesh was not evaluated in this Technical Report and needs to be addressed in subsequent work. Immediate standardization is not required at this time and MPEG V-PCC and MPEG MIV remain candidate codecs for integration in 3GPP specifications in a future release. A potential future requirement for 3GPP is to define or support representations format(s) and codec(s) for streaming of produced volumetric video with single asset and for streaming of produced multiview plus depth video.
- **Gaussian Splatting:** 3DGS gets a lot of attention from academia and industry, but realistic use cases are not yet clear, the format is not yet stabilized and there is no codec from a recognized standards organization. Further study is needed and immediate standardization is not required.
- **AI-Generated Beyond 2D content:** The commercialization of AIGC has attracted attention from both academia and industry. This TR introduces AI-generated stereoscopic video, dynamic mesh and 4D content. However, further study is needed to improve quality and efficiency of AI generated content, as well as to develop the quality assessment methodologies.

10.3 Potential Network Optimizations

The network optimizations were not directly addressed in this study and potential needs can be inferred from the introduction of new video capabilities:

- **Transmission Efficiency:** More efficient transmission methods (e.g., protocols or distribution strategies) for these more complex video data.
- **Bandwidth Optimization and Network Capabilities:** Beyond 2D video technology involves processing, transmitting, and storing massive amounts of data over 3GPP networks, which presents significant challenges to both network bandwidth and user equipment (UE) computational capabilities. Therefore, exploring efficient network solutions and bandwidth optimization is critical for enabling real-time B2D video delivery across a wide range of viewing experiences without sacrificing the sense of immersion.

11 Conclusions and Proposed Next Steps

11.1 Summary and Conclusions

This technical report addresses the evolution of video services from traditional 2D formats to "beyond 2D" video, which includes immersive and interactive experiences based on stereoscopic 3D, multi-view plus depth, dense dynamic point clouds, dynamic meshes, and emerging research formats like Neural Radiance Fields (NeRF), light fields, and 3D Gaussian Splatting (3DGS). The report aims to evaluate the feasibility, performance, and interoperability of these formats and codecs within 3GPP services, considering implementation constraints and network requirements.

The report categorizes beyond 2D video formats as follows:

- **Stereoscopic 3D and Extensions:** Provides depth perception by presenting slightly different images to each eye. Widely supported by current devices and workflows, with extensions for higher resolutions and additional metadata (e.g., depth, alpha).
- **Multi-view Plus Depth:** Offers multiple synchronized camera views, optionally with depth maps, enabling free viewpoint navigation and immersive experiences.
- **Dense Dynamic Point Clouds:** Represents scenes or objects as high-density 3D points with attributes (color, normals, etc.), allowing detailed volumetric rendering.
- **Dynamic Meshes:** Uses vertices, edges, faces and attribute maps to define 3D geometry and texture, supporting animation and real-time rendering, commonly used for avatars and digital twins.
- **Light Fields, NeRF, and 3D Gaussian Splatting:** Advanced research formats that capture and render scenes with high realism and flexibility, though not yet standardized for commercial deployment.

A generic end-to-end reference model is introduced, covering content capture (via cameras or computer graphics), processing (conversion to beyond 2D formats), encoding (compression for efficient transmission), delivery (over 5G/6G networks), decoding, and rendering on various devices (smartphones, VR/AR headsets, autostereoscopic displays). The model emphasizes interoperability points and the need for systematic evaluation.

The report also defines a comprehensive evaluation and characterization framework, including:

- **Reference Scenarios:** Streaming of 1) UE-to-UE live stereoscopic video, 2) professionally produced volumetric video and 3) multi-view plus depth content, each with detailed workflows and constraints.
- **Performance Metrics:** Both objective (e.g. point-based PSNR and PCQM for point clouds, IV-SSIM for multi-view plus depth, HV3D for stereoscopic video) and subjective (user studies) metrics are used to assess quality, efficiency, and user experience. Subjective tests are enabled by the provision of videos allowing interested parties to conduct a formal subjective test, but 3GPP did not organize such a formal subjective test. External objective and subjective test reports are referenced where available.
- **Test Sequences:** A curated set of reference sequences (e.g., volumetric video represented as dense dynamic point cloud and dynamic mesh of people, multi-view scenes) are provided for benchmarking codecs and workflows.

Video codecs supported in existing 3GPP specifications (H.265/HEVC, MV-HEVC) are evaluated for their ability to support beyond 2D formats. The report identifies gaps in current capabilities, especially for new formats like point

clouds and dynamic meshes. Network requirements such as latency, bandwidth, and real-time processing are also discussed, with an emphasis on leveraging 5G capabilities.

The report concludes that certain beyond 2D video formats are maturing and becoming market-relevant, driven by advances in capture, production, compression, and display technologies.

11.2 Recommendations

Based on the evaluation in this document, the following aspects for B2D representation formats are recommended:

- 3GPP TS 26.265 defines representation formats to support stereoscopic video. Based on the conclusions in this document, considering extensions for stereoscopic representation formats including intrinsic and extrinsic camera parameters, depth, alpha and possibly improved colour subsampling formats is worthwhile to study in more details including stereoscopic capturing with optical systems on typical UE form factors and then to:
 - Identify relevant new representation formats not yet documented in TS 26.265 and provide the benefits in terms of user experience. The evaluation includes potential capturing and rendering of the formats. Candidates include support for alpha, support for depth together with stereo, additional color subsampling 4:2:2 or 4:4:4.
 - Study the feasibility of generating video signals following these representation formats on typical UE form factors, in particular smartphones based on existing and emerging optical systems
 - Identify compression options for the representation formats based on existing 3GPP codecs, in particular HEVC and MV-HEVC.
 - Identify the opportunities and needs to integrate the representation formats into different transport systems, including messaging, real-time communication, split rendering and streaming.
 - Define the expected traffic characteristics for new representation formats to meet certain quality thresholds.
 - Define a conformance environment, including hosting, tooling and process, as well as conforming test vectors to support operation points.
 - Identify gaps in existing specifications and provide guidance for potential normative work.
- For other representation formats, in particular dense dynamic point clouds, dynamic meshes and multi-view plus depth, it is recommended to continue monitoring the broader adoption of these formats. Once there is sufficient market traction, the baseline established in the study may be used to define the exact representation formats. This report also provides a good indication of how different HEVC profiles and features may be used to compress such formats. While these formats may provide some new experiences, there is no immediate necessity to add them to 3GPP specifications. Instead, the market should be monitored for traction of these technologies, particularly in content generation and broader availability of content.

As MPEG V-DMC [55] was finalized by MPEG at the closure date for the first version of this technical report, the evaluation of dynamic mesh and the codec MPEG V-DMC is not completed. Therefore, it is recommended to:

- Evaluate the dynamic mesh representation format with MPEG V-DMC and HEVC as underlying video codec by providing objective test results and by delivering videos enabling subjective testing.
- Study dynamic mesh content generation for offline productions in prosumer case (e.g. social media) and for real-time applications.

This recommendation is in line with the 3GPP study on 6G use cases and services requirements in TR 22.870 [157], where clause 9.12 describes a use case on personalized interactive immersive guided tour including assets represented as volumetric video as part of the scene.

- A particular format of interest is 3D Gaussian splats (3DGS) as introduced in clause 4.3.6.3. This format represents a 3D scene with a 3D Gaussian primitive, an anisotropic Gaussian ellipsoid. The rendering process is simple and can be executed in real time by projecting the sorted 3D Gaussian splats onto a screen and rendering them in a photorealistic manner. This results in real scenes rendered in real time with lighting and reflection effects, enhancing the realism of the rendered image. In addition, 3DGS has the potential to be generated with commonly available optical systems on existing and emerging smartphones, supported by AI-based workflows.

All of this makes the formats an attractive candidate, possibly in the 6G era, and further detailed study of this format is recommended including:

- Identification of the use cases for mobile devices that demonstrate the practical applications of 3DGS contents.
- A full definition and analysis of the 3DGS representation format including the relevance and complexity of the parameters of the primitives. This includes the size of the 3D model and the associated processing requirements, as well as quality and complexity aspects.
- Identify the opportunities to generated 3DGS content, in particular with existing optical systems and the ability to integrate AI-based workflows on device and/or in the network.
- Study the integration of such formats into 3GPP services (e.g. messaging), expected traffic characteristics as well as other aspects related to provide fully interoperable solutions.
- Develop an end-to-end reference implementation for content delivery, covering the entire pipeline from content creation on a server or capture on the UE and, through compression, transmission, to rendering on a mobile device platform.
- From network requirements perspective, current evaluations are limited to single-user cases, and some test sequences contain only a single asset (excluding complex scenes or multiple assets). In actual deployment, however, multi-user high-concurrency cases are anticipated, which means even compressed B2D video may demand more network resources than traditional 2D video to satisfy latency and bandwidth requirements. Therefore, the following aspects are recommended for the above-mentioned formats:
 - Study efficient network solutions and bandwidth optimization to enable real-time B2D video delivery across a wide viewing range and including potentially multiple-users and/or multiple assets without compromising perceptual immersion.
 - B2D-related features, such as AI-based stereoscopic video generation, require substantial computing power, which may exceed UE capabilities. To address this point, it is beneficial to investigate the feasibility of implementing these features (fully or partially) at the network level, thereby reducing computing latency and improving energy efficiency.

NOTE: AI-based solutions were used for the stereoscopic 3D content creation from a 2D asset only.

Annex A: Scenario Template

A.1 Introduction

This annex provides a proposed template to introduce a Scenario for Beyond 2D Video. This template has been used to collect the scenarios in this report. The text in blue corresponds to guidelines on the information to be provided with a scenario proposal.

A.2 Template

The following aspects are considered for a scenario:

1. Scenario name

2. Motivation for the scenario

What is the market relevance of the proposed scenario within the next few years? Are there any commercially available or pre-released products or prototypes?

Market relevance key indicators:

a. Technology evaluation on the market

Are there indications of pre-evaluation by service providers, device manufacturers, and/or network operators?

b. Industry activities

Is there relevant work in 3GPP MRPs, industry collaborations or among market stakeholders?

c. Production tools/companies

What is the availability of capturing setups, and production software? Are there endorsed formats for representation, contribution, compression, and storage? Is there an ecosystem of content creators?

d. Delivery solutions

Which delivery type is expected to be used? What are the expected transport formats? Is there SW or HW support and providers?

e. Content decoding and rendering

Is there decoding SW/HW support, and providers? Are there rendering devices and displays available yet?

3. Description of the scenario

This provides a description of beyond 2D video end-to-end workflows, which includes identifying and defining beyond 2D formats being used in the context and representation technologies to delivery these formats. The following aspects may be considered for each workflow:

a. Capturing and processing

b. Encoding

c. Packaging and delivery

d. Decoding

e. *Post-processing

f. Rendering

g. General constraints on latency, bandwidth, reliability and complexity

4. Supporting companies and 3GPP members

- a. *This documents the 3GPP members that support this scenario in terms of providing the information, test material, test requirements and the characterization for the tests. For each of the identified necessities, a tick box is created in the template.*
- b. *Preferably several 3GPP members are included in the support, and in addition a video service provider may be included (not necessarily a 3GPP member).*
- c. *Cross-verification is preferably done by the supporters of the scenario.*

5. Source format properties

This defines a clear range of the considered and relevant source formats, including the signal properties, but also the characteristics of the content. As an example, the texture and depth format properties of the source may be used which include:

- a. *Spatial resolutions*
- b. *Chroma Format*
- c. *Chroma Subsampling*
- d. *Aspect ratios*
- e. *Frame rates*
- f. *Colour space formats*
- g. *Transfer Characteristics*
- h. *Bit depth*
- i. *Viewpoints*
- j. *Other signal properties*

6. Encoding and decoding constraints and settings

Typical encoding constraints and settings such as:

- a) *Relevant Codec and Codec Profile/Levels according to 3GPP TS (e.g., TS 26.119),*
- b) *Random access frequency*
- c) *Error resiliency requirements*
- d) *Bitrates and quality requirements*
- e) *Bitrate parameters (CBR, VBR, CAE, HRD parameters)*
- f) *ABR encoding requirements (switching frequency, etc.)*
- g) *Latency requirements and specific encoding settings*
- h) *Encoding context: real-time encoding, on device encoding, cloud-based encoding, offline encoding, etc.*
- i) *Required decoding capabilities*
- j) *Synchronization requirements*

7. Performance Metrics and Requirements

- a. *A clear definition on how the performance needs to be evaluated including metrics, etc addressing the main KPIs of the scenario.*
- b. *Objective measures such as PSNR, VMAF, etc, may be used*

c. Justification on whether objective metrics are sufficient and representative of the subjective performance.

8. Interoperability Considerations for the application

a. Streaming with DASH/HLS/CMAF/QUIC

b. RTP based delivery

9. Test Sequences

A set of selected test sequences that are provided by the proponents in order to do the evaluation. They should cover a set of source format properties

10. Detailed test conditions

Provides a proposal for detailed test conditions, for example based on a reference software together with the sequences and configuration parameters.

11. External Performance data

References to external performance data that can be added, for example other SDOs, public documents and so on.

12. Additional Information

a. Industry activities

Is there Relevant work in industry forums?

b. Implementation constraints

Are there any indications about scalability of the technology with regards to network and devices?

c. Innovation

Does the technology address a current or a future need on the market? Can it potentially disrupt existing markets?

Annex B: Data Formats and Metrics

B.1 Introduction

This Annex provides a detailed overview on data formats and their usage for metrics computation.

B.2 Raw Video Sequences

B.2.1 Overview

For the raw video sequences used in Scenario 2 and Scenario 3, the JSON schema is defined in Annex B.2.2 and B.2.3.

B.2.2 JSON Schema

JSON schema for the raw format is here

< <https://dash-large-files.akamaized.net/WAVE/3GPP/Beyond2D/ReferenceSequence/raw-schema.json> >

```
1. {
2.   "Sequence": {
3.     "Name": "Example",
4.     "Background": "This is a B2DV format example",
5.     "Scenario": "On-demand",
6.     "Key": "Identifier",
7.     "TR26.956": "Annex X.Y.Z"
8.   },
9.   "Views": [
10.    {
11.      "ViewId": "v0",
12.      "Extrinsics": {
13.        "orientation": {
14.          "qw": 0.9999915361,
15.          "qx": 0.0024327517,
16.          "qy": 0.0024349121,
17.          "qz": -0.0022688841
18.        },
19.        "position": [
20.          -0.0006123598,
21.          0.3035059273,
22.          0.0012498678
23.        ]
24.      },
25.      "Intrinsics": {
26.        "focalLength": 1002.349976,
27.        "principalPoint": {
28.          "horizontalNorm": 960.0,
29.          "vertical": 540.0
30.        }
31.      },
32.      "ProjectionPlaneSize": {
```



```
33.         "columnCount": 1920,
34.         "rowCount": 1080
35.     },
36.     "Quantization": {
37.         "highNormDisp": 2.000000,
38.         "lowNormDisp": 0.200000
39.     },
40.     "Components": [
41.         {
42.             "ComponentId": "texture",
43.             "Data": {
44.                 "URI": "https://dash-large-files.akamaized.net/WAVE/3GPP/some/url/file.yuv",
45.                 "md5": "e537665c18e32bbaf8e5e9d63e18dd2c",
46.                 "thumbnail": "https://dash-large-files.akamaized.net/WAVE/3GPP/some/url/file.png",
47.                 "preview": "https://dash-large-files.akamaized.net/WAVE/3GPP/some/url/file.mp4",
48.                 "size": 7962624000,
49.                 "md5-10": "1c3550197120f95502c4add38d7ebd33"
50.             },
51.             "Properties": {
52.                 "width": 1920,
53.                 "height": 1080,
54.                 "format": "yuv",
55.                 "packing": "planar",
56.                 "scan": "progressive",
57.                 "subsampling": "420",
58.                 "bitDepth": 8,
59.                 "frameRate": 30,
60.                 "colourPrimaries": "1",
61.                 "transferCharacteristics": "1",
62.                 "matrixCoefficients": "1",
63.                 "sampleAspectRatio": "1",
64.                 "duration": 10,
65.                 "frameCount": 600,
66.                 "startFrame": 1,
67.                 "videoFullRangeFlag": "0",
68.                 "chromaSampleLocType": "0"
69.             }
70.         },
71.         {
72.             "ComponentId": "depth",
73.             "Data": {
74.                 "URI": "https://dash-large-files.akamaized.net/WAVE/3GPP/some/url/file.yuv",
75.                 "md5": "e537665c18e32bbaf8e5e9d63e18dd2c",
76.                 "thumbnail": "https://dash-large-files.akamaized.net/WAVE/3GPP/some/url/file.png",
77.                 "preview": "https://dash-large-files.akamaized.net/WAVE/3GPP/some/url/file.mp4",
78.                 "size": 7962624000,
79.                 "md5-10": "1c3550197120f95502c4add38d7ebd33"
80.             },
81.             "Properties": {
82.                 "width": 1920,
83.                 "height": 1080,
84.                 "format": "yuv",
85.                 "packing": "planar",
86.                 "scan": "progressive",
87.                 "subsampling": "420",
```

```

88.         "bitDepth": 16,
89.         "frameRate": 30,
90.         "colourPrimaries": "2",
91.         "transferCharacteristics": "8",
92.         "matrixCoefficients": "0",
93.         "sampleAspectRatio": "1",
94.         "duration": 10,
95.         "frameCount": 600,
96.         "startFrame": 1,
97.         "videoFullRangeFlag": "1",
98.         "chromaSampleLocType": "0"
99.     }
100. }
101. ]
102. }
103. ],
104. "copyright": "Conditions that are suitable for this study",
105. "contact": {
106.     "Name": "Bart Kroon",
107.     "Company": "Philips",
108.     "e-mail": "bart.kroon@philips.com",
109.     "generation": "provided by contact"
110. }
111. }

```

B.2.3 JSON Scheme for Dense Dynamic Point Cloud

JSON schema for the raw dense dynamic point cloud format is here:

<<https://dash-large-files.akamaized.net/WAVE/3GPP/Beyond2D/ReferenceSequences/raw-schema-ddpc.json> >

```

1. {
2.     "$schema": "http://json-schema.org/draft-07/schema",
3.     "$id": "https://dash-large-files.akamaized.net/WAVE/3GPP/Beyond2D/ReferenceSequences/B2DPC-schema.json",
4.     "type": "object",
5.     "title": "The root schema",
6.     "description": "Schema for beyond 2D point cloud sequences.",
7.     "default": {},
8.     "required": [
9.         "Sequence",
10.        "Properties",
11.        "CopyRight",
12.        "Contact"
13.    ],
14.    "properties": {
15.        "Sequence": {
16.            "$id": "#/properties/Sequence",
17.            "type": "object",
18.            "title": "Schema for the Sequence",
19.            "description": "Includes all information about the sequence",
20.            "default": {},
21.            "required": [
22.                "Name",
23.                "Scenario",
24.                "Key"
25.            ],

```

```
26.     "properties": {
27.         "Name": {
28.             "$id": "#/properties/Sequence/properties/Name",
29.             "type": "string",
30.             "title": "The Name schema",
31.             "description": "Provides a unique name of the sequence."
32.         },
33.         "Background": {
34.             "$id": "#/properties/Sequence/properties/Background",
35.             "type": "string",
36.             "title": "The Background schema",
37.             "description": "Provides a background information on the sequence."
38.         },
39.         "Scenario": {
40.             "$id": "#/properties/Sequence/properties/Scenario",
41.             "type": "string",
42.             "title": "The Scenario schema",
43.             "description": "Provides information to which Scenario this sequence relates."
44.         },
45.         "Key": {
46.             "$id": "#/properties/Sequence/properties/Key",
47.             "type": "string",
48.             "title": "The Key schema",
49.             "description": "Provides the key used in TR 26.956.",
50.             "examples": [
51.                 "S41"
52.             ]
53.         },
54.         "TR26.956": {
55.             "$id": "#/properties/Sequence/properties/TR26.956",
56.             "type": "string",
57.             "title": "The TR26.956 schema",
58.             "description": "Provides the reference to TR26.956 used.",
59.             "examples": [
60.                 "S41"
61.             ]
62.         }
63.     },
64.     "additionalProperties": true
65. },
66. "Properties": {
67.     "$id": "#/properties/Properties",
68.     "type": "object",
69.     "title": "The Properties schema",
70.     "description": "the properties of the raw video.",
71.     "required": [
72.         "URI",
73.         "NameFormat",
74.         "frameCount",
75.         "startFrame",
76.         "frameRate",
77.         "geometryPrecision",
78.         "colorformat",
79.         "peak"
80.     ],
```

```

81.         "properties": {
82.             "URI": {
83.                 "$id": "#/properties/Properties/properties/URI",
84.                 "type": "string",
85.                 "title": "The URI schema",
86.                 "description": "Provides a reference/URL to the sequence point cloud data."
87.             },
88.             "thumbnail": {
89.                 "$id": "#/properties/Properties/properties/thumbnail",
90.                 "type": "string",
91.                 "title": "The thumbnail schema",
92.                 "description": "Provides a reference/URL to a typical frame of the video."
93.             },
94.             "preview": {
95.                 "$id": "#/properties/Properties/properties/preview",
96.                 "type": "string",
97.                 "title": "The preview schema",
98.                 "description": "Provides a reference/URL to an mp4 encoded video."
99.             },
100.            "NameFormat": {
101.                "$id": "#/properties/Properties/properties/NameFormat",
102.                "type": "string",
103.                "title": "The NameFormat schema",
104.                "description": "Provides a name of the sequence to which a frame index is added.",
105.                "examples": [
106.                    "exemple%04d.ply"
107.                ]
108.            },
109.            "frameCount": {
110.                "$id": "#/properties/Properties/properties/frameCount",
111.                "type": "integer",
112.                "title": "The frameCount schema",
113.                "description": "The number of point cloud frames in the sequence.",
114.                "examples": [
115.                    327
116.                ]
117.            },
118.            "startFrame": {
119.                "$id": "#/properties/Properties/properties/startFrame",
120.                "type": "integer",
121.                "title": "The startFrame schema",
122.                "description": "The first frame in the sequence that is to be used starting from 1.",
123.                "default": 1,
124.                "examples": [
125.                    1
126.                ]
127.            },
128.            "frameRate": {
129.                "$id": "#/properties/Properties/properties/frameRate",
130.                "type": "integer",
131.                "title": "The frameRate schema",
132.                "description": "Framerate of the video.",
133.                "examples": [
134.                    30
135.                ]

```

```
136.         },
137.         "pointsCountMean": {
138.             "$id": "#/properties/Properties/properties/pointsCountMean",
139.             "type": "integer",
140.             "title": "The average of points number per frame schema",
141.             "description": "The average point number per frame.",
142.             "examples": [
143.                 857241
144.             ]
145.         },
146.         "geometryPrecision": {
147.             "$id": "#/properties/Properties/properties/geometryPrecision",
148.             "type": "integer",
149.             "title": "The geometry procision schema",
150.             "description": "geometry precision in bits of the point cloud sequence.",
151.             "default": "10",
152.             "examples": [
153.                 10
154.             ]
155.         },
156.         "colorformat": {
157.             "$id": "#/properties/Properties/properties/colorformat",
158.             "type": "string",
159.             "title": "The colorformat schema",
160.             "description": "Format of texture attribute",
161.             "default": "rgb",
162.             "enum": [
163.                 "rgb",
164.                 "yuv"
165.             ],
166.             "examples": [
167.                 "rgb"
168.             ]
169.         },
170.         "peak": {
171.             "$id": "#/properties/Properties/properties/peak",
172.             "type": "integer",
173.             "title": "The peak schema",
174.             "description": "peak of the point cloud sequence.",
175.             "default": "1023",
176.             "enum": [
177.                 1023,
178.                 2047
179.             ],
180.             "examples": [
181.                 1023
182.             ]
183.         },
184.         "duration": {
185.             "$id": "#/properties/Properties/properties/duration",
186.             "type": "number",
187.             "title": "The duration schema",
188.             "description": "Duration of the sequence in seconds.",
189.             "examples": [
190.                 5.46
```

```
191.         ]
192.     },
193.     "md5": {
194.         "$id": "#/properties/Properties/properties/md5",
195.         "type": "string",
196.         "title": "The md5 string",
197.         "items": {
198.             "description": "md5 of the zip files containing all point cloud frames.",
199.             "type": "string",
200.             "examples": [
201.                 "d055a94f35f7594776186fc5d09a9fa4"
202.             ]
203.         }
204.     },
205.     "size": {
206.         "$id": "#/properties/Properties/properties/size",
207.         "type": "integer",
208.         "title": "The size integer",
209.         "items": {
210.             "description": "Size in bytes of the zip file containing all point cloud frames.",
211.             "type": "integer",
212.             "examples": [
213.                 149190147
214.             ]
215.         }
216.     }
217. },
218. "additionalProperties": true
219. },
220. "CopyRight": {
221.     "$id": "#/properties/CopyRight",
222.     "type": "string",
223.     "title": "The CopyRight schema",
224.     "description": "Copyright statement."
225. },
226. "Contact": {
227.     "$id": "#/properties/Contact",
228.     "type": "object",
229.     "title": "The Contact schema",
230.     "description": "A contact for the sequence.",
231.     "required": [
232.         "Name"
233.     ],
234.     "properties": {
235.         "Name": {
236.             "$id": "#/properties/Contact/properties/Name",
237.             "type": "string",
238.             "title": "The Name schema",
239.             "description": "The name of a person."
240.         },
241.         "Company": {
242.             "$id": "#/properties/Contact/properties/Company",
243.             "type": "string",
244.             "title": "The Company schema",
245.             "description": "Company."
```

```
246.         },
247.         "e-mail": {
248.             "$id": "#/properties/Contact/properties/e-mail",
249.             "type": "string",
250.             "title": "The e-mail schema",
251.             "description": "e-mail or web page link."
252.         },
253.         "generation": {
254.             "$id": "#/properties/Contact/properties/generation",
255.             "type": "string",
256.             "title": "The generation schema",
257.             "description": "Information on how the data was generated"
258.         }
259.     },
260.     "additionalProperties": true
261. }
262. },
263. "additionalProperties": true
264. }
```

Annex C: Reference Sequences

C.1 Introduction

This annex provides a summary of candidate reference sequences that were discussed to be potentially suitable for one or multiple of the scenarios introduced in clause 6 of this Technical Report. For each candidate reference sequence, at least the following information is provided.

- A summary of the sequence characteristics
- A screenshot of the sequence
- Source sequence properties
- Information where the source sequence is hosted
- Copyright and license information

The content is provided in JSON files here: <https://dash-large-files.akamaized.net/WAVE/3GPP/Beyond2D/ReferenceSequence>. The format of the reference sequences follows the proposed format in Annex B.2.

The sequences are summarized here: <https://dash-large-files.akamaized.net/WAVE/3GPP/Beyond2D/ReferenceSequences/sequences.csv>.

C.2 Test Sequences for Volumetric Video with single asset containing people

C.2.1 Overview

This annex presents candidate test sequences that are available for testing. Some sequences have been made freely available to 3GPP under license agreement but cannot be made publicly available. Some sequences are not free but can be publicly purchased by those who need to work with the source sequences. Yet other sequences are free and publicly available for download by respecting the license.

NOTE: Sequences from Volucap and XD Productions that are freely available to 3GPP members have been converted to the dense dynamic point cloud representation format with around 2 million points per frame.

C.2.2 Juggle Soccer test sequence

C.2.2.1 Description

Soccer player with red shirt is showing soccer tricks with a ball. Particularity with the sequence is that a moving person and a ball are captured in one asset.



Figure C.2.2.1-1 Juggle Soccer - content courtesy XD Productions

C.2.2.2 Sequence properties

The tables C.2.2.2-1 and C.2.2.2-2 summarize the properties of the Juggle Soccer sequence

Table C.2.2.2-1 Juggle Soccer sequence properties dense dynamic point cloud

Parameter	Value
Frame rate	25
#frames	125
Mean #point / frame	1.883.637
Attributes	RGB
Normals	Yes
Geometry Precision	11
Attribute Precision	8
Normal Precision	Float

Table C.2.2.2-2 Juggle Soccer sequence properties dynamic mesh

Parameter	Value
Frame rate	25
#triangles per frame	80K
Texture resolution	4K
#frames	125

The sequence can be accessed: <https://aspera.pub/I4tSQ8k>

3GPP members can request credentials by sending a request per email to: 3GPP_B2D_Datasets@interdigital.com

C.2.2.3 Copyright and license information

XD Productions[143] kindly made this sequence freely available for 3GPP internal usage under license. License XD_Productions_-_InterDigital_Content_license_3GPP is provided in the directory with the sequence.

C.2.3 Mitch test sequence

C.2.3.1 Description

Mitch is slacklining with slow movements allowing to check preserved details in tissue of the shirt and in the face.



Figure C.2.3.1-1 Mitch - content courtesy Volucap

C.2.3.2 Sequence properties

The tables C.2.3.2-1 and C.2.3.2-2 summarize the properties of the Mitch sequence

Table C.2.3.2-1 Mitch sequence properties dense dynamic point cloud

Parameter	Value
Frame rate	25
#frames	475
Mean #point / frame	1.787.791
Attributes	RGB
Normals	Yes
Geometry Precision	11
Attribute Precision	8
Normal Precision	Float

Table C.2.3.2-2 Mitch sequence properties dynamic mesh

Parameter	Value
Frame rate	25
#triangles per frame	30K
Texture resolution	4K
#frames	475

The sequence can be accessed: <https://aspera.pub/I4tSQ8k>

3GPP members can request credentials by sending a request per email to: 3GPP_B2D_Datasets@interdigital.com

C.2.3.3 Copyright and license information

Volucap [136] kindly made this sequence freely available for 3GPP internal usage under license. License “License_Volucap_T097_Mitch2.1-05” is provided in the directory with the sequence.

C.2.4 Thomas test sequence

C.2.4.1 Description

Thomas is waiting and performing slow body and hands/arms movements allowing to check for preserved details in tissue of the shirt and in the face.



Figure C.2.4.1-1 Thomas - content courtesy Volucap

C.2.4.2 Sequence properties

Table C.2.4.2-1 Thomas sequence properties dynamic mesh

Parameter	Value
Frame rate	25
#triangles per frame	30K
Texture resolution	4K
#frames	748

The sequence can be accessed: <https://aspera.pub/I4tSQ8k>

3GPP members can request credentials by sending a request per email to: 3GPP_B2D_Datasets@interdigital.com

C.2.4.3 Copyright and license information

Volucap [136] kindly made this sequence freely available for 3GPP internal usage under license. License “License_Volucap_T003_ThomasScenic-03” is provided in the directory with the sequence.

C.2.5 Nathalie test sequence

C.2.5.1 Description

Nathalie is performing a classic dance, as such the sequence is dynamic.



Figure C.2.5.1-1 Nathalie - content courtesy Volucap

C.2.5.2 Sequence properties

The tables C.2.5.2-1 and C. 2.5.2-2 summarize the properties of the Mitch sequence

Table C.2.5.2-1 Nathalie sequence properties dense dynamic point cloud

Parameter	Value
Frame rate	30
#frames	925
Mean #point / frame	1.641.098
Attributes	RGB
Normals	Yes
Geometry Precision	11
Attribute Precision	8
Normal Precision	Float

Table C.2.5.2-2 Nathalie sequence properties dynamic mesh

Parameter	Value
Frame rate	30
#triangles per frame	30K
Texture resolution	4K
#frames	925

The sequence can be accessed: <https://aspera.pub/I4tSQ8k>

3GPP members can request credentials by sending a request per email to: 3GPP_B2D_Datasets@interdigital.com

C.2.5.3 Copyright and license information

Volucap [136] kindly made this sequence freely available for 3GPP internal usage under license. License “AOM_License Volucap_rp_nathalie_4d_001_dancing-20211214_Gsplats” is provided in the directory with the sequence.

C.2.6 Steam Roller test sequence

C.2.6.1 Description

Steam Roller is a performance on a BMX bike. Particularity with the sequence is that a moving person and an object (bicycle) are captured in one asset and that there are fast movements.



Figure C.2.6.1-1 Steam Roller - content courtesy Volucap

C.2.6.2 Sequence properties

The table 2.6.2 summarizes the properties of the Steam Roller sequence

Table C.2.6.2-1 Steam Roller sequence properties dynamic mesh

Parameter	Value
Frame rate	30
#triangles per frame	70K
Texture resolution	8K
#frames	493

The sequence can be accessed: <https://aspera.pub/I4tSQ8k>

3GPP members can request credentials by sending a request per email to: 3GPP_B2D_Datasets@interdigital.com

C.2.6.3 Copyright and license information

Volucap [136] kindly made this sequence freely available for 3GPP internal usage under license. License “AOM_License Volucap_Rec030_Steam_Roller_no_hands_Gsplats” is provided in the directory with the sequence.

C.2.7 Aliyah test sequence

C.2.7.1 Description

Aliyah is performing a modern dance, as such the sequence is pretty dynamic.



Figure C.2.7.1-1 DancingAliyah - content courtesy Renderpeople

C.2.7.2 Sequence properties

The tables 2.7.2-1 and 2.7.2-2 summarize the properties of the Aliyah sequence

Table C.2.7.2-1 Aliyah sequence dense dynamic point cloud

Parameter	Value
Frame rate	30
#frames	1112
Mean #point / frame	1.732.973
Attributes	RGB
Normals	Yes
Geometry Precision	11
Attribute Precision	8
Normal Precision	Float

Table C.2.7.2-2 Aliyah sequence properties dynamic mesh

Parameter	Value
Frame rate	30
#triangles per frame	30K
Texture resolution	4K
#frames	1112

Renderpeople [144] provides a free and publicly downloadable “4D People” source sequence under license. This source sequence is provided in file formats for 3ds Max, Maya, Blender, Cinema 4D and Alembic and can be stored or converted to mesh or dense point cloud format.

The sequence can be accessed: <https://renderpeople.com/free-3d-people/>
Select then the free sequence under 4D People

C.2.7.3 Copyright and license information

General terms and conditions can be found here: <https://renderpeople.com/general-terms-and-conditions/>

C.2.8 Henry test sequence

C.2.8.1 Description

Henry is performing a stretching exercise, as such the sequence is dynamic.



Figure C.2.8.1-1 Henry - content courtesy Renderpeople

C.2.8.2 Sequence properties

The tables 2.8.2-1 and 2.8.2-1 summarizes the properties of the Henry sequence

Table C.2.8.2-1 Henry sequence properties dense dynamic point cloud

Parameter	Value
Frame rate	30
#frames	733
Mean #point / frame	1.773.110
Attributes	RGB
Normals	Yes
Geometry Precision	11
Attribute Precision	8
Normal Precision	Float

Table C.2.8.2-2 Henry sequence properties dynamic mesh

Parameter	Value
Frame rate	30
#triangles per frame	30K
Texture resolution	4K
#frames	733

Renderpeople[145] provides a catalogue of currently 130 “4D People” under license and the catalog is growing. These source sequences are provided in file formats for 3ds Max, Maya, Blender, Cinema 4D and Alembic and can be stored or converted to mesh or dense point cloud format. Sequences from the 4D catalog are not free and need to be purchased. Henry is one of the sequences in the catalog that has been picked up as it is dynamic and different from the other presented sequences.

The “4D People” shop is accessible here: https://renderpeople.com/3d-people/?_product=4d-people

C.2.8.3 Copyright and license information

General terms and conditions can be found here: <https://renderpeople.com/general-terms-and-conditions/>

C.2.9 Ultra Video Group of Tampere University test sequences

C.2.9.1 Description

Ultra Video Group of Tampere University [146] kindly provides 12 downloadable sequences @25 fps under license.

Detailed descriptions and thumbnails are directly available on the website and are not reproduced here.

C.2.9.2 Sequence properties

All sequences can be downloaded as dense dynamic point cloud or as dynamic mesh. Different quality levels are proposed when downloading.

The Table C.2.9.2-1 summarizes the properties of the UVG sequences

Table C.2.9.2-1 UWG sequences properties dynamic point cloud

Parameter	Value
Frame rate	25
#frames	250
Mean #point / frame	Depends per sequence and on geometry precision
Attributes	RGB
Normals	Yes
Geometry Precision	9, 10, 11 and 12 bit
Attribute Precision	8
Normal Precision	9, 10, 11 or 12 bit integer

C.2.9.3 Copyright and license information

The license agreement can be found here: https://ultravideo.fi/UVG-VM/UVG-VM_LICENSE_AGREEMENT.pdf

C.2.10 OwlII Inc test sequences

C.2.10.1 Description

OwlII Inc [44]. makes available 2 dynamic human mesh sequences *basketball_player* and *dancer*.



Figure C.2.10.1-1 basketball_player and dancer - OwlII Inc

C.2.10.2 Sequence properties

All sequences can be downloaded as dynamic mesh. The Table C.2.10.1-1 summarizes the properties of the OwlII Inc sequences.

Table C.2.10.1-1 OwlII Inc sequences properties dynamic mesh

Test material dataset filename	#Frames	#Vertices	#Faces	Geometry Precision	Texture Coord. Precision	Texture Map Size
basketball_player [44]	600	20k	40k	12 bits	12 bits	2k x 2k
dancer [44].	600	20k	40k	12 bits	12 bits	2k x 2k

The sequence can be accessed:

- OwlII Dynamic Human Textured Mesh Sequence Dataset: <https://mpeg-pcc.org/index.php/pcc-content-database/owlII-dynamic-human-textured-mesh-sequence-dataset/>

C.2.10.3 Copyright and license information

The sequences are used as potential test material for MPEG standardization efforts, as well as non-commercial use subject to the accompanying license agreement by the wider research community.

NOTE: The licences for using these datasets in 3GPP need to be checked. The sequences are not used in this study.

C.2.11 Vologram Ltd test sequences

C.2.11.1 Description

Vologram Ltd [45] makes 2 available dynamic human mesh sequences *levi* and *Rafa*.



Figure C.2.11.1-1 levi and Rafa - Vologram Ltd

C.2.11.2 Sequence properties

All sequences can be downloaded as dynamic mesh. The Table 2.11.1 summarizes the properties of the Vologram Ltd sequences.

Table C.2.11.1-1 Vologram Ltd sequences properties dynamic mesh

Test material dataset filename	#Frames	#Vertices	#Faces	Geometry Precision	Texture Coord. Precision	Texture Map Size
levi [45]	150	20k	40k	12 bits	13 bits	4k x 4k
Rafa [45]	150	20k	40k	12 bits	13 bits	4k x 4k

The sequence can be accessed:

- Vologram research dataset: <https://www.volograms.com/research-datasets>

C.2.11.3 Copyright and license information

The sequences are used as potential test material for MPEG standardization efforts, as well as non-commercial use subject to the accompanying license agreement by the wider research community.

NOTE: The licences for using these datasets in 3GPP need to be checked. The sequences are not used in this study.

C.2.12 Exercise test sequences

C.2.12.1 Description

The sequence is captured by a lightweight capture system, equipped with 4 Azure Kinect™ depth cameras and proprietary 3D reconstruction algorithms, is capable of capturing and reconstructing dynamic mesh sequences. It can reconstruct 29.9695 frames per second, with each dynamic human mesh containing more than 50K triangles. The texture maps have resolutions ranging from 1K to 8K (depends on the settings).

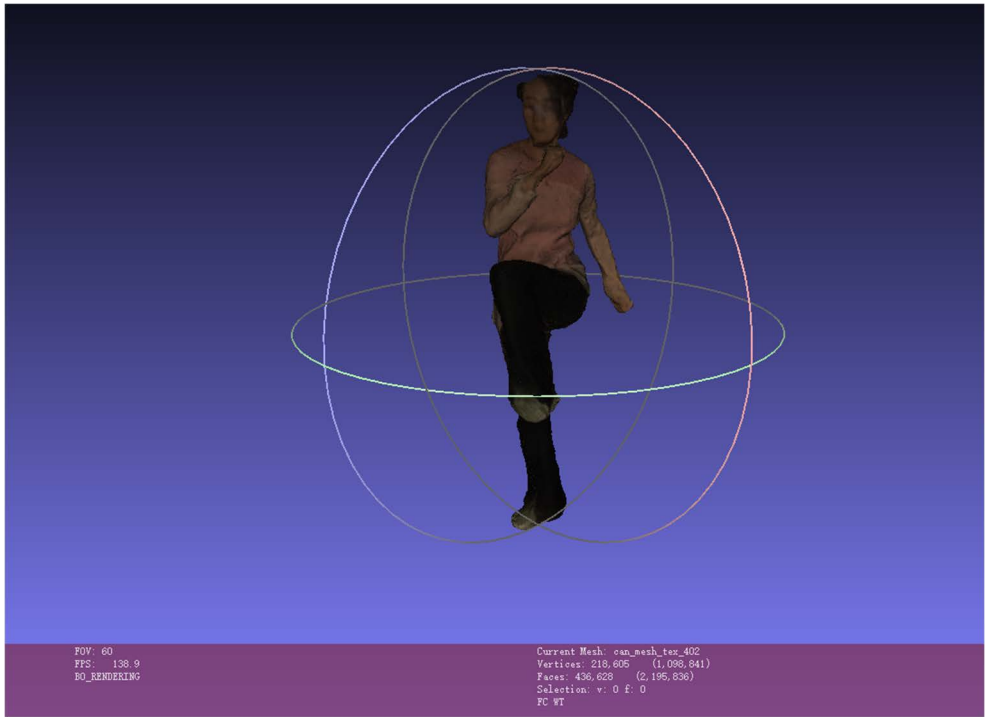


Figure C.2.12.1-1 Exercise’s Sequence

C.2.12.2 Sequence properties

The sequence can be downloaded as dynamic mesh. The dynamic human mesh containing more than 50K triangles. The texture maps have resolution of 8K (depends on the settings).

Table C.2.12.1-1 Exercise sequence properties dynamic mesh

Test material dataset filename	#Frames	#Vertices	#Faces	Geometry Precision	Texture Coord. Precision	Texture Map Size
live_model_guangboticao_8192	200	40k	80k	12 bits	13 bits	8192*8192

The dataset contain one PNG file representing texture, one MTL representing material, and one OBJ file representing geometry for each mesh frame.

The sequence can be accessed:
https://www.dropbox.com/scl/fi/rkbw1sb2i8nbc69po00m/live_model_guangboticao_8192.zip?rlkey=qza774eses1p1jfiybe50c1j2&st=gabk43hw&dl=0

3GPP members can request the password by contacting xujiayi@chinamobile.com.

C.2.12.3 Copyright and license information

Exercise © 2025 by XU is licensed under CC BY-ND 4.0. To view a copy of this license, visit <https://creativecommons.org/licenses/by-nd/4.0/>

C.3 Test Sequences for UE-to-UE Stereoscopic Video Live Streaming

C.3.1 Overview

This annex presents candidate test sequences that are available for testing. Some sequences have been made freely available to 3GPP under license agreement but cannot be made publicly available. Some sequences are free and publicly available for download by respecting the license.

C.3.2 Street View - captured test sequence

C.3.2.1 Description

Real-time street view capture using stereoscopic cameras, combining moving pedestrians, vehicles, and static background elements in a single 3D scene.



Figure C.3.2.1-1 StreetView - captured

C.3.2.2 Sequence properties

The tables C.3.2.2 summarizes the properties of the StreetView captured sequence

Table C.3.2.2-1 StreetView - captured sequence properties

Parameter	Value
Resolution	1920 ×1080 (Per Eye)
Frame Rate	30
Bit Depth	8
Length	344
YUV format	4:2:0
Color Component	ITU-R.BT2020
Color Space	HDR PQ

The sequence can be accessed: <https://pan.baidu.com/s/18ZQXrdm3LTTvV4JDE0sr2A?>

3GPP members can request the password by contacting xujiayi@chinamobile.com.

C.3.2.3 Copyright and license information

sbs_streatView_data © 2025 by Jie Li is licensed under CC BY-ND 4.0. To view a copy of this license, visit <https://creativecommons.org/licenses/by-nd/4.0/>

C.3.3 Cute Dog - Captured test sequence

C.3.3.1 Description

A cute dog plays in the road. The scene is dynamic, with lively movements and detailed fur textures.



Figure C.3.3.1-1 Cute Dog - Captured

C.3.3.2 Sequence properties

The Tables C.3.3.2 summarizes the properties of the Cute Dog - Captured sequence

Table C.3.3.2-1 Cute Dog - Captured sequence properties

Parameter	Value
Resolution	1920 ×1080 (Per Eye)
Frame Rate	30
Bit Depth	8
Length	505
YUV format	4:2:0
Color Component	ITU-R.BT2020
Color Space	HDR PQ

The sequence can be accessed: <https://pan.baidu.com/s/1DgjHpQJ8I-jay75PjCj3NQ?>

3GPP members can request the password by contacting xujiayi@chinamobile.com.

C.3.3.3 Copyright and license information

sbs_Cute_Dog © 2025 by Jie Li is licensed under CC BY-ND 4.0. To view a copy of this license, visit <https://creativecommons.org/licenses/by-nd/4.0/>

C.3.4 Moving Girl - Captured test sequence

C.3.4.1 Description

The girl presents a doll and a signboard to the audience, with particularly strong 3D depth effects visible especially when she extends the objects toward the camera.



Figure C.3.4.1-1 MovingGirl-Captured

C.3.4.2 Sequence properties

The tables C.3.4.2 summarizes the properties of the MovingGirl-Captured sequence

Table C.3.4.2-1 MovingGirl-Capture properties

Parameter	Value
Resolution	1920 ×1080 (Per Eye)
Frame Rate	30
Bit Depth	8
Length	312
YUV format	4:2:0
Color Component	ITU-R.BT2020
Color Space	HDR PQ

The sequence can be accessed: <https://pan.baidu.com/s/1rpdbtxs7TrLGla8sYAvC8w?>

3GPP members can request the password by contacting xujiayi@chinamobile.com.

C.3.4.3 Copyright and license information

TestSequence_JuneXie_2 © 2025-03-26 by June Xie is licensed under CC BY-ND 4.0. To view a copy of this license, visit <https://creativecommons.org/licenses/by-nd/4.0/>

C.3.5 Street View - Generated test sequence

C.3.5.1 Description

The sequence uses the left view of the stereoscopic videos collected in Annex C.3.2 as input, and generate the right view through AI algorithms to synthesize side-by-side stereoscopic videos.



Figure C.3.5.1-1 StreetView - Generated

C.3.5.2 Sequence properties

The StreetView - generated sequence has same properties as defined in C 3.2.2, it can be accessed: <https://pan.baidu.com/s/16tKuVdLWMfyrtn8JbA4QVw>

3GPP members can request the password by contacting xujiayi@chinamobile.com.

C.3.5.3 Copyright and license information

Streetview_lina_generated © 2025 by Lina is licensed under CC BY-ND 4.0. To view a copy of this license, visit <https://creativecommons.org/licenses/by-nd/4.0/>

C.3.6 Cute Dog - Generated test sequence

C.3.6.1 Description

The sequence uses the left view of the stereoscopic videos collected in Annex C.3.3 as input, and generate the right view through AI algorithms to synthesize side-by-side stereoscopic videos.



Figure C.3.6.1-1 Cute Dog - Generated

C.3.6.2 Sequence properties

The Cute Dog - generated sequence has same properties as defined in C 3.3.2, it can be accessed:
<https://pan.baidu.com/s/1TVTtsHriJMeXflrd3CweFQ>

3GPP members can request the password by contacting xujiayi@chinamobile.com.

C.3.6.3 Copyright and license information

Dog_lina_generated © 2025 by Lina is licensed under CC BY-ND 4.0. To view a copy of this license, visit
<https://creativecommons.org/licenses/by-nd/4.0/>

C.3.7 Moving Girl - Generated test sequence

C.3.7.1 Description

The sequence uses the left view of the stereoscopic videos collected in Annex C.3.4 as input, and generate the right view through AI algorithms to synthesize side-by-side stereoscopic videos.



Figure C.3.7.1-1 Moving Girl - Generated

C.3.7.2 Sequence properties

The Moving Girl - generated sequence has same properties as defined in C 3.4.2, it can be accessed:
https://pan.baidu.com/s/1aduJg1C_6j3tq3_-3p7aIw?

3GPP members can request the password by contacting xujiayi@chinamobile.com.

C.3.7.3 Copyright and license information

TestSequence_JuneXie_1 © 2025-03-26 by June Xie is licensed under CC BY-ND 4.0. To view a copy of this license, visit <https://creativecommons.org/licenses/by-nd/4.0/>

C.4 Test Sequences for Streaming of Multi-view plus depth Produced Content

C.4.1 Overview

This annex presents candidate test sequences that are available for testing. The sequences have been made available for 3GPP internal usage under their license agreements.

C.4.2 Breakfast test sequence

C.4.2.1 Description

The Breakfast sequence (Figure C4.2.1-1) is part of the MIV CTC [162] but has not been used for the development of ISO/IEC 23090-12:2023. It consists of 97 frames @ 30 fps, and was captured using a 5 x 3 planar rig of BlackMagic Micro Studio cameras. The total size of the rig is about 1 m wide and 0.5 m tall. The cameras have a field of view of about 66° by 40°. The scene has been shot in the dining room of Chateau de la Ballue, 35560 Bazouges-la-Pérouse, France. While InterDigital originally provided depth maps, later these have been replaced by depth maps that were

produced by ETRI using their internal tools, comprising block matching based on plane-sweeping, cost aggregation, semi-global matching, and depth refinement.



Figure C4.2.1-1: Breakfast sequence (view 7, frame 0)

C.4.2.2 Sequence properties

The properties of Breakfast sequence are summarized in Table C.4.2.2-1.

Table C.4.2.2-1 Breakfast sequence properties

Parameter	Value
Resolution	1920 ×1080
#views	15
#frames	97
Frame Rate	30
Texture Bit Depth	10
Depth Bit Depth	16
YUV format	4:2:0
Texture Color Component	ITU-R BT.709
Depth Color Component	Full range linear

The test sequence and derived work are hosted by InterDigital on their Aspera server (Annex F.2.1.1).

C.4.2.3 Copyright and license information

The license is provided on the Aspera server of InterDigital (Annex F.2.1.1).

C.4.3 Bartender test sequence

C.4.3.1 Description

The Bartender sequence (Figure Cx.3.1-1) was originally provided by ETRI for the 3D Implicit Neural Video Representation (3D-INVR) activity in SC 29/WG 4 [168], and it is currently being used as a reference sequence in the MPEG Gaussian splat coding (GSC) exploration in WG 4/WG 7. The sequence has not been used for the development of ISO/IEC 23090-12 MIV. It consists of 300 frames @ 30 fps, and was captured using a 7 x 3 planar rig of BlackMagic Micro Studio 4K cameras. The total size of the rig is about 3 m wide and 0.5 m tall. The calibration of the camera system was done using Reality Capture, mainly relying on bundle adjustment.

Permission was obtained from ETRI to use this sequence for this study. The license requires citation of [168]. Because the cited document is not publicly available, it was agreed with ETRI to distribute the cited document with the test sequence.

As the sequence was provided without depth maps, Philips has generated depth maps using Immersive Video Depth Estimator (IVDE) 8.0 [169] by Pozań University of Technology. We have used the default parameters of the software, except that we disabled temporal enhancement. Each frame was estimated in parallel and the resulting YUV files were concatenated.

NOTE: Visual inspection of the depth maps shows that the quality of the depth maps is below that of Breakfast, and this has an impact on the evaluation.



Figure C4.3.1-1: Bartender sequence (view 10, frame 0)

C.4.3.2 Sequence properties

The properties of Bartender sequence are summarized in Table C.4.3.2-1.

Table C.4.3.2-1 Bartender sequence properties

Parameter	Value
Resolution	1920 ×1080
#views	21
#frames	300
Frame Rate	30
Texture Bit Depth	10
Depth Bit Depth	16
YUV format	4:2:0
Texture Color Component	ITU-R BT.709
Depth Color Component	Full range linear

ETRI and Philips have agreed that Philips will provide the source material, original MPEG contribution, and derived work on a Philips-managed OneDrive server, and participants of this study may request access based on their e-mail address, by sending an e-mail to the contact person: Bart Kroon <bart.kroon@philips.com>.

C.4.3.3 Copyright and license information

To view a copy of the copyright license, visit the sequence folder on the Philips-managed OneDrive server.

C.4.4 DanceMoves test sequence

C.4.4.1 Description

The DanceMoves sequence (Figure C.4.4.1-1) was captured by Philips on location in Veghel with the help of a professional production company, a self-built capture rig, and volunteer actors. The sequence has not been used for the development of any standard. It consists of 449 frames @ 15 Hz, and was captured using a linear rig of six Azure

Kinect cameras. The total size of the rig is about 60 cm wide. Due to limitations with the capture system (a single laptop with USB 3 interface) the frame rate was limited to 15 fps.

As the depth maps of the Azure Kinect cameras were cropped compared to the texture frames, Philips has generated new depth maps using Immersive Video Depth Estimator (IVDE) 8.0 [169] by Poznań University of Technology. We have used the default parameters of the software, except that we disabled temporal enhancement. Each frame was estimated in parallel and the resulting YUV files were concatenated.

NOTE: Visual inspection of the depth maps shows that the quality of the depth maps is below that of Breakfast, and this has an impact on the evaluation.



Figure C.4.4.1-1: DanceMoves sequence (view 3, frame 0)

C.4.4.2 Sequence properties

The properties of DanceMoves sequence are summarized in Table C.4.4.2-1

Table C.4.4.2-1 DanceMoves sequence properties

Parameter	Value
Resolution	1920 ×1080
#views	6
#frames	449
Frame Rate	315
Texture Bit Depth	10
Depth Bit Depth	16
YUV format	4:2:0
Texture Color Component	ITU-R BT.709
Depth Color Component	Full range linear

Philips will provide the source material and derived work on a Philips-managed OneDrive server, and participants of this study may request access based on their e-mail address, by sending an e-mail to the contact person: Bart Kroon <bart.kroon@philips.com>.

C.4.4.3 Copyright and license information

To view a copy of the copyright license, visit the sequence folder on the Philips-managed OneDrive server.

Annex D: Software Package

D.1 Introduction

This Annex documents software packages used in this TR.

D.2 Video Processing

D.2.1 Overview

The stereoscopic video processing for scenario 1 is supported by a software package provided in Annex D.2.2, D.2.3, and D.2.4.

D.2.2 Common Color Conversion

A Python script for converting YUV to RGB format can be found below:

```
1. import cv2
1. import subprocess
2.
3. def videoInfo(filename):
4.     proc = subprocess.run([
5.         *"ffprobe -v quiet -print_format json -show_format -
        show_streams".split(),
6.         filename
7.     ], capture_output=True)
8.     proc.check_returncode()
9.     return json.loads(proc.stdout)
10.
11. def readVideo(filename):
12.     cmd = ["ffmpeg", "-i", filename]
13.     streams = 0
14.     for stream in videoInfo(filename)["streams"]:
15.         index = stream["index"]
16.         if stream["codec_type"] == "video":
17.             width = stream["width"]
18.             height = stream["height"]
19.             cmd += "-map", f"0:{index}"
20.             streams = streams + 1
21.     cmd += "-f", "rawvideo", "-pix_fmt", "rgb24", "-"
22.     shape = np.array([streams, height, width, 3])
23.     with subprocess.Popen(cmd, stdout=subprocess.PIPE) as proc:
24.         while True:
25.             # One byte per each element
26.             data = proc.stdout.read(shape.prod())
27.             if not data:
28.                 return
29.             yield np.frombuffer(data, dtype=np.uint8).reshape(shape)
```

```

30.
31. if __name__ == '__main__':
32.     import matplotlib.pyplot as plt
33.     from PIL import Image
34.     idx = 1
35.     for left, right in readVideo("./StereoCaptured.mp4"):
36.         # concatenate left and right
37.         img = np.concatenate((left[0:1080,:], right[0:1080,:]), axis=1)
38.         img = Image.fromarray(img)
39.         img.save(f"imgs/{idx}.jpg")
40.         idx += 1

```

D.2.3 Video Composition

A Python script for reading the right/left views and concatenating them into one video frame can be found below:

```

41. import numpy as np
42. import json
43. import subprocess
44.
45. def videoInfo(filename):
46.     proc = subprocess.run([
47.         *"ffprobe -v quiet -print_format json -show_format -
48.         show_streams".split(),
49.         filename
50.     ], capture_output=True)
51.     proc.check_returncode()
52.     return json.loads(proc.stdout)
53.
54. def readVideo(filename):
55.     cmd = ["ffmpeg", "-i", filename]
56.     streams = 0
57.     for stream in videoInfo(filename)["streams"]:
58.         index = stream["index"]
59.         if stream["codec_type"] == "video":
60.             width = stream["width"]
61.             height = stream["height"]
62.             cmd += "-map", f"0:{index}"
63.             streams = streams + 1
64.     cmd += "-f", "rawvideo", "-pix_fmt", "rgb24", "-"
65.     shape = np.array([streams, height, width, 3])
66.     with subprocess.Popen(cmd, stdout=subprocess.PIPE) as proc:
67.         while True:
68.             # One byte per each element
69.             data = proc.stdout.read(shape.prod())
70.             if not data:
71.                 return
72.             yield np.frombuffer(data, dtype=np.uint8).reshape(shape)
73.
74. if __name__ == '__main__':
75.     import matplotlib.pyplot as plt
76.     from PIL import Image
77.     idx = 1

```

```
77.     for left, right in readVideo("./StereoCaptured.mp4"):
78.         # concatenate left and right
79.         img = np.concatenate((left[0:1080,:], right[0:1080,:]), axis=1)
80.         img = Image.fromarray(img)
81.         img.save(f"imgs/{idx}.jpg")
82.         idx += 1
```

D.2.4 FFmpeg Tools

FFmpeg is a universal media converter. It can read a wide variety of inputs - including live grabbing/recording devices - filter, and transcode them into a plethora of output formats.

- An FFmpeg 7.0 command for storage each frame into .yuv:

```
1. ffmpeg -start_number 1 -r 30 -i %d.jpg -pix_fmt yuv420p -s 1920x1200 out.yuv
```

D.3 Scenario 2 Processing

D.3.1 Overview

The generation of objective metrics and 2D videos for subjective viewing for scenario 2 is supported by a software package provided in the repository: <https://github.com/5G-MAG/rt-beyond2d-evaluation-framework> in the folder "point_cloud".

The software package permits the following functionalities:

- Test sequence preparation
- Bitstream generation and objective metric generation
- 2D video generation using a camera path for subjective viewing

D.3.2 Installation

D.3.2.1 Cloning

```
git clone https://github.com/5G-MAG/rt-beyond2d-evaluation-framework
```

```
cd rt-beyond2D-evaluation-framework/point_cloud
```

Please use a [python virtual environment](#) to install dependencies and run the scripts. A requirements.txt file is provided such that a suitable virtual environment can be set-up as follows:

```
python3 -m venv venv
venv\Scripts\activate # on Windows
. venv/bin/activate   # on Linux
python -m pip install -upgrade pip
pip install -r requirements.txt
```

D.3.2.2 Working Directory

The scripts assume that the current directory is a local working directory, at the root of the repository.

D.3.3 Test sequence preparation

D.3.3.1 Dense dynamic point cloud

This clause describes how reference sequences provided in dynamic mesh format are converted to the dense dynamic point cloud format with the target quality (vox11, approximately 2M points/frame). Please follow instructions in annex C.2 for downloading the sequences Mitch, Juggle Soccer, Nathalie, Aliyah and Henry in dense dynamic mesh format. The sequences Aliyah and Henry are provided as Blender project and generation of dense dynamic mesh is described in the complementary document doc/readme_ply_generation.md in the repository.

D.3.3.1.1 Generation of target dense dynamic point clouds

To proceed with the generation, the user needs to navigate to the /ply_generation/ directory, which contains:

- *.py: Python scripts for generating PLY (point cloud) files.
- output_info/: Directory containing all expected md5sum result files for meshes (*_mesh_md5.txt) and PLY files (*_output.log) for each sequence.
- jsons/: Directory with an example of input configuration files.-

A JSON file named 3gpp_selection.json is provided as input and is located in the jsons/ directory. It contains all information listed in Table D.3.3.1.1-1 This JSON file needs to be updated for each sequence with the correct paths to the meshes for your environment (MeshObjPath and MeshTxtPath).

Table D.3.3.1.1-1 conversion parameters

Sequence	Geo Quantization Bitdepth	Ratio	1 st Frame Index	Frame Number
Mitch	11	0.70	1	475
JuggleSoccer	11	1	0	125
Henry	11	0.75	1	733
Nathalie	11	1	1	925
Aliyah	11	0.88	1	1112

Once the JSON file is updated with the correct mesh paths, the PLY generation can be launched using the script exec_ply_generation.py which goes through the following steps:

- The MPEG mmtric software [53] is automatically downloaded to the output directory within the dependencies directory.
- A sampling pass gathers information on the sequence for quantifying the number of expected points. A ratio is provided via the JSON file to ensure each sequence generates point clouds with approximately 2M points/frame.
- Quantization pass.
- Cleaning pass: This step removes all duplicate points using PyntCloud in Python.

The script is launched from the python environment with the following command:

```
python3 ply_generation/exec_ply_generation.py -i
ply_generation/jsons/3gpp_selection.json -o $YOUR_OUTPUT_PATH
```

For help on the script see complementary document doc/readme_ply_generation.docx in the doc folder installed by Git.

In the output directory, you will find the generated PLY files and corresponding log files for each sequence.

To ensure the PLY generation proceeded as expected, md5 checksums for meshes, the number of points and the md5 checksums for point clouds are provided for each frame of each sequence. These details are compiled into a single file per sequence and stored in `ply_generation/output_info`.

D.3.4 Bitstream and objective metric generation

D.3.4.1 Dense dynamic point cloud

This clause assumes that all test sequences are available in the dense point cloud representation format as described in clause D.3.3.1. This clause describes how to execute the test environment using the provided scripts. Deeper information on the functioning of the scripts is given the documentation installed via Git. Interested users are referred to the document `doc/readme_ply_to_bin.md` in the repository.

D.3.4.1.1 Executing tests

Python scripts are provided to:

- Build the test environment under the output “dependencies” directory. The MPEG V-PCC test model [147] will be used to encode and decode test sequences. The MPEG mmetric software [53] will be used to compute metrics. These tools are automatically downloaded and built by the script.
- Perform tests, including:
 - Encode each sequence for each condition, rate point and profile.
 - Decode the corresponding sequence.
 - Compute the objective metrics.
 - Generate CSV tables and graph worksheets.

To execute the tests, the user should navigate to the “`ply_to_bin/`” directory, which contains:

- `*.py`: Python scripts to encode, decode, compute metrics and generate CSV and XLSM workbooks.
- `templates/`: Directory with template XLSM sheet used for graph generation.
- `jsons/`: Directory with configurations
 - `sequences.json`: Describes the list of input sequences to test. It contains information on the location of point cloud sequences and has to be set by the user to point to the right location. It also has information on the name of the configuration file used for the encoding step (`${test_sequence}.cfg`).
 - `3gpp_test_configuration.json`: Describes the test lists to perform. For each profile, it defines the encoding parameters (“`--profileToolsetIdc`”, “`--profileReconstructionIdc`”, “`--mapCountMinus1`”), the number of frames to test (typically 300) and the list of sequences to be tested. This list includes:
 - The “`id`” corresponding to the one set in the `sequences.json` file.
 - The condition to test, here, random access.
 - A list of 5 rate points as defined in Table D.3.4.1.1-1.

Table D.3.4.1.1-1 with QP selection for obtaining the fixed target bitrates

Rate	Target Bitrate (mbps)	S01 Mitch			S02 Juggle Soccer			S03 Henry			S04 Nathalie			S05 Aliyah		
		QP Geo	QP Att	Occ Prec	QP Geo	QP Att	Occ Prec	QP Geo	QP Att	Occ Prec	QP Geo	QP Att	Occ Prec	QP Geo	QP Att	Occ Prec

R01	5	29	33	4	30	39	4	23	34	4	25	39	4	28	39	4
R02	10	23	29	2	19	35	2	15	30	2	24	30	4	20	32	4
R03	20	19	25	2	11	28	2	8	26	2	20	26	4	20	26	4
R04	30	15	23	2	9	24	2	7	23	2	18	24	2	18	24	2
R05	50	11	21	2	5	21	2	6	20	2	17	21	2	7	23	2

A script “exec_binGenerator.py” is provided to automate all steps including encoding, decoding, objective metrics computation and output generation. It can be launched from your Python environment with the following command:

```
python exec_binGenerator.py -o $YOUR_OUTPUT_DIR -i jsons/sequences.json -t
jsons/test_configuration.json
```

For help on the script see the complementary document readme_ply_to_bin in the doc folder installed by Git.

The output directory structure is:

- cmd: Directory with job command and logs.
- dependencies: Compilation of TMC2 and mmetric software used to perform the test.
- A list “Fyy_ProfileName” directories with Fyy corresponds to the number of tested frames, ProfileName corresponds to the tested profile and includes generated bitstreams.
- A list of CSV files with extracted metric information per profile for a given number of frames.
- Excel worksheets with graphs per profile for a given number of frames.

D.3.4.1.2 Objective results

CSV and workbook files are automatically generated by the scripts. The output log containing all metrics information is used to extract metrics and a build CSV files. Each CSV file concatenates metrics information for each condition and selected profile and is generated for all sequences and rate points.

The following information is stored in a CSV file:

- SeqId: identifier of the sequence
- CondId: tested condition (RA)
- RateId: tested rate number [R1..R5]
- nbFrame: number of tested frames
- NbInputPoints: number of points in the source sequence
- NbOutputPoints: number of points in the candidate test sequence
- MeanOutputPoints: mean number of points in the candidate test sequence
- MeanDuplicatePoints: mean number of duplicated points (with same geometry) in the candidate test sequence
- TotalBitstreamBits: size of the bistream in bits
- geometryBits: size of the geometry stream in bits
- metadataBits: size of the metadata stream in bits
- attributeBits: size of the attribute stream in bits
- D1Mean: mseF,PSNR (p2point)
- D2Mean: mseF,PSNR (p2plane)

- LumaMean: c[0],PSNRF
- CbMean: c[1],PSNRF
- CrMean: c[2],PSNRF
- PCQM: PCQM PSNR
- SelfEncoderRuntime: encoder time for current process
- ChildEncoderRuntime: encoder time for child processes
- SelfDecoderRuntime: decoder time for current process
- ChildDecoderRuntime: decoder time for child processes

From the CSV file, an excel spreadsheet is generated from the template xlsx sheet (in the “templates” directory) to get tables and graphs for interpretation of the results.

D.3.5 Video generation

D.3.5.1 Dense dynamic point cloud

This clause describes how to generate 2D videos with a predefined camera path. It is assumed that test sequences are available either in raw dense point cloud format or as bitstream encoded with V-PCC. Please check clauses D.3.3 and D.3.4 on how to generate these inputs.

The provided scripts use the MPEG V-PCC test model [147] for decoding V-PCC bitstreams and the MPEG Representative Renderer [140] to generate videos from PLY files. Both are automatically cloned and built when running the scripts for the first time.

To proceed with the video generation, the user needs to navigate to the /bin_to_video/ directory, which contains:

- *.py: Python scripts for generating PLY (point cloud) files.
- jsons/: Directory with an example of input configuration files.
- Multiple JSON files are available in the jsons/ directory:
- 3gpp_selection_src.json provides the information for the sources dense point clouds sequences. This JSON file needs to be updated for each sequence with the correct paths to the source .PLY files for your environment (PathDec parameter).
- 3gpp_selection_dec.json provides the information for the encoded dense point cloud sequences. This JSON file needs to be updated for each sequence with the correct paths to the V-PCC encoded .BIN files for your environment (PathEnc parameter).
- 3gpp_test_configuration.json contains an example configuration to generate multiple videos with different MPEG Representative Renderer [140] settings for each sequence. The provided rendering settings are described in the Table D3

Table D.3.5.1-1 with rendering settings for video generation

Rendering Job Name	Point Primitive	Renderer Arguments	Background
Cube size 1	Cube	--floor=1 --type=0 --size=1	No
Blend size 2.4 alpha 1.8 linear	Linear blended splat	--floor=1 --type=3 --alphaFalloff=1.8 --size=2.4 -- blendMode=1	No
Bck blend size 2.4 alpha 1.8 linear	Linear blended splat	--type=3 --alphaFalloff=1.8 --size=2.4 -- blendMode=1	Yes

The JSON directory also contains the /camerapath/ and /background/ folders, providing additional configuration files used by the MPEG Representative Renderer [140].

- /camerapath/ contains files describing pre-recorded camera trajectories for each content
- /background/ contains files describing the position, orientation and scale of external 3D assets used as background for each content. These files need to be updated for each sequence with the correct path to the assets.

To generate the video of the sources:

```
python3 bin_to_video/exec_binToVideo.py \  
-c bin_to_video/jsons/3gpp_test_configuration.json \  
-i bin_to_video/jsons/3gpp_selection_src.json \  
-o $YOUR_OUTPUT_DIR -v
```

To generate the video of the encoded content:

```
python3 bin_to_video/exec_binToVideo.py \  
-c bin_to_video/jsons/3gpp_test_configuration.json \  
-i bin_to_video/jsons/3gpp_selection_dec.json \  
-o $YOUR_OUTPUT_DIR
```

The scripts generate uncompressed .RGB videos. For delivery purposes, the videos were compressed with an external tool to lossless HEVC. No such feature is delivered with this package.

Detailed information on the functioning of the scripts is given in the document doc/readme_bin_to_video.md in the repository.

Annex E: Testing support material

E.1 3D background models for scenario 2 tests

E.1.1 Introduction

Volumetric Video test sequences presented in annex C.2 contain single assets without background. For subjective tests of scenario 2 representation formats, videos using a camera path are generated on top of a neutral background and of a 3D background model. The purpose of the 3D background model is to enable the evaluation of interference between a test sequence and a background. For example, artefacts of the borderline of a test sequence may be less visible in front of a background model than on a neutral background, see clause 7.3.4.3 where the impact of a background is explained and illustrated. The focus of the subjective evaluation should be on the test sequence itself and not the background, the background is only a support.

This chapter presents three 3D background models that are free of charge and publicly available under license.

E.2 Crouch End Station 3D background model

E.2.1 Description

This model presents the remains of the Crouch End Station in London.



Figure E.2.1-1 Remains Crouch End Station – content courtesy artfletch / Sketchfab

E.2.2 Copyright and license information

The Remains Crouch End Station 3D background model is available under the Creative Commons license:
<https://creativecommons.org/licenses/by/4.0/>

This model is free of charge and publicly available.

E.3 Great Drawing Room 3D background model

E.3.1 Description

This model represents the Great Drawing Room of the Hallwyl Museum in Stockholm/Schweden. The room is inspired by the baroque style of the 17th century.



Figure E.3.1-1 Great Drawing Room – content courtesy The Hallwyl Museum / Sketchfab

This background model can be publicly accessed: <https://sketchfab.com/3d-models/the-great-drawing-room-feb9ad17e042418c8e759b81e3b2e5d7>

E.3.2 Copyright and license information

The Great Drawing Room 3D background model is available under the Creative Commons license: <https://creativecommons.org/licenses/by/4.0/>

This model is free of charge and publicly available.

E.4 Southbank Undercroft Skatepark 3D background model

E.4.1 Description

This model represents a skate park in the Undercroft beneath Queen Elizabeth Hall on the Southbank, London.



Figure E.4.1-1 Southbank Undercroft Skatepark – content courtesy artfletch / Sketchfab

The background model can be publicly accessed: <https://sketchfab.com/3d-models/southbank-undercroft-skatepark-add37d5dc6a2456eb0d7607c6fbd884d>

E.4.2 Copyright and license information

The Southbank Undercroft Skatepark background model is available under the Creative Commons license: <https://creativecommons.org/licenses/by/4.0/>

This model is free of charge and publicly available.

Annex F: Data Management and Hosting

F.1 Reference Sequences

F.1.1 Hosting

Data for reference sequences are hosted at the public server:

<https://dash-large-files.akamaized.net/WAVE/3GPP/Beyond2D/ReferenceSequences/>

Below this folder, for each reference sequence a dedicated folder with the name of the sequence is created. The folder includes at least the following information:

- JSON file for raw content according to schema in clause B.2, depending on the representation format.

In addition, the folder may include the following information:

- the raw sequence, if hosted on the public server,
- a preview file,
- a thumbnail image.

For some reference sequences the license may not allow storage on the indicated public server and in this case additional servers including information how to obtain the credentials are listed below.

F.1.1.1 Scenario 2

Freely available reference sequences to 3GPP members coming from the content providers Volucap and XD Productions are hosted at:

<https://aspera.pub/I4tSQ8k>

3GPP members can request credentials by sending a request per email to: 3GPP_B2D_Datasets@interdigital.com

The folder hierarchy follows the same structure as on the public server and information provided here is complementary to the public server.

F.1.1.2 Scenario 3

The reference sequences are freely available to 3GPP members:

The Breakfast sequence is hosted by InterDigital at:

<https://aspera.pub/I4tSQ8k>

3GPP members can request credentials by sending a request per email to: 3GPP_B2D_Datasets@interdigital.com

The Bartender and DanceMoves sequences are hosted by Philips. 3GPP members can request access by sending a request per email to: bart DOT kroon AT philips DOT com.

The folder hierarchy follows the same structure as on the public server and information provided on both servers is complementary to the public server.

F.1.2 Uploading

For uploading a new reference sequence to the public server, please create a new issue here https://github.com/haudiobe/Beyond2D-Content/issues/new?assignees=&labels=request&template=reference_sequence.md&title=

Please specify all information that needs to be added to the JSON file for raw video sequences as defined in clause B.2.

F.1.3 Downloading

Sequences can be downloaded from the above servers and folders. Licensing terms must be obeyed. Downloaded files should be MD5 verified.

On Windows

Open Command Prompt

Open your downloads folder by typing `cd Downloads . . .`

Type `certutil -hashfile` followed by the file name and then MD5

Check that the value returned matches the value the MD5 file you downloaded from the json

On Mac

Open Terminal

Type `md5` and hit the SPACE button

Drag the file you have downloaded into the Terminal Window. ...

Hit Enter

You should now see the MD5 Checksum so you can compare it to the string you have been given in json

On Linux

Open Terminal

Type `md5` and add file name – hit enter

Verify the json provided MD5

F.2 Anchors and Tests

F.2.1 Hosting

Test results are hosted at the public server:

<https://dash-large-files.akamaized.net/WAVE/3GPP/Beyond2D/Bitstreams/>.

Below this folder, a hierarchy is created:

- Scenario
- Codec
- Metrics (contains all metrics)
- Videos (contains all videos for subjective viewing)
- Characterization (contains all characterization results)

- CFG (contains all cfgs)
- streams.csv (Documents the streams and generation)
- videos.csv (Documents the videos)
- crosscheck.csv (documents all cross-check results)
- verification.csv (documents all verification results)

The folder includes at least the following information:

- JSON file for binary video according to schema in clause B.2.2.
- Potentially log files

A basis example hierarchy is as follows:

```
/Bitstreams
  /Scenario-2
    /V-PCC
      /Metrics
      /Videos
      /Aliya
```

For some reference sequences the license may not allow storage of information on the indicated public server and in this case additional servers including information how to obtain the credentials are listed below.

F.2.1.1 Scenario 2

Results for freely available reference sequences to 3GPP members coming from the content providers Volucap and XD Productions are hosted at:

<https://aspera.pub/I4tSQ8k>

3GPP members can request credentials by sending a request per email to: 3GPP_B2D_Datasets@interdigital.com

The folder hierarchy follows the same structure as on the public server and information provided here is complementary to the public server.

A basis example hierarchy is as follows:

```
/Bitstreams
  /Scenario-2
    /V-PCC
      /Mitch
      /Soccer
      /Nathalie
```

F.2.1.2 Scenario 3

Results for the reference sequences are freely available to 3GPP members:

Results for the Breakfast sequence are hosted by InterDigital at:

<https://aspera.pub/I4tSQ8k>

3GPP members can request credentials by sending a request per email to: 3GPP_B2D_Datasets@interdigital.com

Results for the Bartender and DanceMoves sequences are hosted by Philips. 3GPP members can request access by sending a request per email to: bart DOT kroon AT philips DOT com.

The folder hierarchy follows the same structure as on the public server and information provided on both servers is complementary to the public server.

F.2.2 Uploading

For uploading data on the public server, please create a new issue here

- [https://github.com/haudiobe/Beyond2D-](https://github.com/haudiobe/Beyond2D-Content/issues/new?assignees=&labels=request&template=anchor_tuple.md&title=)

[Content/issues/new?assignees=&labels=request&template=anchor_tuple.md&title=](https://github.com/haudiobe/Beyond2D-Content/issues/new?assignees=&labels=request&template=anchor_tuple.md&title=)

Please specify all required information that needs to be added to the JSON file for encoded video sequences as defined in clause B.3.

F.2.3 Downloading

Data can be downloaded for free from the above folders. Licensing terms must be obeyed. Downloaded files should be MD5 verified.

Annex G:

Change history

Change history							
Date	Meeting	TDoc	CR	Rev	Cat	Subject/Comment	New version
2024-04	SA4#127-bis	S4-240825				Initial Version	0.0.1
2024-05	SA4#128	S4-240947				Updated version based on SA4-post 127-bis, 24,May,2024	0.0.2
2024-05	SA4#128	S4-241319				Update style and include agreed content in S4-241266, S4-241336 and S4-241318	0.0.3
2024-08	SA4#129-e	S4-241491				Updated version based on agreed Tdoc S4aV240023, S4aV240040 In SA4-post 128 meeting.	0.0.4
2024-08	SA4#129-e	S4-241721				Updated version based on agreed Tdoc S4-241708, S4-241709 and S4-241710 during SA4#129-e meeting.	0.1.0
2024-11	SA4#130	S4-241867				Updated version based on agreed Tdoc S4aV240062 during SA4-post 129-e meeting.	0.1.1
2024-11	SA4#130	S4-242196				Updated version based on agreed Tdoc S4-241997, S4-242000, S4-241871, S4-242098, S4-242199, S4-242226, S4-242227, S4-242193 during SA4 130 meeting.	0.2.0
2025-02	SA4#131	S4-250076				Updated version based on 1). the agreed Tdoc S4aV250012 from the SA4 post-130 meeting. 2). Editorial updates: Duplicate references within the Draft TR have been resolved, and typos have been corrected.	0.2.1
2025-02	SA4#131	S4-250372				Updated version based on agreed Tdoc S4-250379, S4-250378, S4-250365, S4-250345, S4-250344, S4-250273, S4-250272, S4-250070, S4-250071, S4-250072, S4-250073	0.3.0
2025-04	SA4#131-bis-e	S4-250680				Updated version based on agreed Tdoc S4-250727, S4-250729, S4-250731, S4-250732, S4-250717, S4-250718, S4-250677, S4-250678	0.4.0
2025-05	SA4#132	S4-251053				Updated version based on 1) agreed Tdoc S4-251048, S4-251049, S4-251055, S4-251056, S4-250939, S4-251131 2)Editorial updates: Add references. Remove uncopyrighted images and replace them with properly licensed alternatives.	0.5.0
2025-06	SA#108	SP-250638				Agreed version sent to SA for information.	1.0.0
2025-07	SA4#133-e	S4-251593				Updated version based on 1) agreed Tdoc S4aV250055, S4-251263, S4-251495, S4-251496, S4-251289, S4-251476, S4-251320, S4-251532, S4-251524; 2) Editorial updates. 3) Solve ENs.	1.3.0
2025-09	SA#109	SP-251492				Version 2.0.0 created by MCC to be sent to TSG SA for approval	2.0.0
2025-09	SA#109					Version 19.0.0 created by MCC for publishing upon approval in TSG SA	19.0.0

History

Document history		
V19.0.0	October 2025	Publication