**INTERNATIONAL ORGANISATION FOR STANDARDISATION**

**ORGANISATION INTERNATIONALE DE NORMALISATION**

**ISO/IEC JTC1/SC29/WG11**

**CODING OF MOVING PICTURES AND AUDIO**

**ISO/IEC JTC1/SC29/WG11 MPEG2017/m41627**

**October 2017, Macao, China**

|  |  |
| --- | --- |
| **Source** | **Fraunhofer HHI** |
| **Status** | **Input contribution** |
| **Title** | **[ISOBMFF] Extractors reference software** |
| **Author** | Dimitri Podborski, Yago Sanchez, Robert Skupin, Cornelius Hellge, Thomas Schierl |

# Introduction

This contribution describes the software additions to the file format reference software for enabling the usage of Extractors according to Section A.7 of [2]. Details regarding the implementation are provided in Section 2.

# Contribution of Extractors to ISOBMFFLib

## Overview

The software for the HEVC Extractors (as defined in Annex A.7 of [2]) builds on top of the C library ‘libisomediafile’ taken from the following MPEG SVN repository:

<http://wg11.sc29.org/svn/repos/MPEG-04/Part12-ISO_Base_Media_File_Format/trunk/>IsoLib/libisomediafile

The software provides an application which allows to resolve NAL units of type 49 (HEVCExtractor) in any track with sample entry type ‘hvc2’. It iterates over all constructors in each Extractor NAL unit, determines the constructor type and, depending on its type (inline or sample-constructor) copies the referenced data. In case of an inline constructor the data is taken directly from the current sample and in case of a sample-constructor the referenced sample is taken from the dependent ‘hvc1’ track while the required portion of the data is signalized by data\_offset and data\_length entries in the sample constructor.

The software allows to aggregate a single HEVC compliant bitstream from multiple ‘hvc1’ tracks by playing a single ‘hvc2’ track which contains extractors. The test vectors provided with the contribution [1] can be used together with the software. In addition, the class HEVCExtractorReader (described in Section 2.4) can be compiled as a library and used in another application as shown in the Android extractor player in [3].

## Changes overview

The following list contains all changes:

|  |  |  |
| --- | --- | --- |
| **Changed file** | **Status** | **Reason** |
| ISOSampleDescriptions.c | Modified | Added support for ‘resv’ sample entry for retrieval of: parameter sets, lengthSizeMinusOne flag, and originalFormat |
| MP4Atoms.h | Modified | Definitions for new atoms and functions. |
| MP4Atoms.c | Modified | Added new atom to MP4CreateAtom (MP4CreateHEVCConfigAtom) |

Table 1 - Libisomediafile updates

The software can be found here:

URL: <https://datacloud.hhi.fraunhofer.de/nextcloud/index.php/s/FkKMkj8NvfB7GYf>

Password: LKS9F6ykwBFf

In the project root directory an additional ‘hevc\_extractors’ folder with the following structure is created:

.

├── TestData

│   └── DownloadCase.mp4

├── linux

│   └── Makefile

├── macosx

│   ├── CMakeLists.txt

│   └── build.sh

└── src

├── HEVCExtractor.cpp

├── HEVCExtractor.h

└── hevc\_extractors.cpp

It includes four directories: TestData, linux, macosx and src. Testdata contains an OMAF test vector from [1] for download case. Linux and macosx directories contain makefiles in order to build the project (on macosx cmake is used). To compile the source code on macosx go to macosx folder and execute ./build.sh. This will compile ‘libisomediafile’ library first and link it to the application. On linux you have to compile ‘libisomediafile’ manually before you run make inside the linux directory.

## libisomediafile library updates

These following functions have been added to libisomediafile:

|  |  |
| --- | --- |
| **Function** | **Description** |
| ISOGetRESVSampleDescriptionPS | Returns parameter sets from ‘resv’ sample entry |
| ISOGetRESVLengthSizeMinusOne | Obtains lengthSizeMinusOne flag from ‘resv’ sample entry |
| ISOGetRESVOriginalFormat | Obtains the original format string from ‘resv’ sample entry |

Table 2 – New functions of libisomediafile

These following data types have been added to libisomediafile:

|  |  |
| --- | --- |
| **Data type** | **Description** |
| MP4RestrictedInfoAtom | Restricted info atom struct |
| MP4RestrictedVideoAtomType | ‘resv’ FourCC |
| MP4RestrictedInfoAtomType | ‘rinf’ FourCC |
| MP4SchemeTypeAtomType | ‘schm’ FourCC |
|  |  |

Table 3 – New data types of libisomediafile

Add new atom creation routine in MP4CreateAtom for ISOHEVCConfigAtomType (‘hvcC’)

## HEVCExtractorReader class

This class is written in C++ and uses available, as well as new, interfaces of libisomediafile to retrieve samples from an OMAF compliant ISOBMFF file which contains extractor track(s) ‘hvc2’ as well as ‘hvc1’ video tracks.

When initializing an object of HEVCExtractorReader class, first we open the provided file using ISOOpenMovieFile and iterate over all tracks while assigning an ISOTrackReader for each of the tracks. Then we parse original format of the track to determine if the track is ‘hvc1’ or ‘hvc2’. If the track is ‘hvc2’ we also store trackIDs of all referenced ‘hvc1’ tracks which are used later when extractors are resolved.

Then we select a track and obtain samples by accessing the bytes indicated in the 'trun' box within the fragments using the ISOTrackReader of the corresponding track. When the sample of ‘hvc2’ track contains extractors (i.e. NAL unit that have nal\_unit\_type equal to 49), the body of the extractor is parsed and, depending on the constructor type, the body is replaced with the corresponding data. The data that is replaced can be either data encapsulated within the extractor (inline constructor), or data from other tracks (sample constructor). When an inline constructor is found, its length field indicates how many bytes by the following data field should be copied to the body of the extracted sample. When a sample constructor is found, the ‘trun’ box of the track with trackID corresponding to the track reference index is parsed and the sample with same decoding time is obtained, then the bytes from the sample following data\_offset up to data\_offset + data\_length (clipping if bigger than the referenced sample size) are extracted to the extracted data.

## Extractor application

The compiled application ‘extractor\_player’ is a simple command line application which uses the HEVCExtractorReader class, to extract the HEVC bitstream from a selected ‘hvc2’ or ‘hvc1’ track. The following command line parameters are defined:

|  |  |
| --- | --- |
| **Parameter** | **Description** |
| -h | Prints a description of the parameters |
| -i InputFile | Inputfile (mandatory) |
| -o OutputFile | Outputfile: out.265 (Default) |
| -l | List all trackIDs |
| -t trackID | TrackID to play |
|  |  |

Table 4 – Command line parameters for extractor\_player

**Usage examples:**

Lists all trackIDs and their types ‘hvc1’ or ‘hvc2’:

./extractor\_player –i DownloadCase.mp4 -l

Extracts the bitstream of the first viewport of [1] from ‘hvc2’ track with trackID 13:

./extractor\_player –i DownloadCase.mp4 -t 13 –o viewport1.265

Extracts the bitstream of the first low resolution tile of [1] from ‘hvc1’ track with trackID 1:

./extractor\_player –i DownloadCase.mp4 -t 1 –o tile1\_lowRes.265

# Conclusion

It is suggested to include the provided software to the MPEG reference software repository and to discuss in which form it should be included.

# References

1. [Dimitri Podborski](mailto:dimitri.podborski@hhi.fraunhofer.de), et al., “Text and test vectors for a viewport-dependent profile with tile streaming”, [m40803](http://phenix.int-evry.fr/mpeg/doc_end_user/current_document.php?id=58461&id_meeting=171), 119th MPEG meeting in Torino.
2. ISO/IEC 14496-15 Carriage of network abstraction layer (NAL) unit structured video in the ISO base media file format
3. Dimitri Podborski, et al., “Android ISOBMFF extractor demonstrator”, m41628, 120th MPEG meeting in Macao