|  |  |
| --- | --- |
| **Joint Collaborative Team on Video Coding (JCT-VC)**  **of ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29/WG 11**  24th Meeting: Geneva, CH, 26 May – 1 June 2016 | Document: JCTVC-X0053 |

|  |  |  |  |
| --- | --- | --- | --- |
| *Title:* | HDRTools: Software status | | |
| *Status:* | Input Document to JCT-VC | | |
| *Purpose:* | Information | | |
| *Author(s) or Contact(s):* | Alexis Michael Tourapis,  David Singer  Apple, Inc. 1 Infinite Loop Cupertino, CA 95014 USA | Tel: Email: | +1-408-228-7983 atourapis@apple.com |
| *Source:* | Apple, Inc. | | |

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

# Abstract

This report summarizes the activities relating to the HDRTools software package development that have taken place between the 23rd and 24th JCT-VC meetings. Activities focused on integration of software adoptions, software maintenance, and various other enhancements.

# Introduction

The HDRTools software package [1] was created for providing the necessary signal processing and conversion tools to the video standardization community, in order to study the processes involved for the ingestion and processing of High Dynamic Range (HDR) and Wide Colour Gamut (WCG) video data. The package basically includes a variety of tools for converting video data from one format to another, including color space conversions, applications of transfer functions, chroma format conversions, scaling, primitive tone mapping, and filtering among others. It also includes some basic video analysis tools, including tools for video quality measurement. The package is currently not only limited to HDR/WCG video data and could be used for other applications if desired.

The tool was previously only available to MPEG and VCEG members and access was protected by a password. However, since April of 2016, read access to this tool was made public and is currently accessible without a password from <https://gitlab.com/standards/HDRTools/> [1]. Software maintenance and control is still managed by the original developers of this software. There is currently no intention in making this an open-development software. Any suggestions, new modules, or bug fixes should be provided directly to the original developers and integration in the software will remain at their discretion.

# History of Revisions

At the 23rd JCTVC meeting a few HDRTools related contributions were made, including two new distortion metrics [5][6], new methods for luma adjustment [7][8][9], and adaptive downsampling and upsampling methods [10] . All of these contributions were adopted into the software, some after considerable code revision and rewriting, and are now available in the newly released version of the software (version 0.11 [2]). This version was released on May 20th 2016, although this code was already available through the software’s development branch.

Additional extensions were made to the software given considerable interest in evaluating new functionality, such as the ICtCp [11][12] colour representation, improved support for AVI and initial support for DPX file formats, and use of look up tables for transfer function conversions among other things. Some of the code was also considerably restructured and memory handling vastly improved.

In more detail, the changes in HDRTools in version 0.11 since version 0.10 include the following:

- Additional luma adjustment methods - JCTVC-W0052 (AT)

- Addition of Canon's JCTVC-W0056 luma adjustment method (VK, AT)

- Addition of Netflix's JCTVC-W0107 luma adjustment method (AN, AT)

- LUT support for Transfer Functions (both for conversions and metrics).

Feature not properly supported for HLG (AT)

- Support of luma adjustment for HLG (AT)

- Implementation of Adaptive downsampling - JCTVC-W0051 (AT)

- Implementation of Adaptive upsampling - JCTVC-W0051 (AT)

- VIF support (HRZ, MA, MTP, PN, AT)

- VQM support (SS, KP, AT)

- limited DPX file input support (AT)

- extended AVI support (AT)

- Support of HLG (MN, MP, AT)

- ICtCp support (AT)

- Various bug fixes, optimizations, and code improvements (AT)

- Memory restructuring (AT)

# Further Work

Users of HDRTools are strongly encouraged to file bug reports in the available Issue Reporting system [4]. Currently there are no bug reports filed relating to the software. However, there are several known limitations of the HDRTools software that depending on time and resources may be addressed in the future. Some of the limitations include:

* Format limitations in most image/video file formats including the following:
  + Only floating point raw (uncompressed) data are supported for OpenEXR.
  + Limited TIFF file format support
  + Only ST 2084 (PQ) encoded files supported in DPX. No DPX output supported.
  + No AVI input support
  + No YUV4MPEG output support
  + No floating point data support in RAW (headerless) files
* I/O could be considerably enhanced.
* Limited Image Scaling support.
* No temporal buffers for temporal processing.
* Most variables are public instead of private. This should be corrected in the future, with variables changed into private, and appropriate Set/Get functions created. This would improve encapsulation among others.
* Many processes are inefficient and slow. Code could be improved.
* Not all fixed precision processing is properly implemented. Priority has been placed in floating/double precision processing. All processes should be carefully tested and validatedusing different types of data.
* Distortion computation is currently limited to input sources that are of the same type.
* Functionalities such as data blending, concatenation, weighting, and adding offsets on images/image planes, etc, would be highly desirable.
* Code duplication in processing of different file types (e.g. EXR, YUV, and TIFF), and flow limitations. Changes in the order of processing need to currently be performed through code modifications instead of these being specified during runtime.
* Error checking and reporting is limited.
* There are considerable limitations in filtering and other similar processing tools that may be of interest to the video community. The existing code should be reviewed and enhanced with additional functionality.

Another major limitation of the software is the lack of usage documentation. A software manual is necessary, and will be provided in the future.

New software development will continue in the 0.12 development branch of the software that can be accessed from the gitlab location specified in [3].

The creators of this software strongly encourage and expect new contributions to the HDRTools software by the video standardization community experts. However, the creators also request that any such contributions follow certain quality and implementation guidelines, since otherwise these will not be adopted into the software. In particular, the following are requested:

* Any new code should be written in C++.
* Comments and clean code are strongly encouraged.
* New files should be placed in their appropriate locations. If the new files that are introduced involve a generic/common processing functionality, then the files should be added in their appropriate location under the common directory. If a new tool is presented, then a new module should be created under projects and the new files appropriately placed in the new module folders.
* The code should try to maintain the same naming conventions as the remainder of the software. Camelcase is used in naming functions and variables, member variables of a class should always start with “m\_”, while use of private variables and use of get/set functions is encouraged even though that is currently not practiced in the software. Functions and variables should be intelligently named. Memory wasting is strongly discouraged.
* Use of non-standard external libraries is strongly discouraged. If functionality from other libraries is desired, then that should be re-implemented in C++, preferably within the scope of the new module that is introduced (unless it is determined that this functionality is useful elsewhere).
* The new code should be tested on at least two platforms, Windows as well as Linux or Mac OS X (preferred). No warnings should be reported during compilation on either platform.

# References

1. HDRTools Master branch: <https://gitlab.com/standards/HDRTools/>
2. HDRTools version 0.11: <https://gitlab.com/standards/HDRTools/tags/v0.11>
3. HDRTools dev. branch 0.12 dev: <https://gitlab.com/standards/HDRTools/tree/0.12-dev>
4. HDRTools Issue Reporting: <https://gitlab.com/standards/HDRTools/issues>
5. K. Pachauri and S. Sahota, “HDR-VQM Reference Code and its Usage”, JCTVC contribution, JCTVC-W0041, February 2016, San Diego, USA
6. H. R. Tohidypour, M. Azimi, M. T. Pourazad, and P. Nasiopoulos, “Software implementation of visual information fidelity (VIF) for HDRTools”, JCTVC contribution, JCTVC-W0115, February 2016, San Diego, USA
7. A. Norkin, “Closed form HDR 4:2:0 chroma subsampling (HDR CE1 and AHG5 related)”, JCTVC contribution, JCTVC-W0107, February 2016, San Diego, USA
8. C. Rosewarne and V. Kolesnikov, “CE1-related: LUT-based luma sample adjustment”, JCTVC contribution, JCTVC-W0056, February 2016, San Diego, USA
9. A.M. Tourapis, Y. Su, and D. Singer, “Enhanced Luma Adjustment Methods”, JCTVC contribution, JCTVC-W0052, February 2016, San Diego, USA
10. A.M. Tourapis, Y. Su, and D. Singer, “Enhanced Filtering and Interpolation Methods for Video Signals”, JCTVC contribution, JCTVC-W0051, February 2016, San Diego, USA
11. J. Pytlarz, “Overview of IPT-PQ Color Space”, JCTVC-W0050, March 2016.
12. J. Pytlarz, R., T. Lu, F. Pu, P. Yin, T. Chen and W. Husak, “Overview of IPT-PQ Color Space”, MPEG contribution, doc. m37266, Oct. 2015.