ISO/IEC JTC 1/SC 29/WG 1
(& ITU-T SG16)

Coding of Still Pictures

JBIG JPEG

Joint Bi-level Image Experts Group	Joint Photographic Experts Group

TITLE: **JPEG PLENO LIGHT FIELD CODING COMMON TEST CONDITIONS V3.3**

SOURCE: WG1

CONTRIBUTORS: Fernando Pereira, Carla Pagliari, Eduardo da Silva, Ioan Tabus, Hadi Amirpour, Marco Bernardo, Antonio Pinheiro

PROJECT: ISO/IEC 21794-1 (JPEG PLENO)

STATUS: Draft

DISTRIBUTION: Public

Contact:
ISO/IEC JTC 1/SC 29/WG 1 Convener – Prof. Touradj Ebrahimi
EPFL/STI/IEL/GR-EB, Station 11, CH-1015 Lausanne, Switzerland
Tel: +41 21 693 2606, Fax: +41 21 693 7600, E-mail: Touradj.Ebrahimi@epfl.ch

JPEG PLENO - LIGHT FIELD CODING COMMON TEST CONDITIONS

CONTENTS

JPEG PLENO LIGHT FIELD CODING COMMON TEST CONDITIONS

1 Scope

This document describes the Common Test Conditions for the JPEG Pleno light field coding experiments.

2 Test Materials

This section describes the test material selected for JPEG Pleno core and exploration experiments. The selection is justified using an appropriate descriptor, which is able to ascertain the geometric diversity of the light field images in the test set.

2.1 JPEG Pleno Light Field Images Test Set

The JPEG Pleno light field images test set is diverse in terms of

- Acquisition/creation technology, notably
    - Lenslet Lytro Illum camera
    - High Density Camera Array (HDCA)
    - Synthetic creation
- Scene geometry
- Spatial resolution
- Number of views/perspectives
- Bit depth
- Texture

In the following, the JPEG Pleno test set will be organized according to the acquisition/creation technology as this factor determines several content characteristics, notably the views baseline and amount of interview redundancy.

2.1.1 Lenslet Lytro Illum Camera

The Lenslet Lytro Illum Camera data set is part of the EPFL dataset described in [1][2]. Information related to copyright can be found at http://mmspg.epfl.ch/EPFL-light-field-image-dataset. The light field images were captured using a Lytro Illum B01 (10-bit) light field camera. According to the JPEG Pleno Call for Proposals [3], the Matlab implementation of the Light Field Toolbox v0.42 [4] has been employed to convert the lenslet light field camera raw sensor data to sub-aperture images including the demosaicing and devignetting pre-processing steps. The sub-aperture images, containing appropriate samples from the micro-lenses, are stacked forming a 5D array of 15×15×625×434×4 dimensions. An additional version of the sub-aperture images has been made available with spatial resolution 15×15×626×434×4 by adding a column of zeros to have a resolution more compatible with HEVC coding, which was one of the anchors selected for the Call for Proposals.

Currently, the JPEG Pleno test set includes four lenslet light field images in the Lenslet Lytro Illum Camera data set, notably ***Bikes***, ***Danger de Mort***, ***Stone Pillars Outside*** and ***Fountain&Vincent 2****,* see Figure 1.

- **Link –** Sub-aperture images should be made available at the JPEG Pleno database, https://jpeg.org/jpegpleno/plenodb.html.
- **Format –** PPM images (RGB color components, non-interlaced).

- **Content –** Natural, outdoors.
- **Number of views -** 15×15 views but only the central 13×13 views are used to avoid using the dark views associated to vignetting.
- **Spatial resolution -** 625×434.
- **Bit depth –** 10-bit.
- **Auxiliary data** - Depth map for central sub-aperture image.

2.1.2 Fraunhofer High Density Camera Array (HDCA)

The Fraunhofer High Density Camera Array (HDCA) has been generated using a robotized high quality camera [5,6]. The geometry of acquisition employed a horizontal step size of 4 mm (distance between adjacent cameras in the horizontal direction) and a vertical step size of 6 mm (distance between adjacent cameras in the vertical direction). The original HDCA 14-bit test materials, with 101 horizontal images and 21 vertical images of 7956×5304 spatial resolution were made available in a 10-bit version already pre-processed, which included global registration, as well as resolution and chroma down-sampling by a factor of 2 [6]. The 10-bit version with a spatial resolution of 3840×2160 pixels has been adopted; this region corresponds to a cropping of the subsampled resolution to the region 85-3924 in horizontal and 209-2368 in vertical directions (with numbering starting with '0') [5].

Currently, the JPEG Pleno test set includes only one light field image from the four initially provided in the context of the Fraunhofer HDCA set. Figure 2 shows the four images initially made available with *Set2* being the single one currently adopted for computational complexity reasons. Moreover, as even a single full set was very heavy to encode, the selected ***Set2*** light field image has been subsampled in terms of views and each view cropped in terms of spatial resolution to 1920×1080; this light field is referred as ***Set2 2K sub***.

- **Link –** https://www.iis.fraunhofer.de/en/ff/amm/dl/lightfielddataset.html
- **Format –** PPM images (RGB color components, non-interlaced).
- **Content –** Indoor images with different levels of detail, specularities, regular patterns and objects at different depths.
- **Number of views -** 101×21 views in the full version and 33×11 in the view subsampled version.
- **Spatial resolution -** 3840×2160 for the full version and a cropped part with 1920×1080 resolution for the subsampled version; the lower resolution corresponds to a cropping of the full resolution views which takes the center 1920×1080 portion of the image (notably the region 85-3924 in horizontal and 209-2368 in vertical directions (with numbering starting with '0')) and discards the rest [7], see Figure 3
- **Bit depth –** 10-bit.
- **Auxiliary data** - NA.

Figure 1 - Bikes, Danger de Mort, Stone Pillars Outside and Fountain&Vincent 2 example views.

Figure 2 - Set 2, Set 6, Set 9 and Set 10 example views.

Figure 3 - Set2 2K sub example view.

2.1.3 Poznan HDCA

The Poznan HDCA light field images have been acquired with a 2D parallel array of cameras with 10 mm spacing for both vertical and horizontal directions [9, 10]. The sequences remain the property of Poznan University of Technology but they are licensed for free use within ISO/IEC JTC1/SC29/WG11 (MPEG) and ISO/IEC JTC1/SC29/WG01 (JPEG) for the purposes of research and development of standards [9].

Currently, the JPEG Pleno test set includes only one light field image from the two provided in the context of the Poznan HDCA set, named ***Laboratory 1***, see Figure 4.

- **Link -** ftp://multimedia.edu.pl/ftv
- **Format –** Individual TIFF images [11] (RGB color components, non-interlaced).
- **Content –** Natural, indoors.
- **Number of views –** 31×31.
- **Spatial resolution –** 1936×1288; these images have been acquired as JPEG images with a resolution of 5184×3456, post-processed to remove lens distortions and subsampled to 1936×1288 [9, 10].
- **Bit depth –** 8-bit (available at https://jpeg.org/jpegpleno/plenodb.html as 10-bit PPM).
- **Auxiliary data** – NA.

Figure 4 - Laboratory 1 example view.

2.1.4 Stanford HDCA

The Computer Graphics Laboratory at Stanford University has acquired several light field images for research in computer graphics and vision and they are made publicly available to researchers [12]. The light fields may be used for academic and research purposes, but are not to be used for commercial purposes, nor should they appear in a product for sale without permission. If you use these models in a publication, please credit the Stanford Computer Graphics Laboratory.

Currently, the JPEG Pleno test set includes only one light field image in the Stanford HDCA data set from the many images available in [12]. The selected light field image is shown in Figure 5 and has been named ***Tarot Cards***; it corresponds to the Tarot Cards & Crystal Ball (small angular extent) image available at [12]. ***This dataset has positive and negative disparities.***

- **Link -** http://lightfield.stanford.edu/
- **Format –** Views available as individual PNG images [13], (RGB color components, non-interlaced).
- **Content –** Indoors with complex specularities.
- **Number of views –** 17×17.
- **Spatial resolution -** 1024×1024.
- **Bit depth –** 8-bit (available at https://jpeg.org/jpegpleno/plenodb.html as 10-bit PPM).
- **Auxiliary data** – For some scenes, calibration information (such as homographies for keystone correction and camera positions) is provided as a text file; for others, simple calibration targets are visible in the original camera images [12].

Figure 5 - Tarot Cards example view.

2.1.5 Synthetic HCI HDCA

These light field images were synthetically created and are made available by the Heidelberg Collaboratory for Image Processing (HCI) [14]. Two light field images from the Synthetic HCI HDCA data set are included in the JPEG Pleno sets, notably ***Greek*** and ***Sideboard***, see Figure 6.

The light field images are made available with the following characteristics and auxiliary disparity data:

- **Link** - http://hci-lightfield.iwr.uni-heidelberg.de/
- **Content –** Synthetic objects
- **Format** – Views available as individual PNG [13] images (RGB color components, non-interlaced).
- **Number of views** - 9×9
- **Spatial resolution** - 512×512
- **Bit depth –** 8-bit (available at https://jpeg.org/jpegpleno/plenodb.html as 10-bit PPM).
- **Auxiliary data -** Configuration files with camera settings and disparity ranges; for the center view (except for four images), 512×512 and 5120×5120 depth and disparity maps are provided as PFMs [15].

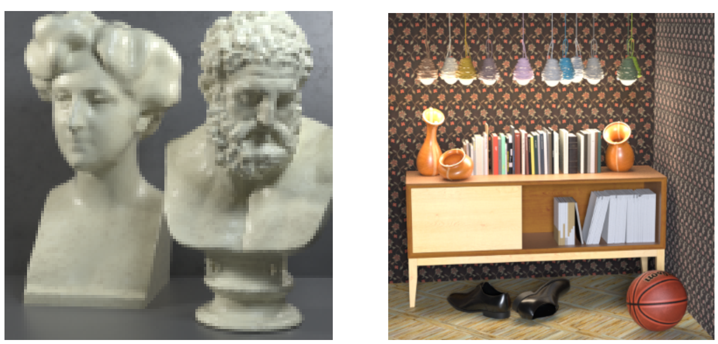


Figure 6 - Greek and Sideboard example views.

2.2 Analysis of the JPEG Pleno Light Field Images Data Set

The JPEG Pleno light fields constituting the test set have been selected considering the criteria listed above. Furthermore a quantitative descriptor, so-called *Geometric Space-View Redundancy* (GSVR) descriptor, has been used to characterize each light field image [16]. The GSVR descriptor expresses information about the trade-offs involved in effectively exploring the 4D space-view redundancy (joint intra- and inter-view redundancy) of a light field for efficient coding. Such redundancy is measured by the probability, for each space block size and range of views, that the image of a point in 3D space remains in the space block in all views. A large permanence probability inside a 4D block means that the image of a point in 3D space traces a long trajectory inside it, thus generating many redundant points inside the 4D block, which is equivalent to saying that there is significant 4D space-view redundancy to be exploited. The GSVR descriptor thus gives the relation between the intra-view and inter-view block dimensions guaranteeing the existence of such 4D space-view redundancy with a given probability. In brief, for a permanence probability *p, GSVR(p)* is defined as the ratio of the required intra-view block size given an inter-view block size guaranteeing this permanence probability. If the GSVR indicates that the intra-view block size should be much larger than the inter-view block size for proper exploration of the space-view redundancy, this implies that the light field has a small inter-view redundancy, requiring a large intra-view block size to exploit it, and vice and versa. In the following, the JPEG Pleno data set is characterized by their GSVR descriptors, thus allowing to analyse how different are the several selected light field images in terms of how easy is to exploit their 4D space-view redundancy *Figure 7* to *Figure **9*** show the GSVR curves for the selected JPEG Pleno light fields test set, notably:

- **Lenslet Lytro Illum camera light field images** – As shown in Figure 7 the four lenslet light field images have rather low GSVR values, thus indicating that the inter-view redundancy of a 4D block can be well exploited using space blocks of small sizes. *Fountain&Vincent2* has the highest GSVR values, while *Stone Pillars Outside* presents the lowest GSVR values.
- **Fraunhofer HDCA Full and 2K Subsampled light field images** – Figure 8 left) clearly shows that three out of the four Fraunhofer HDCA light field images are very similar in terms of GSVR, notably *Set6, Set9* and *Set10*. As these images are very heavy to process, it was decided to select only one for the JPEG Pleno data set, notably *Set2*. Still for computational complexity reasons the Fraunhofer HDCA 2K subsampled images (33x11 views) have been created but it is clear that this had significant implications on the resulting HDCA content, notably HDCA sets less dense in space as confirmed by the difference between the GSVR values for a specific full and 2K subsampled set; as expected the GSVR values are higher for the 2K subsampled sets as the 2K subsampled dataset is more sparsely sampled. This implies that the exploitation of the 4D space-view redundancy of the 2K subsampled HDCA content would require intra-view block sizes that are much larger than the inter-view block sizes, and thus the 4D space-view redundancy tends to be harder to exploit than in the case of the non-subsampled dataset.
- **All HDCA light field images** – As shown in Figure 9 left), the set of selected HDCA light field images (Fraunhofer HDCA, Stanford HDCA and Poznan HDCA) covers a large range of GSVR curves, notably GSVR values from 3 to 26 for a permanence probability *p=0.9*. This guarantees some diversity in terms of scene and content characteristics without increasing too much the size of the data set and thus the overall complexity to run experiments.
- **All JPEG Pleno light field images** – The full JPEG Pleno light field images data set shows a large variety of GSVR behaviours, see Figure 9 right), thus highlighting that rather different light field images are being exercised in the development of the JPEG Pleno light field coding standard.

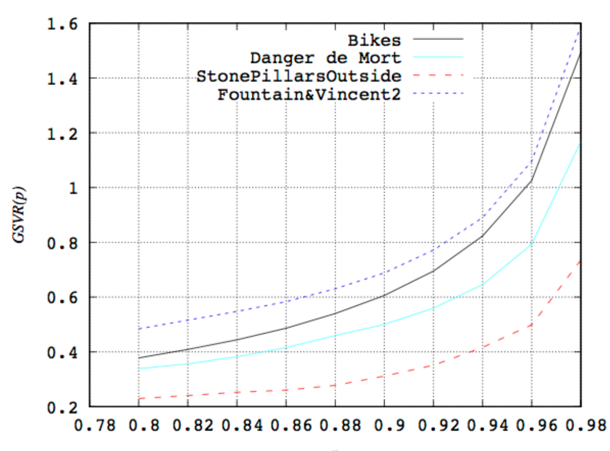
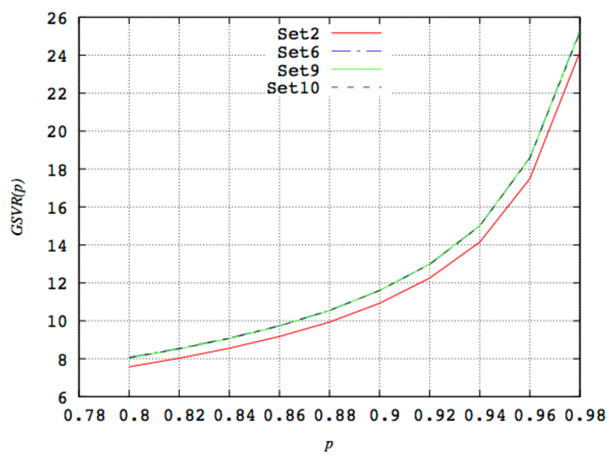


Figure 7 - GSVR curves for the lenslet light field images.



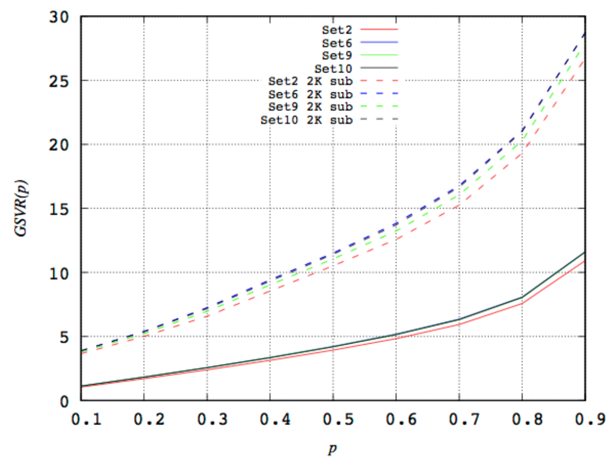
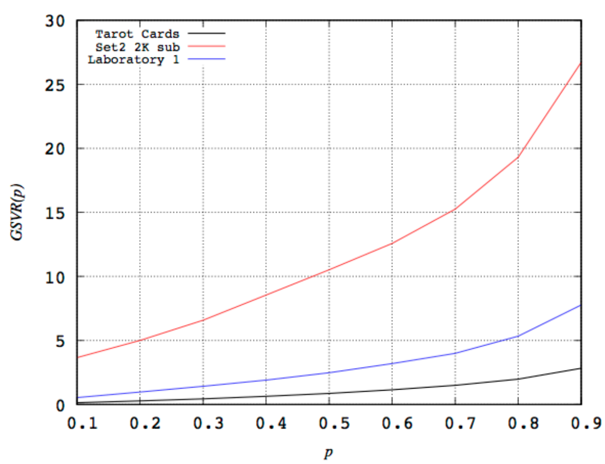


Figure 8 - GSVR curves for the Fraunhofer HDCA light field images: left) Full; right) Full and Subsampled.



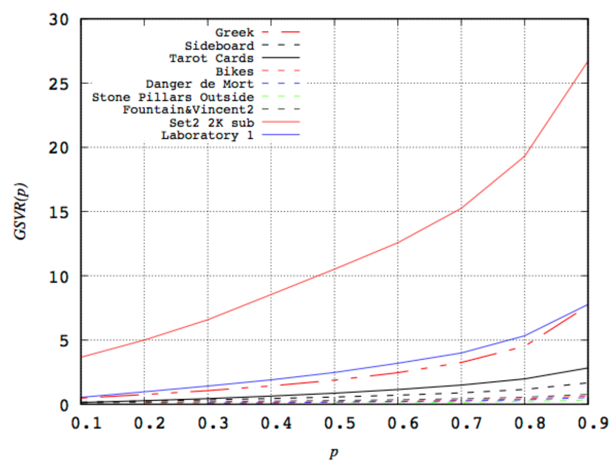


Figure 9 - GSVR curves: left) All HDCA light field images; right) All JPEG Pleno light field images.

2.3 JPEG Pleno Light Field Images Data Set: Summary

Table 1 includes a summary of the selected JPEG Pleno light field images.

Table 1 – Summary of the selected JPEG Pleno light field images.

Source	Type	Light field image name	Number of views	Spatial resolution (pixels)	Link
EPFL	LL	Bikes Danger de Mort Stone Pillars Outside Fountain & Vincent 2	15×15	625 × 434	https://jpeg.org/jpegpleno/plenodb.html
Fraunhofer IIS	HDCA	Set 2 2K sub	33×11	1920 × 1080	, https://jpeg.org/jpegpleno/plenodb.html
Poznan University of Technology	HDCA	Laboratory1	31×31	1936 × 1288	https://jpeg.org/jpegpleno/plenodb.html
New Stanford LF Archive	HDCA	Tarot Cards	17×17	1024 × 1024	https://jpeg.org/jpegpleno/plenodb.html
HCI	HDCA	Greek Sideboard	9×9	512 × 512	https://jpeg.org/jpegpleno/plenodb.html

3 Definition of Performance Metrics

This section defines the image quality metrics that will be used in JPEG Pleno core experiments.

3.1 Quality Metrics (QM)[1]

The Matlab scripts described below are available to compute the metrics; for 10-bit images 10-bit metrics should be used, for 8-bit images 8-bit metrics should be used.

[1] All references to Matlab in this section refer to Matlab version '9.4.0.813654 (R2018a)'

3.1.1 RGB to YCbCr Conversion

The 10-bit RGB images are converted to double precision and these double precision RGB images are converted to double precision Y, Cb, Cr using the ITU-R BT.709-6 recommendation [19]. These double precision Y, Cb, Cr images are further rounded to integers. The following Matlab script (rgb2ycbcrn.m) implements this conversion (the input parameter *n*, which gives the number of bits, should be equal to 10):

```
% Author :  Hadi Amirpour
% email  :  hadi.amirpour@gmail.com
% Copyright(c) EmergIMG,
%              Universidade da Beira Interior

% script for RGB 444 to YCbCr 444 color space conversion in n bits
% Input:
        % 1-rgb   ---> RGB image in double format
        % 2-n     ---> number of the bits, this number should be either 8 or 10

% Output:
        % 1-ycbcr ---> YCbCr image in n bits

function [ycbcr] = rgb2ycbcrn(rgb,n)

%  Recommendation ITU-R BT.709-6
M = [ 0.212600   0.715200  0.072200 ;
     -0.114572  -0.385428  0.500000 ;
      0.500000  -0.454153 -0.045847];

if (nargin < 2)
    n = 8;
end

ycbcr          = reshape(double(rgb), [], 3) * M';
ycbcr          = reshape(ycbcr, size(rgb));
ycbcr(:,:,1)   = (219*ycbcr(:,:,1)+16)*2^(n-8); %Luminance
ycbcr(:,:,2:3) = (224*ycbcr(:,:,2:3) + 128)*2^(n-8);

if(n==8)
    ycbcr = uint8(ycbcr);
elseif (n==10 || n==16)
    ycbcr = uint16(ycbcr);
else
    print('invalid bit depth')

end
```

3.1.2 PSNR Definition and Computation

The *PSNR* between the original component, *I,* and the reconstructed component, *I',* (both n-bit) is computed as follows:

$$PSNR = 10log_{10}\left(\frac{2^n - 1}{MSE}\right)$$

where the MSE between the two M×N images, *I* and *I',* is given by:

$$MSE = \frac{1}{MN}\sum_{i=0}^{M-1}\sum_{j=0}^{N-1}(I(i,j) - I'(i,j))^2$$

Once the $PSNR_Y$, $PSNR_{Cb}$ and $PSNR_{Cr}$ are computed, one should compute the $PSNR_{YCbCr}$ as follows:

$$PSNR_{YCbCr} = \frac{6 * PSNR_Y + PSNR_{Cb} + PSNR_{Cr}}{8}$$

The PSNRs for the whole light field are computed by averaging the PSNRs of the individual views or sub-aperture images. The Matlab script (QM.m) in implements the PSNR computations (it uses Matlab built-in function 'psnr' to evaluate $PSNR_Y$, $PSNR_{Cb}$ and $PSNR_{Cr}$). It is described at the end of Sub-section 3.1.3. $PSNR_Y$ and $PSNR_{YCbCr}$ calculations should be reported.

3.1.3 SSIM Definition and Computation

The SSIM is defined according to [20]. The SSIM for the whole light field is computed by averaging the SSIM values of the individual views or sub-aperture images.

The Matlab script (QM.m) implements the SSIM computations using the Matlab built-in function 'ssim' to evaluate the SSIM. The SSIM is computed only for the luminance component (Y). The input images should be double precision, with pixels normalized to be in the interval [0,1].

The following Matlab script (QM.m) implements the PSNRs and SSIM computations:

```
% Author :  Hadi Amirpour
% email  :  hadi.amirpour@gmail.com
% Copyright(c) EmergIMG,
%           Universidade da Beira Interior
% inputs :
        %ref   = reference image,
        %rec   = reconstructed image,
        %n_rgb = number of the bits for RGB inputs, this number should be either 8 or 10.
        %n_yuv = number of bits for YUVs, this number is the same as n_rgb
% outputs:
        %Y_PSNR
        %YUV_PSNR
        %Y_SSIM

function [Y_PSNR YUV_PSNR Y_SSIM]=QM(ref,rec,n_rgb,n_yuv)

   %convert rgb to double
   ref_double = double(ref)./(2^n_rgb-1);
   rec_double = double(rec)./(2^n_rgb-1);

   %convert double rgb images to n_yuv bits ycbcr (4:4:4)
   ref_YCbCr_444=rgb2ycbcrn(ref_double,n_yuv);
   rec_YCbCr_444=rgb2ycbcrn(rec_double,n_yuv);



   Y1 = ref_YCbCr_444(:,:,1);
   U1 = ref_YCbCr_444(:,:,2);
   V1 = ref_YCbCr_444(:,:,3);

   Y2 = rec_YCbCr_444(:,:,1);
   U2 = rec_YCbCr_444(:,:,2);
   V2 = rec_YCbCr_444(:,:,3);


   % Objective metrics
```

```
   Y_MSE=immse(Y1,Y2);
   U_MSE=immse(U1,U2);
   V_MSE=immse(V1,V2);

   Y_PSNR   = 10*log10((2^n_yuv-1)*(2^n_yuv-1)/Y_MSE);
   U_PSNR   = 10*log10((2^n_yuv-1)*(2^n_yuv-1)/U_MSE);
   V_PSNR   = 10*log10((2^n_yuv-1)*(2^n_yuv-1)/V_MSE);

   YUV_PSNR = (6*Y_PSNR+U_PSNR+V_PSNR)/8;

   Y_SSIM = ssim(double(Y1)./(2^n_yuv-1),double(Y2)./(2^n_yuv-1));
end
```

Note that Matlab function imread when reading an image file scales it to either 8-bit or 16-bit. Therefore, for 10-bit PPM images (img1.ppm and img2.ppm) one should proceed as follows:

```
im1=imread('img1.ppm');
im2 =imread('img2.ppm');

ref=bitshift(im1,-6); rec=bitshift(im2,-6);

[Y_PSNR YUV_PSNR Y_SSIM]=QM(ref,rec,10,10)
```

3.2 Bjøntegaard Metric

Use of the Bjøntegaard metric that provides numerical averages between RD-curves as part of the presentation of results.

The Matlab code for computing BD-Bitrate (BD-BR) and BD-PSNR is found in [26].

3.3 Coding Conditions

This section describes the target bitrates for the experiments performed with JPEG Pleno dataset, see *Table **2*** .

Table 2 – Target bitrates for the JPEG Pleno light field datasets.

Dataset	**Target Bitrate (bpp)**					
All Lenslets		0.001[2]	0.005[1]	0.02[1]	0.1[1]	0.75[1]
Greek and Sideboard		0.001[2]	0.005[1]	0.02[1]	0.1[1]	0.75[1]
Tarot Cards		0.001[2]	0.005[1]	0.02[1]	0.1[1]	0.75[1]
Set2 2K sub	0.0005[2]	0.001[2]	0.005[1]	0.01[1,2]	0.05[2]	0.1[2]
Laboratory 1	0.0005[2]	0.001[2]	0.005[1]	0.01[1,2]	0.05[2]	0.1[2]

[1]See reference [17]

[2]See reference [18]

3.4 Rate Metrics

This subsection describes the rate metrics adopted as well as the way they should be computed.

The bitrates specified in the test conditions above and reported for the experiments with the various codecs should account for the total number of bits, *N_TOT_BITS*, necessary for generating the encoded file (or files) out of which the decoder can reconstruct a lossy or lossless version of the entire input light field.

The main rate metric is the number of bits per pixel (bpp) defined as:

$$BPP = \frac{N\_TOT\_BITS}{N\_TOT\_PIXELS} \qquad \textbf{Equation 1}$$

where *N_TOT_BITS* is the number of bits for the compressed representation of the light field and *N_TOT_PIXELS* is the number of pixels in the whole light field which also corresponds to the number of luminance samples in the whole light field.

For encoding the HDCA Set2 2K sub dataset that has 33×11 views, each with 1920×1080 resolution, the total number of pixels is *N_TOT_PIXELS* = 33×11×1920×1080= 752,716,800 pixels.

For encoding one lenslet dataset that has 13×13 views, each with 625×434 resolution, the total number of pixels is *N_TOT_PIXELS* = 13×13×625×434= 45,841,250 pixels.

3.5 Random Access Metric

This is not the sole criterion but just one criterion for determining random access capability.

The random access penalty metric is defined as the ratio of amount of encoded bits required to access a Region of Interest (RoI) and the total amount of encoded bits as shown in:

$$Random\ Access\ Penalty = \frac{Total\ amount\ of\ encoded\ bits\ required\ to\ access\ an\ RoI}{Total\ amount\ of\ encoded\ bits\ to\ decode\ the\ full\ light\ field}$$

The total amount of encoded bits required to access an RoI such as: bits in encoded Reference Views to access the RoI, bits in IV prediction and estimation parameters to access an RoI, bits in Metadata to access an RoI, bits in prediction residuals to access an RoI, bits in encoded disparity data to access an RoI.

The result should be reported for the specific RoI type that gives the worst possible result for the Random Access Penalty Metric. For example, if the RoI is a single view, the Random Access Penalty should be reported for the view that gives the largest number.

3.5.1 Examples of RoIs for Random Access:

Types of RoIs for random access are based on the different types of rendering operations that can be done on the light field data. References [27] and [28] illustrate the use of various RoIs in light field rendering operations.

3.5.1.1.1 Single View

A single sub-aperture view has to be accessed from the encoded light field

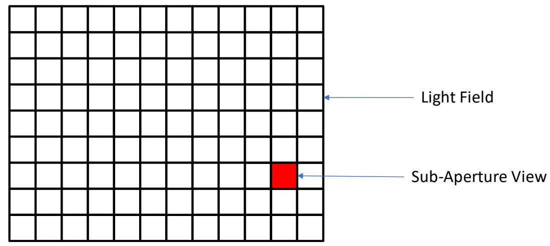


Figure 10. Example of a single sub-aperture view from the light field

3.5.1.1.2 Multiple Views

Multiple sub-aperture views have to be accessed from the encoded light field

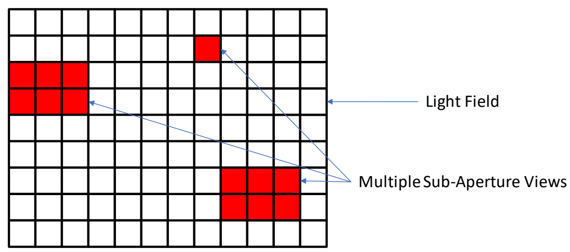


Figure 11. Example of multiple sub-aperture views from the light field

For multiple views a 4x3 views region should be considered.

3.5.1.1.3 Single pixel from all views

A single pixel from all the sub-aperture images from the same location in the sub-aperture images.

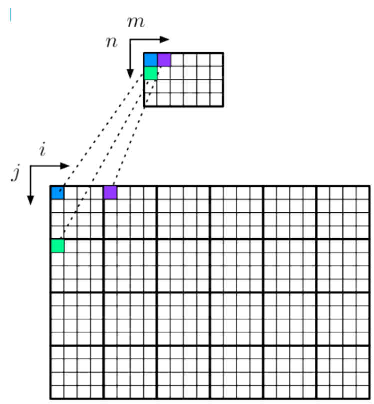


Figure 12. Example of a single pixel from all the sub-aperture images from the same location in the sub-aperture image. Image from [28]

3.5.1.1.4 A group of pixels from all views

A group of pixels from all the sub-aperture images from the same location in the sub-aperture images.

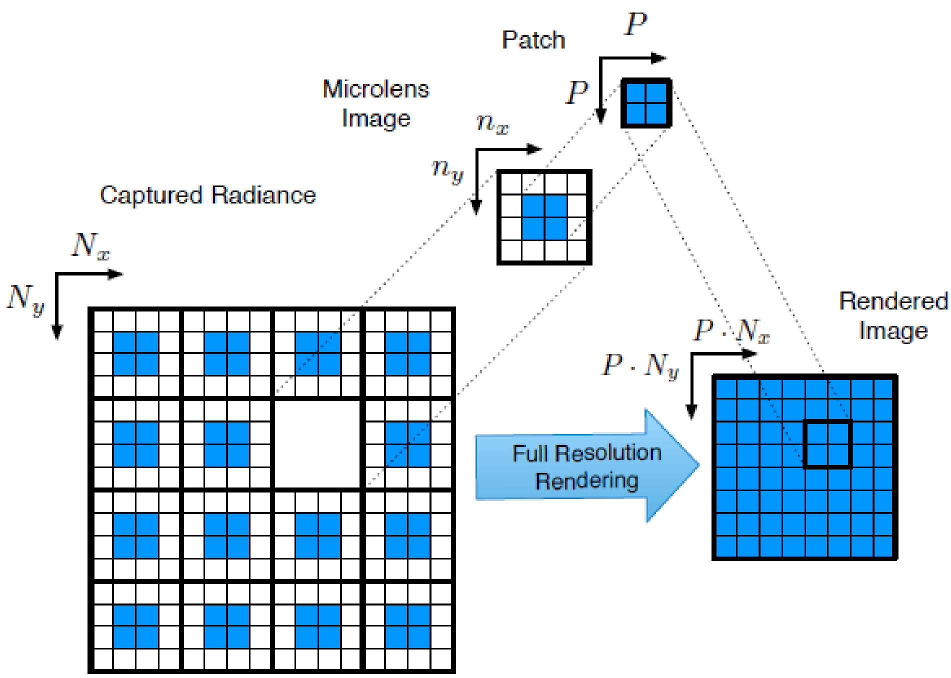


Figure 13 Example of a group of pixels from all the sub-aperture images from the same location in the sub-aperture images. Image from [28].

A 64x64 region from center of each view should be decoded.

4 Image and Data Outputs

This section describes the various image and data outputs which should allow assessing the codec performance.

4.1 Naming Convention for Images

The naming convention for the proponent generated image material is based on the following schema

Type_DataSetID_FramePosition_Rate.ppm

where:

- Type: either “LL” (Lenslet) or “HDCA” (High density camera array)
- DataSetID:
  - In case of LL: Bikes, Danger_de_Mort, Stone_Pillars_Outside and Fountain&Vincent2
  - In case of HDCA: Set2 2K sub, Tarot Cards, Laboratory1, Greek and Sideboard.
- FramePosition In the format “xxx_yyy” where xxx = the frame index horizontally and yyy = the frame index vertically
- Rate: The target bitrate used to code the frame, expressed in units of 10^{-6} bpp

Using this convention, the frame located at (column, row) position (10, 20) in the HDCA data set Set2 2K sub, coded at a bit rate of 0.00025 bpp would be named as follows:

HDCA_Set2_2K_sub_010_020_000250.ppm

4.2 Output data and metrics required

Depending on the experiment, the participants may be required to provide the output data listed in Table 3.

Table 3 – Output data required.

Image Files	Format	Requirement
Decoded Reference Views (RV)	PPM	All decoded RVs for each bit rate must be provided
Decoded Intermediate Views (IV)	PPM	1× decoded IV with max PSNR for each bit rate 1× decoded IV with min PSNR for each bit rate Participant must be prepared to provide additional views within 24 hours, as requested
Decoded Disparity Data	PPM	Participant must be prepared to provide these views within 24 hours, as requested
Predicted Intermediate Views	PPM	Participant must be prepared to provide these views within 24 hours, as requested
Prediction Error Images	PPM	Participant must be prepared to provide these views within 24 hours, as requested

4.3 Objective Metrics Output Requirements

Depending on the experiment, the participants may be required to provide the objective metrics listed in Table 4.

Table 4 – Objective metrics required.

Objective Metric	Format	Requirement
#bits spent on RVs	# bits	Provide for each bit rate
#bits spent on Disparity Data	# bits	Provide for each bit rate
#bits spent on Prediction Coefficients	# bits	Provide for each bit rate
#bits spent on Prediction Residuals	# bits	Provide for each bit rate
PSNR/SSIM of Decoded RVs	PSNR/SSIM	Provide Average PSNR/SSIM, for each bit rate
PSNR/SSIM of Decoded Disparity Data	PSNR/SSIM	Provide Average PSNR/SSIM, for each bit rate
PSNR/SSIM of Predicted Intermediate Views	PSNR/SSIM	Provide Average PSNR/SSIM, for each bit rate
PSNR/SSIM of Decoded Intermediate Views	PSNR/SSIM	Provide Average PSNR/SSIM, minimum PSNR/SSIM, maximum PSNR/SSIM, and standard deviation of PSNR/SSIM, for each bit rate
Overall RD Performance Graphs	Plot	Plot of PSNR/SSIM vs. bit rate
BD-BR and BD-PSNR of Decoded RVs	Table	Obtained from Average PSNR, for each bit rate
Random Access Penalty Metric	Table	Calculated for each bit rate as defined in Section 3.3 at defined RoI size and type.

5 HEVC Anchor Generation

This section describes the anchor generation process as detailed in [17]; the selected anchor is the HEVC standard. The encoding/decoding pipeline is presented in Figure 14:

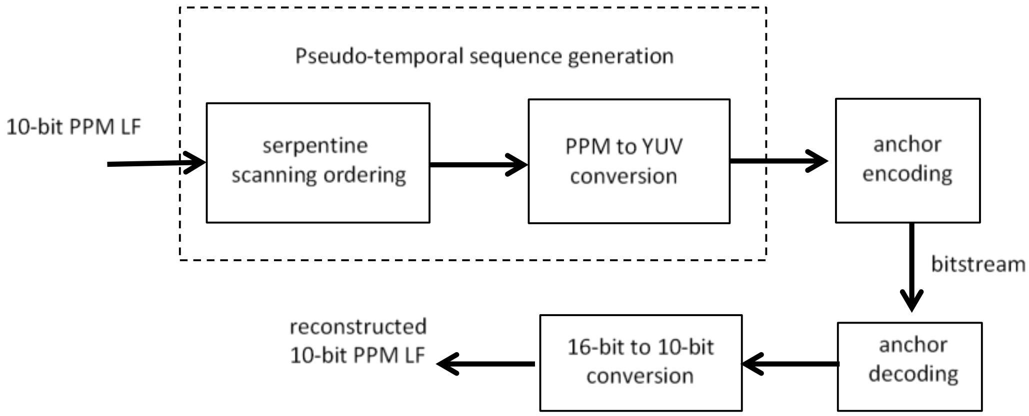


Figure 14 - Encoding/decoding pipeline.

The pseudo-temporal sequence generation is described in sub-section 5.1.The anchor encoding is performed by the x.265 implementation (x265.org) of HEVC, detailed in sub-section 5.2, while the decoding process and the 16-bit to 10-bit conversion is described in sub-section 5.3.

5.1 Pseudo-Temporal Sequence Generation

The input file for the anchor encoding scheme is an YUV 4:4:4 10-bit file (in ppm file format), scanned and concatenated following a serpentine scanning order, as defined in the JPEG Pleno input document [23], repeated here for convenience as Figure 15. Please note that, the PPM to YUV conversion needs to use the ITU-R Recommendation BT. 709-6 [19].

An example on how to generate such a pseudo-temporal for a 3×3 light field ABCD with dimensions 400×300 is given below:

ffmpeg -r 30 -f concat -safe 0 –I list.txt -s 400x300 -framerate 30 -c:v rawvideo -pix_fmt yuv444p10le[2] ABCD_3x3.yuv

where the file list.txt should be as in Figure 16:

[2] yuv444p10le : planar YUV 4:4:4, 30bpp, (1 Cr & Cb sample per 1x1 Y samples), little-endian [24]

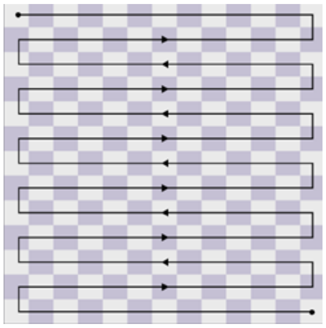


Figure15: Pseudo-temporal sequence scan (from [23]).

```
file 'ABCD_000_000.ppm'
duration 1
file 'ABCD_001_000.ppm'
duration 1
file 'ABCD_002_000.ppm'
duration 1
file 'ABCD_002_001.ppm'
duration 1
file 'ABCD_001_001.ppm'
duration 1
file 'ABCD_000_001.ppm'
duration 1
file 'ABCD_000_002.ppm'
duration 1
file 'ABCD_001_002.ppm'
duration 1
file 'ABCD_002_002.ppm'
duration 1
```

Figure 16: File with serpentine scanning order.

5.2 HEVC Anchor Encoding

This section describes the software packages used to generate the anchor data and metrics.

The x.265 implementation (x265.org) of HEVC, source version 2.3 (available from https://bitbucket.org/multicoreware/x265) should be used. The source has to be compiled with **MAIN10 (10-bit profile) profile support.** Please read the Build Considerations at http://x265.readthedocs.io/en/default/api.html#multi-library-interface. The build instructions for Linux use the shell script make-Makefiles.bash, while the one for Windows uses the make-solutions.bat file. Please note that the build configuration to 10-bit support sets the HIGH_BIT_DEPTH parameter. The HIGH_BIT_DEPTH parameter should be turned ON, during the build process, in order to create a binary file for the requested bit depth (10-bit). This parameter allows 10-bit encoding; however, the compressed version of the 10-bit files, once decoded, appear as if their were scaled so that their dynamic range is 16 bits. Therefore, after decoding the reconstructed images have to be scaled back to the original 10 bits dynamic range. (Note: Check the latest version of the software and make sure it works). This code requires re-compiling for 8-bit dynamic range.

5.2.1 Lenslet and Synthetic Anchors

For HEVC encoding a YUV 4:4:4 file, a two-step encoding process is used, which is based on the following command line arguments:

Step 1:

```
x265 --input ${IN_PATH} --input-depth 10 --input-csp i444 --fps 30 --input-res "${I_W}x${I_H}" --output-depth 10 --profile main444-10 --output /dev/null --pass 1 --stats "stats_log"
```

Step 2:

```
x265 --input ${IN_PATH} --input-depth 10 --input-csp i444 --fps 30 --input-res "${I_W}x${I_H}" --output-depth 10 --profile main444-10 --bitrate ${RATE} --output ${OUT_PATH} --pass 2 --stats "stats_log"
```

To be specified: {RATE} values and I_W x I_H.

The 8-bit datasets (Tarot Cards, Greek and Sideboard) should be converted to 10-bit representation prior to encoding and performance computation.

5.2.2 HDCA Anchors

For HEVC encoding a YUV 4:4:4 file, the following command line arguments should be used:

```
x265 --input ${IN_PATH} --input-depth 10 --input-csp i444 --fps 30 --input-res"${I_W}x${I_H}" --output-depth 10 --profile main444-10 --output ${OUT_PATH} --crf ${CRF}
```

To be specified: I_W x I_H plus CRF.

The 8-bit dataset (Laboratory1) should be converted to 10-bit representation prior to encoding and performance computation.

5.3 HEVC Anchor Decoding

In principle, any complied decoder can be used. For convenience, an example is provided below on how the ffmpeg tool [21] can be used for decoding:

```
ffmpeg -i HDCA_Set2_2K_sub_010_020_444_000250.265 results_directory/ HDCA_Set2_2K_sub_010_020_000250_%03d.ppm
```

Please note that the output of the ffmpeg command is a16-bit PPM of 10-bit PPM reconstructed light fields. To convert from 16-bit to 10-bit, one can use any tool. For convenience, an example is provided below on how the ImageMagick [21] convert tool can be used for converting 16-bit PPM files to 10-bit PPM files:

```
convert HDCA_Set2_2K_sub_010_020_444_000250_0001.ppm –depth 10 HDCA_Set2_2K_sub_010_020_444_000250_0001_10bit.ppm
```

5.4 HEVC Anchor Rate-Distortion (RD) Performance

Annex 1 includes tables with the HEVC RD performance for the target bitrates defined in Table 2, notably their associated PSNR and SSIM values.

6 References

1. *M. Řeřábek and T. Ebrahimi, "New light field image dataset", 8th International Conference on Quality of Multimedia Experience (QoMEX), Lisbon, Portugal, June 2016.*
2. *M. Řeřábek, L. Yuan, L. Authier, T. Ebrahimi, "EPFL Light-Field Image Dataset", 69th JPEG Meeting, Warsaw, Poland, July 2015.*
3. *JPEG Convenor, "JPEG Pleno call for proposals on light field coding", Doc. ISO/IEC JTC1/SC29/WG1/N/74014, JPEG Geneva Meeting, Jan. 2017.*
4. *D. G. Dansereau, "Light field toolbox v0.4", Feb. 2015. [Online]. https://www.mathworks.com/matlabcentral/fileexchange/49683-light-field-toolbox-v0-4. [Accessed 10 01 2017].*
5. *Fraunhofer HHI, "Specifications for High-density Camera Array (HDCA) Data Sets", Doc. ISO/IEC JTC1/SC29/WG1/M75008, 75th JPEG Meeting, Sydney, Australia, March 2017.*
6. *M. Ziegler, R. op het Veld, J. Keinert and F. Zilly, "Acquisition system for dense lightfield of large scenes", 3DTV Conference: The True Vision - Capture, Transmission and Display of 3D Video (3DTV-CON), Copenhagen, Denmark, 2017.*
7. *WG1, "Core experiments for JPEG Pleno", Doc. ISO/IEC JTC1/SC29/WG1/N76047, 76th JPEG Meeting, Torino, Italy, July 2017.*
8. *WG1, "Core Experiments Set #2 for JPEG Pleno", Doc. ISO/IEC JTC1/SC29/WG1/N/77016, 77th JPEG Meeting, Macau, Oct. 2017.*
9. *M. Domański, K. Klimaszewski, M. Kurc, R. Ratajczak, O. Stankiewicz, K. Wegner, "Super-multi-view light-field images from Poznan University of Technology", Doc. ISO/IEC JTC1/SC29/WG11/M36566, Poland, Warsaw, July 2015.*
10. *M. Domański, K. Klimaszewski, M. Kurc, A. Luczak, O.Stankiewicz, K.Wegner, "FTV: Poznam Laboratory - a test light field sequence from Poznan University of Technology", ISO/IEC JTC1/SC29/WG11/M35071, October 2014, Poland, Warsaw*
11. *ITU-T. TIFF 6.0 Specification. https://www.itu.int/itudoc/itu-t/com16/tiff-fx/docs/tiff6.*
12. *The (New) Stanford Light Field Archive, http://lightfield.stanford.edu/*
13. *W^{3}C, PNG (Portable Network Graphics) Specification. https://www.w3.org/TR/PNG/PNG-Structure.html*
14. *HCI 4D Light Field Dataset, http://hci-lightfield.iwr.uni-heidelberg.de/*
15. *PFM Graphic Image File Format. http://netpbm.sourceforge.net/doc/pfm.html*
16. *M. P. Pereira, G. O. Alves, C. L. Pagliari, M. B., da Silva Carvalho, E. A. B. Silva, F. Pereira, "Geometric Space-View Redundancy Descriptor for Light Fields: Predicting the Compression Potential of the JPEG Pleno Light Field Datasets", International Workshop on Multimedia Signal Processing, Luton, UK, October 2017.*
17. *WG1, "JPEG Pleno Call for Proposals – Submission Process Details", Doc. ISO/IEC JTC 1/SC29/WG1N75024 75th Meeting, Sydney, Australia, March 26-31, 2017*

18. *WG1, "Core Experiments Set #3 for JPEG Pleno", Doc. ISO/IEC JTC 1/SC 29/WG1N78010, December 31, 2017*

19. *ITU-R, Recommendation ITU-R BT.709-6: Parameter values for the HDTV standards for production and international programme exchange, June 2015.*

20. *Zhou Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: from error visibility to structural similarity", IEEE Trans. on Image Processing, vol. 13, nº. 4, pp. 600-612, April 2004.*

21. *https://www.ffmpeg.org*

22. *https://www.imagemagick.org/*

23. *WG1, "Overview of anchor generation and subjective and objective evaluations", Doc. ISO/IEC JTC1/SC29/WG1M76070, 76th Meeting, Torino, Italy, July 15-21, 2017*

24. *https://ffmpeg.org/doxygen/2.8/pixfmt_8h_source.html*

25. *G. Bjontegaard, "Calculation of average PSNR differences between RD-curves (VCEG-M33)," in VCEG Meeting (ITU-T SG16 Q.6), Austin, Texas, USA, Apr. 2001*

26. *http://www.mathworks.com/matlabcentral/fileexchange/27798-bjontegaardmetric/content/bjontegaard.m*

27. *M. Levoy and P. Hanrahan. 1996. Light field rendering. In Proceedings of the 23rd annual conference on Computer graphics and interactive techniques (SIGGRAPH '96). ACM, New York, NY, USA, 31-42.*

28. *T. Georgiev and A. Lumsdaine Theory and Methods of Lightfield Photography, Eurographics 2010, Norrkoping, Sweden, May 4, 2010*

ANNEX 1

HEVC ANCHOR RATE-DISTORTION TABLES AND PLOTS

Bikes

Target bitrates (bpp)	0.001	0.005	0.02	0.1	0.75
Obtained bitrates (bpp)		**0.0050**	**0.0205**	**0.1016**	**0.7546**
PSNR-Y (dB)		25.14	32.19	37.29	42.55
PSNR-Cb (dB)		31.93	34.46	37.26	40.51
PSNR-Cr (dB)		31.86	34.31	36.76	39.95
PSNR-YCbCr (dB)		26.83	32.74	37.22	41.97
SSIM		0.677	0.888	0.952	0.981

Danger_de_Mort

Target bitrates (bpp)	0.001	0.005	0.02	0.1	0.75
Obtained bitrates (bpp)		**0.0050**	**0.0204**	**0.0999**	**0.7589**
PSNR-Y (dB)		25.16	30.57	35.33	40.94
PSNR-Cb (dB)		28.79	31.49	34.58	38.17
PSNR-Cr (dB)		31.23	33.04	35.64	38.96
PSNR-YCbCr (dB)		26.37	30.97	35.27	40.35
SSIM		0.635	0.857	0.943	0.978

Stone_Pillars_Outside

Target bitrates (bpp)	0.001	0.005	0.02	0.1	0.75
Obtained bitrates (bpp)		**0.0050**	**0.0200**	**0.1005**	**0.7510**
PSNR-Y (dB)		28.41	33.76	37.55	42.39
PSNR-Cb (dB)		34.86	36.99	38.20	40.54
PSNR-Cr (dB)		35.30	36.77	37.82	40.49
PSNR-YCbCr (dB)		30.79	34.54	37.67	41.92
SSIM		0.707	0.888	0.946	0.976

Fountain&Vincent2

Target bitrates (bpp)	0.001	0.005	0.02	0.1	0.75
Obtained bitrates (bpp)		**0.0050**	**0.0201**	**0.1005**	**0.7585**
PSNR-Y (dB)		26.03	32.84	37.29	42.09
PSNR-Cb (dB)		31.88	33.77	36.15	39.53
PSNR-Cr (dB)		32.04	33.71	35.77	39.18
PSNR-YCbCr (dB)		27.11	33.06	36.96	41.40
SSIM		0.755	0.917	0.960	0.981

Tarot_Cards

Target bitrates (bpp)	0.001	0.005	0.02	0.1	0.75
Obtained bitrates (bpp)		**0.0051**	**0.0206**	**0.1005**	**0.7575**
PSNR-Y (dB)		29.95	36.15	39.95	45.62
PSNR-Cb (dB)		34.77	39.97	44.22	47.39
PSNR-Cr (dB)		36.55	41.97	46.20	49.01
PSNR-YCbCr (dB)		31.38	37.35	41.26	46.26
SSIM		0.904	0.968	0.983	0.992

Greek

Target bitrates (bpp)	0.001	0.005	0.02	0.1	0.75
Obtained bitrates (bpp)		**0.0051**	**0.0201**	**0.1000**	**0.7407**
PSNR-Y (dB)		31.59	38.27	42.83	47.86
PSNR-Cb (dB)		40.32	47.80	52.58	55.46
PSNR-Cr (dB)		44.89	51.08	55.68	58.09
PSNR-YCbCr (dB)		34.35	41.06	45.66	50.09
SSIM		0.867	0.946	0.974	0.988

Sideboard

Target bitrates (bpp)	0.001	0.005	0.02	0.1	0.75
Obtained bitrates (bpp)		**0.0051**	**0.0204**	**0.0997**	**0.7521**
PSNR-Y (dB)		23.25	27.70	33.83	42.07
PSNR-Cb (dB)		32.57	34.50	39.07	45.04
PSNR-Cr (dB)		31.01	32.88	38.07	43.97
PSNR-YCbCr (dB)		25.38	29.20	35.01	42.68
SSIM		0.554	0.816	0.948	0.988

Set2 2K sub

Target bitrates (bpp)	0.0005	0.001	0.005	0.01	0.05	0.1
Obtained bitrates (bpp)		**0.0019**	**0.0051**	**0.0104**	**0.0506**	**0.1015**
PSNR-Y (dB)		27.07	32.01	34.80	41.50	44.20
PSNR-Cb (dB)		34.89	37.20	39.57	45.89	48.45
PSNR-Cr (dB)		34.55	36.90	39.39	46.05	48.74
PSNR-YCbCr (dB)		28.98	33.27	35.97	42.62	45.30
SSIM		0.763	0.875	0.920	0.976	0.990

Laboratory1

Target bitrates (bpp)	0.0005	0.001	0.005	0.01	0.05	0.1
Obtained bitrates (bpp)		**0.0027**	**0.0050**	**0.0100**	**0.0504**	**0.1001**
PSNR-Y (dB)		27.93	32.07	35.37	41.29	42.58
PSNR-Cb (dB)		37.24	38.76	40.63	44.43	45.26
PSNR-Cr (dB)		37.26	38.97	40.83	44.27	44.95
PSNR-YCbCr (dB)		30.26	33.76	36.71	42.05	43.21
SSIM		0.832	0.899	0.935	0.970	0.974