 ISO/IEC JTC 1/SC 29/WG 04 N0096

**ISO/IEC JTC 1/SC 29/WG 04  
MPEG Video Coding   
Convenorship: CN**

**Document type:** Output Document

**Title:** User Manual of Reference Lenslet content Convertor (RLC 1.0)

**Status:** Approved

**Date of document:** 2021-05-21

**Source:** ISO/IEC JTC 1/SC 29/WG 04

# Expected action: None

# Action due date: None

**No. of pages:** 6 (without cover page)

**Email of Convenor:** yul@zju.edu.cn

**Committee URL:** <https://isotc.iso.org/livelink/livelink/open/jtc1sc29wg4>

**INTERNATIONAL ORGANIZATION FOR STANDARDIZATION**

**ORGANISATION INTERNATIONALE DE NORMALISATION**

**ISO/IEC JTC 1/SC 29/WG 04 MPEG VIDEO CODING**

**ISO/IEC JTC 1/SC 29/WG 04 N** **0096**

**April 2021, Online**

|  |  |
| --- | --- |
| **Title** | **User Manual of Reference Lenslet content Convertor (RLC 1.0)** |
| **Source** | **WG 04, MPEG Video Coding** |
| **Status** | **Approved** |
| **Serial Number**  **Editor** | **20385**  **Mehrdad Teratani, Sarah Fachada, Daniele Bonatto** |

# Abstract

This document is manual to the Reference Lenslet content Convertor (1.0) [1-7] for lenslet video captured by plenoptic camera 2.0 to Multiview video.

1. The updated source code of (RLC 1.0) is provided in [8]. The update is as follows:
   1. The updated RLC requires OpenCV 4.0 or above.
   2. More microlens alignment are added.
   3. Both Linux and Windows are supported.
2. The software is available for academic and MPEG standardization purposes [10]. No password is required to access this code.

# Introduction

This reference software is currently implemented for conversion of plenoptic camera 2.0. The current convertor works only for focused plenoptic cameras.

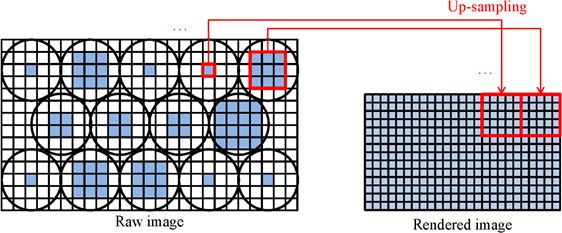
Example of data captured by Raytrix [<https://raytrix.de>], as shown in Figure 1. The multi-focused plenoptic camera (Raytrix) is a powerful device that captures light field (LF), which is interpreted as a set of dense multi-view images. This camera has potential to acquire LFs with high spatial/view resolution and deep depth-of-field. To extract multi-view images, we need a sophisticated rendering process due to the complicated optical system of Raytrix. Nagoya University originally proposed a rendering method, and released source code under Linux platform to the community, for academic and MPEG standardization purposes. The former version is RLC0.3 [(https://www.fujii.nuee.nagoya-u.ac.jp/multiview-data/]((https:/www.fujii.nuee.nagoya-u.ac.jp/multiview-data/)) [7]. In this document, Université Libre de Bruxelles (ULB) provides additional updates mentioned in RLC1.0 [9]. This update does not have any effect on the performance.



**Figure 1.** Raytrix captured lenslet video to Multiview video (Tunnel\_Train 2)

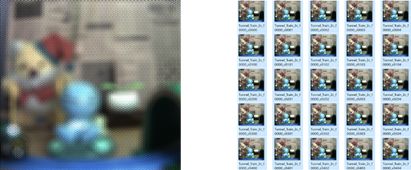
# Conversion from Raytrix to Multiview Video

The straightforward rendering method is to extract not a pixel but a patch with a fixed size from each image of a microlens because pixels behind each microlens include information of several spatial points. However, using a fixed patch size causes severe artifacts. This tool adaptively adjusts the patch size to solve the problem, but few studies have addressed this problem, but they are not robust to camera configurations.



**Figure 2.** Rendering Multiview video from microlens array in focused plenoptic camera.

RLC estimates variable patch in different microlenses and renders the Multiview video, c.f. Figure 2. It follows with integration processing of different lens types for the multi-focused plenoptic camera. Figure 3 shows an example of the conversion. Details of this conversion is explained in [1-8, 10].



**Figure 3:** Conversion from LL to MV video

# Conversion tool

Below, the instruction on how to run the program and parameters used in the source code of the RLC1.0 are explained.

|  |
| --- |
| ===================================================  Prerequisites  ===================================================  1. Boost  2. OpenCV 4.0 or above  3. RLC1.0 can be complied on Windows. RLC0.3 was implemented on Ubuntu 18.04 LTS with g++ 7.3.0 compiler. |

|  |
| --- |
| ===================================================  Instruction to run the program  ===================================================  1 Using Linux and MakeFile  To build the software, execute the make command in the working directory.  $ make  An executable program, named 'converting' will be generated.  2 Using Cmake and VisualStudio  To build the visual studio solution, use the CmakeList.txt provided. A solution that can be directly compiled will be generated in your selected build directory.    To demonstrate the software, we have included some test data captured by Raytrix R5, you can execute 'converting'  with a parameter file.  $ ./converting param.cfg    When the program finishes, it will automatically create a folder for storing the multi-view video. Please make sure that you designate valid file paths.    Notes:  1.      It is recommended to do a test with single frame input data firstly, while you set the parameters to default value.  If you find that the generated three lens-type images (LensType\_1.png, LensType\_2.png and LensType\_3.png) are not arranged with default mode (lmode: FMN), please switch it to the suitable one. By single frame conversion, you may set the other parameters too.  2.      RLC is designed for video processing, so we added frame number on input filename. In the case that you want to process a single image without a frame number, please append a frame number to the filename.  3.      In the case that input data is YUV420 format, please specify height and width of input data.  4.      RLC 1.0 is successfully tested for conversion of raw video captured by Raytrix R5 to multi-view video. |

|  |
| --- |
| ===================================================  Parameters  ===================================================  viewNum 5  rmode 1  pmode 0  mmode 2  lmode 1  Calibration\_xml .././TestDataset/R5\_fujita/CalibData.xml  RawImage\_Path .././TestDataset/R5\_fujita/img%03d.png  Output\_Path .././Output/fujita/Res\_%03d  Debayer\_mode 0  Isfiltering 0  isCLAHE 0  Gamma 1.0  Lambda 0.05  Sigma 0  input\_model 0  output\_model 0  start\_frame 1  end\_frame 2  height 2048  width 2048 |

|  |
| --- |
| ===================================================  Parameter Description  ==================================================  viewNum: Number of views at one direction. For example: viewNum = 5 means 5x5 views in 2D configuration  Currently, we recommend you to set this value to 5 or 7 to ensure good rendering quality from Lenslet video to Multi-view Video.  viewNum = 5 (default value) 5x5 views in 2D configuration  viewNum = 7 7x7 views in 2D configuration  rmode: Resolution mode, and this value must be a integer.  It is corresponded to the "skipped\_number\_of\_pixel\_around\_ML" in the "MultiViewImage.cpp".  (for Raytrix R5)  Resolution mode = 1: 912x880 (default value)  Resolution mode = 2: 832x792  Resolution mode = 3: 744x712  Resolution mode = 4: 656x624  Resolution mode = 5: 568x544  Resolution mode = 6: 480x456  pmode: patch size estimation mode. Please refer to our paper [1][2] for more details.  patchEstimationMode,  LAPLACIAN\_BASE\_ALL = 0, (default value) calculating Laplacian sum using all views,  LAPLACIAN\_BASE\_CENTER = 1, calculating Laplacian sum using center view only.  There are two patch size estimation method using depth information"  DEPTH\_BASE\_V1 = 2,  DEPTH\_BASE\_V2 = 3,  Please refer following paper for the details,  Ref: T. Georgiev and A. Lumsdaine, "Reducing Plenoptic Camera Artifacts", Computer Graphics forum, 2010.  mmode: merge mode for different lens type. Please refer to our paper for more details.  mergeMode AVERAGE = 0, merge different lens type using arithmetic mean,  GAUSSIAN = 1, merge different lens type using Gaussian weights,  BLEND = 2, (default value), merge different lens type using arithmetic mean and Gaussian weights both.  lmode: lens Alignment mode:  NMF = 0, (N = LensType\_1.png, M = LensType\_2.png and F = LensType\_3.png)  FMN = 1, (F = LensType\_1.png, M = LensType\_2.png and N = LensType\_3.png) (default)  NFM = 2, (N = LensType\_1.png, F = LensType\_2.png and M = LensType\_3.png)  FNM = 3 (F = LensType\_1.png, N = LensType\_2.png and M = LensType\_3.png)  MFN = 4, (M = LensType\_1.png, F = LensType\_2.png and N = LensType\_3.png) (new)  MNF = 5, (M = LensType\_1.png, N = LensType\_2.png and F = LensType\_3.png) (new)    This sets the alignment order of three micro-lens type.  NMF means {(N)ear, (M)iddle, (F)ar} focal distances for {LensType\_1.png, LensType\_2.png and LensType\_3.png}, respectively.  For example: NMF represents that the generated lenstype should be arranged with Near focal distance, Middle focal distance, Far distance.  We recommend you to do a test with single frame input data firstly, while you set the parameters to default value.  If you find that the generated three lens-type images (LensType\_1.png, LensType\_2.png and LensType\_3.png)  are not arranged with default mode (lmode: FMN), please switch it to the suitable one.  By single frame conversion, you may set the other parameters too.  ===========Path Setting===============================================  Calibration\_xml: The directory of camera calibration file and the file name.  RawImage\_Path: The directory of lenslet raw images and their file name format.  Output\_Path: The directory for output multi-view video  ===========Pre-processing===============================================  Debayer\_mode: Demosaicing method applied to input raw image.  0 means without Demosaicing processing (default)  1 means BG to RGB; 2 means GB to RGB; 3 means RG to RGB; 4 mean GR to RGB.  Please refer following link for more details about OpenCV demosaicing precessing.  https://docs.opencv.org/3.4.0/de/d25/imgproc\_color\_conversions.html  Isfiltering: Applying a Guassain filter on raw images before patch size estimation.  0 means turn off (default), 1 means turn on.  Please refer following link for more details about CLAHE:  https://docs.opencv.org/3.2.0/d5/daf/tutorial\_py\_histogram\_equalization.html  ============Post-processing===============================================  isCLAHE: Applying CLAHE (Contrast Limited Adaptive Histogram Equalization) on rendering multi-view images.  0 means turn off (default), 1 means turn on.  Gamma: This is to adjust parameter of Gamma correction.  A positive floating point number is needed, we set this parameter to 1.0 (default).  Please refer following link for more details about Gammar correction:  https://docs.opencv.org/3.4/d3/dc1/tutorial\_basic\_linear\_transform.html  ============Temporal Consistency===============================================  Lambda: This value should be a positive floating point number.  It is the coeffient of penalty of temporal consitency.  0 means without penalty of temporal consitency.  We recommend setting this parameter to 0.05. (default)  Sigma: This value should be a floating point number.  It is the sigma of Gaussian funtion of temporal consitency.  0 means without Gaussian filter for temporal consitency. (default)  If you want to activate this function, we recommend setting this parameter to 0.667.  ============Others===============================================  input\_model: 0 or 1; 0 means RGB model, 1 means YUV420 model  output\_model: 0 or 1; 0 means RGB model, 1 means YUV420 model  start\_frame: To set the start frame number of input data.  end\_frame: To set the end frame number of input data. |

# Summary

An updated source code of the Reference Lenslet content Convertor (RLC 1.0) developed by Nagoya University is released by Université Libre de Bruxelles. RLC1.0 is only available for academic and MPEG standardization activities. The source code of RLC1.0 is used as software platform and reference software for exploration experiments on dense light field activity of Ad Hoc Group on Lenslet Video Coding (LVC). The updated tool can convert Lenslet data captured by any type of plenoptic 2.0 in colour domain.

**Coordination:**

Sarah Fachada, [sarah.fernandes.pinto.fachada@ulb.be](mailto:sarah.fernandes.pinto.fachada@ulb.be)

Daniele Bonatto, [daniele.bonatto@ulb.be](mailto:daniele.bonatto@ulb.be)

Mehrdad Teratani, [mehrdad.teratani@ulb.be](mailto:mehrdad.teratani@ulb.be)

**Reference**

[1] Mehrdad Teratani, Shu Fujita, Wenzhe Ouyang, Keita Takahashi, Toshiaki Fujii, "3D Imaging System Using Multi-Focus Plenoptic Camera and Tensor Display," 2018 International Conference on 3D Immersion (IC3D2018) in IEEE Xplore Digital Library, Brussels, Belgium, December 2018.

[2] Shu Fujita, Sho Mikawa, Mehrdad Teratan, Keita Takahashi, and Toshiaki Fujii, "Extracting Multi-View Images from Multi-Focused Plenoptic Camera, " International Workshop on Advanced Image Technology (IWAIT)/International Forum on Medical Imaging in Asia (IFMIA), in SPIE Digital Library, Singapore, January 2019.

[3] Mehrdad Teratani, Sho Mikawa, Shu Fujita, Keita, Toshiaki Fujii, Response to EE\_conversion on Dense Light Field- Light Field Video to Multiview Video Conversion Tool”, ISO/IEC JTC1/SC29/WG11, M42693, San Diego, USA, April 2018.

[4] Mehrdad Teratani, Shu Fujita, Wenzhe Ouyang, Keita, Toshiaki Fujii, “[MPEG-I Visual] Conversion Tool – Raytrix Data to Multiview Video”, ISO/IEC JTC1/SC29/WG11, M44731, Macao, SRA China, October 2018.

[5] Mehrdad Teratani, Wenzhe Ouyang, Toshiaki Fujii, “[MPEG-I Visual] Temporal Consistency Algorithms for Reference Lenslet content Convertor”, ISO/IEC JTC1/SC29/WG11, M49668, Gothenburg, SE, July 2019.

[6] Mehrdad Teratani, Wenzhe Ouyang, Toshiaki Fujii, “[MPEG-I Visual] Update on Reference Lenslet content Convertor (RLC0.2)”, ISO/IEC JTC1/SC29/WG11, M49665, Gothenburg, SE, July 2019.

[7] Mehrdad Teratani, Toshiaki Fujii, “Manual of Reference Lenslet Content Convertor (RLC 0.3)”, ISO/IEC JTC1/SC29/WG11, N18567, Gothenburg, SE, July 2019.

[8] Sarah Fachada, Daniele Bonatto, Gauthier Lafruit, Mehrdad Teratani, “[DLF] Update for Reference Lenslet content Convertor (RLC1.0)” ISO/IECJTC1/SC29/WG04, M56418, Online, April 2021.

[9] <http://mpegx.int-evry.fr/software/MPEG/Video/lvc/rs/rlc>