INTERNATIONAL ORGANISATION FOR STANDARDISATION
ORGANISATION INTERNATIONALE DE NORMALISATION
ISO/IEC JTC 1/SC 29/WG04
MPEG VIDEO CODING

ISO/IEC JTC 1/SC 29/WG 04 m57753
October 2021, Online

Title: 3D-HEVC in MIV verification tests

Source: Adrian Dziembowski, Dawid Mieloch, Jarosław Samelak
Poznań University of Technology

Abstract

This contribution contains HTM configuration files, which allow 3D-HEVC encoding for MIV content. Three configuration files are attached, for sequences B, P, and J. The document contains also results obtained for these sequences. The conclusion is simple: 3D-HEVC performs well for linear multicamera systems, but it cannot provide reasonable results for other camera arrangements.

1 The pipeline

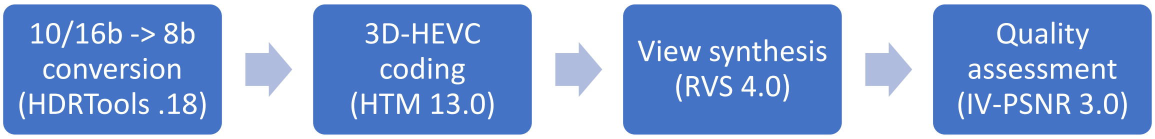

Fig. 1. The pipeline of the experiment

In the first step, input views and depth maps were converted to 8bps 4:2:0 format. All further steps were performed using 8bps data.

To estimate the objective quality, views synthesized using RVS were compared with input views converted to 8bps format.

The HTM software was configured by setting the compilation flag HEVC_EXT to 2 in the TypeDef.h file.

2 HTM configuration file

All the fragments of the configuration file, which should be changed depending on the sequence are shown in the figures below (on the example of SP – Carpark sequence). Presented line numbers may change if another sequence is used.

The list of input files (and reconstructed files) contains all the views and depth maps with defined order: starting from the central view, then neighboring left, neighboring right, next left, next right, etc.

BitstreamFile (line 26) will contain the encoded bitstream.

Each input video will be treated by 3D-HEVC as a separate layer.

```
5  #======== File I/O =====================
6
7  InputFile_0                    : _SEQ/v4_texture_1920x1088_yuv420p.yuv
8  InputFile_1                    : _SEQ/v4_depth_1920x1088_yuv420p.yuv
9  InputFile_2                    : _SEQ/v3_texture_1920x1088_yuv420p.yuv
10 InputFile_3                    : _SEQ/v3_depth_1920x1088_yuv420p.yuv
11 InputFile_4                    : _SEQ/v5_texture_1920x1088_yuv420p.yuv
12 InputFile_5                    : _SEQ/v5_depth_1920x1088_yuv420p.yuv
13 InputFile_6                    : _SEQ/v2_texture_1920x1088_yuv420p.yuv
14 InputFile_7                    : _SEQ/v2_depth_1920x1088_yuv420p.yuv
15 InputFile_8                    : _SEQ/v6_texture_1920x1088_yuv420p.yuv
16 InputFile_9                    : _SEQ/v6_depth_1920x1088_yuv420p.yuv
17 InputFile_10                   : _SEQ/v1_texture_1920x1088_yuv420p.yuv
18 InputFile_11                   : _SEQ/v1_depth_1920x1088_yuv420p.yuv
19 InputFile_12                   : _SEQ/v7_texture_1920x1088_yuv420p.yuv
20 InputFile_13                   : _SEQ/v7_depth_1920x1088_yuv420p.yuv
21 InputFile_14                   : _SEQ/v0_texture_1920x1088_yuv420p.yuv
22 InputFile_15                   : _SEQ/v0_depth_1920x1088_yuv420p.yuv
23 InputFile_16                   : _SEQ/v8_texture_1920x1088_yuv420p.yuv
24 InputFile_17                   : _SEQ/v8_depth_1920x1088_yuv420p.yuv
25
26 BitstreamFile                  : out/PST_QP25.bin      #
27
28 ReconFile_0                    : out/v4_QP25_texture_1920x1088_yuv420p.yuv
29 ReconFile_1                    : out/v4_QP25_depth_1920x1088_yuv420p.yuv
30 ReconFile_2                    : out/v3_QP25_texture_1920x1088_yuv420p.yuv
31 ReconFile_3                    : out/v3_QP25_depth_1920x1088_yuv420p.yuv
32 ReconFile_4                    : out/v5_QP25_texture_1920x1088_yuv420p.yuv
33 ReconFile_5                    : out/v5_QP25_depth_1920x1088_yuv420p.yuv
34 ReconFile_6                    : out/v2_QP25_texture_1920x1088_yuv420p.yuv
35 ReconFile_7                    : out/v2_QP25_depth_1920x1088_yuv420p.yuv
36 ReconFile_8                    : out/v6_QP25_texture_1920x1088_yuv420p.yuv
37 ReconFile_9                    : out/v6_QP25_depth_1920x1088_yuv420p.yuv
38 ReconFile_10                   : out/v1_QP25_texture_1920x1088_yuv420p.yuv
39 ReconFile_11                   : out/v1_QP25_depth_1920x1088_yuv420p.yuv
40 ReconFile_12                   : out/v7_QP25_texture_1920x1088_yuv420p.yuv
41 ReconFile_13                   : out/v7_QP25_depth_1920x1088_yuv420p.yuv
42 ReconFile_14                   : out/v0_QP25_texture_1920x1088_yuv420p.yuv
43 ReconFile_15                   : out/v0_QP25_depth_1920x1088_yuv420p.yuv
44 ReconFile_16                   : out/v8_QP25_texture_1920x1088_yuv420p.yuv
45 ReconFile_17                   : out/v8_QP25_depth_1920x1088_yuv420p.yuv
46
```


In lines 49-51, three sequence-dependent parameters have to be set. In line 52, the number of layers is set. The number of layers is equal to the total number of input videos (textures + depth maps).

Fields ViewOrderIndex and Depth flag should contain as many values, as the number of videos is used.

View Ids in line 62 should be ordered in the same way, as input files in the figure above.

Layer sets (lines 67-75) should contain layer ids, as presented in the figure below. If more views are being encoded, sets 7, 8, and 9 should be expanded.

```
FramesToBeEncoded                : 17
FrameRate                        : 25
SourceWidth                      : 1920
SourceHeight                     : 1088
NumberOfLayers                   : 18      } num of views *2
TargetEncLayerIdList             :

#======== VPS ========================
ScalabilityMask                  : 3
DimensionIdLen                   : 1 4
ViewOrderIndex                   : 0 0 1 1 2 2 3 3 4 4 5 5 6 6 7 7 8 8 } 8-view case, expand if needed
DepthFlag                        : 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 }
LayerIdInNuh                     : 0
SplittingFlag                    : 0
ViewId                           : 4 3 5 2 6 1 7 0 8 } the same order, as in input files
OutputVpsInfo                    : 0
#======== VPS/ Layer sets ==============
VpsNumLayerSets                  : 10
LayerIdsInSet_0                  : 0
LayerIdsInSet_1                  : 0 1      C
LayerIdsInSet_2                  : 0 1 2 3 C, 1L
LayerIdsInSet_3                  : 0 1 4 5 C, 1R
LayerIdsInSet_4                  : 0 1 2 3 4 5  C, 1L, 1R
LayerIdsInSet_5                  : 0 1 6 7
LayerIdsInSet_6                  : 2 3 6 7
LayerIdsInSet_7                  : 0 1 2 3 6 7 10 11 14 15  C, 1L, 2L, 3L, 4L, ...
LayerIdsInSet_8                  : 0 1 4 5 8 9 12 13 16 17  C, 1R, 2R, 3R, 4R, ...
LayerIdsInSet_9                  : 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 ALL
LayerIdsInSet_10                 :
LayerIdsInSet_11                 :
LayerIdsInSet_12                 :
LayerIdsInSet_13                 :
LayerIdsInSet_14                 :
LayerIdsInSet_15                 :
LayerIdsInSet_16                 :
LayerIdsInSet_17                 :
```

The number of fields “DirectRefLayers_*” and “DependencyTypes_*” is the same, as the number of layers (num of views * 2). DependencyTypes always contain two 2’s, the value of “DirectRefLayers_x” is defined as (x-4) and (x-3) for even x, and (x-4) and (x-1) for odd x.

Base view camera numbers contain camera numbers in the same order, as for input views.

```
#======== VPS / Dependencies ================
DirectRefLayers_1            : 0          # Indices in VPS of direct reference layers
DirectRefLayers_2            : 0 1        # Indices in VPS of direct reference layers
DirectRefLayers_3            : 1 2        # Indices in VPS of direct reference layers
DirectRefLayers_4            : 0 1        # Indices in VPS of direct reference layers
DirectRefLayers_5            : 1 4        # Indices in VPS of direct reference layers
DirectRefLayers_6            : 2 3        # Indices in VPS of direct reference layers
DirectRefLayers_7            : 3 6        # Indices in VPS of direct reference layers
DirectRefLayers_8            : 4 5        # Indices in VPS of direct reference layers
DirectRefLayers_9            : 5 8        # Indices in VPS of direct reference layers   } num of views *2
DirectRefLayers_10           : 6 7        # Indices in VPS of direct reference layers
DirectRefLayers_11           : 7 10        # Indices in VPS of direct reference layers
DirectRefLayers_12           : 8 9        # Indices in VPS of direct reference layers
DirectRefLayers_13           : 9 12        # Indices in VPS of direct reference layers
DirectRefLayers_14           : 10 11        # Indices in VPS of direct reference layers
DirectRefLayers_15           : 11 14        # Indices in VPS of direct reference layers
DirectRefLayers_16           : 12 13        # Indices in VPS of direct reference layers
DirectRefLayers_17           : 13 16        # Indices in VPS of direct reference layers

DependencyTypes_1            : 2          # Dependency types of direct reference layers, 0: Sample 1: Motion 2: Sample+Motion
DependencyTypes_2            : 2 2        # Dependency types of direct reference layers, 0: Sample 1: Motion 2: Sample+Motion
DependencyTypes_3            : 2 2        # Dependency types of direct reference layers, 0: Sample 1: Motion 2: Sample+Motion
DependencyTypes_4            : 2 2        # Dependency types of direct reference layers, 0: Sample 1: Motion 2: Sample+Motion
DependencyTypes_5            : 2 2        # Dependency types of direct reference layers, 0: Sample 1: Motion 2: Sample+Motion
DependencyTypes_6            : 2 2        # Dependency types of direct reference layers, 0: Sample 1: Motion 2: Sample+Motion
DependencyTypes_7            : 2 2        # Dependency types of direct reference layers, 0: Sample 1: Motion 2: Sample+Motion
DependencyTypes_8            : 2 2        # Dependency types of direct reference layers, 0: Sample 1: Motion 2: Sample+Motion   } here too
DependencyTypes_9            : 2 2        # Dependency types of direct reference layers, 0: Sample 1: Motion 2: Sample+Motion
DependencyTypes_10            : 2 2        # Dependency types of direct reference layers, 0: Sample 1: Motion 2: Sample+Motion
DependencyTypes_11            : 2 2        # Dependency types of direct reference layers, 0: Sample 1: Motion 2: Sample+Motion
DependencyTypes_12            : 2 2        # Dependency types of direct reference layers, 0: Sample 1: Motion 2: Sample+Motion
DependencyTypes_13            : 2 2        # Dependency types of direct reference layers, 0: Sample 1: Motion 2: Sample+Motion
DependencyTypes_14            : 2 2        # Dependency types of direct reference layers, 0: Sample 1: Motion 2: Sample+Motion
DependencyTypes_15            : 2 2        # Dependency types of direct reference layers, 0: Sample 1: Motion 2: Sample+Motion
DependencyTypes_16            : 2 2        # Dependency types of direct reference layers, 0: Sample 1: Motion 2: Sample+Motion
DependencyTypes_17            : 2 2        # Dependency types of direct reference layers, 0: Sample 1: Motion 2: Sample+Motion

#========== Camera parameters ==========     the same order, as in input files
BaseViewCameraNumbers        : 4 3 5 2 6 1 7 0 8   #   camera numbers of coded views ( in coding order per view )
CameraParameterFile          : PST.cfg    #    camera parameter file
CodedCamParsPrecision        : 5          # precision used for coding of camera parameters (in units of 2^(-x) luma samples)
GlobalLabel                  : 0      #
GlobalLabelBaseView          : 0      #
```

Camera parameter file links to another configuration file, presented in the next section.

For each layer, it is necessary to set coding parameters. These parameters do not change between layers, so the entire 9-line field may be copied if necessary (only the field name should be changed, as it contains the id of the layer).

```
FrameI_114:  B   0   3   0.442    0   0   0   1   0                      0                          1   0   0   -1
Frame1_114:  B   8   4   0.442    0   0   0   4   4   -8 -10 -12 -16     0                          1   0   1   -1
Frame2_114:  B   4   5   0.3536   0   0   0   3   3   -4 -6  4           1    4   5   1 1 0 0 1     1   0   1   -1
Frame3_114:  B   2   6   0.3536   0   0   0   3   4   -2 -4  2 6         1    2   4   1 1 1 1       1   0   1   -1
Frame4_114:  B   1   7   0.68     0   0   0   3   4   -1  1  3 7         1    1   5   1 0 1 1 1     1   0   1   -1
Frame5_114:  B   3   7   0.68     0   0   0   3   4   -1 -3  1 5         1   -2   5   1 1 1 1 0     1   0   2   -1
Frame6_114:  B   6   6   0.3536   0   0   0   3   4   -2 -4 -6 2         1   -3   5   1 1 1 1 0     1   0   2   -1
Frame7_114:  B   5   7   0.68     0   0   0   3   4   -1 -5  1 3         1    1   5   1 0 1 1 1     1   0   2   -1
Frame8_114:  B   7   7   0.68     0   0   0   3   4   -1 -3 -7 1         1   -2   5   1 1 1 1 0     1   0   2   -1

FrameI_115:  B   0   3   0.442    0   0   0   1   0                      0                          1   0   0   -1
Frame1_115:  B   8   4   0.442    0   0   0   4   4   -8 -10 -12 -16     0                          1   0   1   -1
Frame2_115:  B   4   5   0.3536   0   0   0   3   3   -4 -6  4           1    4   5   1 1 0 0 1     1   0   1   -1
Frame3_115:  B   2   6   0.3536   0   0   0   3   4   -2 -4  2 6         1    2   4   1 1 1 1       1   0   1   -1
Frame4_115:  B   1   7   0.68     0   0   0   3   4   -1  1  3 7         1    1   5   1 0 1 1 1     1   0   1   -1
Frame5_115:  B   3   7   0.68     0   0   0   3   4   -1 -3  1 5         1   -2   5   1 1 1 1 0     1   0   2   -1
Frame6_115:  B   6   6   0.3536   0   0   0   3   4   -2 -4 -6 2         1   -3   5   1 1 1 1 0     1   0   2   -1
Frame7_115:  B   5   7   0.68     0   0   0   3   4   -1 -5  1 3         1    1   5   1 0 1 1 1     1   0   2   -1
Frame8_115:  B   7   7   0.68     0   0   0   3   4   -1 -3 -7 1         1   -2   5   1 1 1 1 0     1   0   2   -1

FrameI_116:  B   0   3   0.442    0   0   0   1   0                      0                          1   0   0   -1
Frame1_116:  B   8   4   0.442    0   0   0   4   4   -8 -10 -12 -16     0                          1   0   1   -1
Frame2_116:  B   4   5   0.3536   0   0   0   3   3   -4 -6  4           1    4   5   1 1 0 0 1     1   0   1   -1
Frame3_116:  B   2   6   0.3536   0   0   0   3   4   -2 -4  2 6         1    2   4   1 1 1 1       1   0   1   -1
Frame4_116:  B   1   7   0.68     0   0   0   3   4   -1  1  3 7         1    1   5   1 0 1 1 1     1   0   1   -1
Frame5_116:  B   3   7   0.68     0   0   0   3   4   -1 -3  1 5         1   -2   5   1 1 1 1 0     1   0   2   -1
Frame6_116:  B   6   6   0.3536   0   0   0   3   4   -2 -4 -6 2         1   -3   5   1 1 1 1 0     1   0   2   -1
Frame7_116:  B   5   7   0.68     0   0   0   3   4   -1 -5  1 3         1    1   5   1 0 1 1 1     1   0   2   -1
Frame8_116:  B   7   7   0.68     0   0   0   3   4   -1 -3 -7 1         1   -2   5   1 1 1 1 0     1   0   2   -1

FrameI_117:  B   0   3   0.442    0   0   0   1   0                      0                          1   0   0   -1
Frame1_117:  B   8   4   0.442    0   0   0   4   4   -8 -10 -12 -16     0                          1   0   1   -1
Frame2_117:  B   4   5   0.3536   0   0   0   3   3   -4 -6  4           1    4   5   1 1 0 0 1     1   0   1   -1
Frame3_117:  B   2   6   0.3536   0   0   0   3   4   -2 -4  2 6         1    2   4   1 1 1 1       1   0   1   -1
Frame4_117:  B   1   7   0.68     0   0   0   3   4   -1  1  3 7         1    1   5   1 0 1 1 1     1   0   1   -1
Frame5_117:  B   3   7   0.68     0   0   0   3   4   -1 -3  1 5         1   -2   5   1 1 1 1 0     1   0   2   -1
Frame6_117:  B   6   6   0.3536   0   0   0   3   4   -2 -4 -6 2         1   -3   5   1 1 1 1 0     1   0   2   -1
Frame7_117:  B   5   7   0.68     0   0   0   3   4   -1 -5  1 3         1    1   5   1 0 1 1 1     1   0   2   -1
Frame8_117:  B   7   7   0.68     0   0   0   3   4   -1 -3 -7 1         1   -2   5   1 1 1 1 0     1   0   2   -1
```

such a set of values (9 lines) should be set for each layer (num of views * 2); the same values for each

The last parameter, which should be changed in the configuration file is the QP (line 394). In our tests, we have used 5 values:

- QP25: 25 30 25 30 25 30,
- QP30: 30 35 30 35 30 35,
- QP35: 35 40 35 40 35 40,
- QP40: 40 45 40 45 40 45,
- QP45: 45 50 45 50 45 50.

```
393  #======== Quantization =============
394  QP                          : 25 30 25 30 25 30       # QP ( mc )
395  MaxDeltaQP                  : 0             # CU-based multi-QP optimization
396  MaxCuDQPDepth               : 0             # Max depth of a minimum CuDQP for sub-LCU-level delta QP
397  DeltaQpRD                   : 0             # Slice-based multi-QP optimization
398  RDOQ                        : 1             # RDOQ
399  RDOQTS                      : 1             # RDOQ for transform skip
```

3 Camera parameter file

The HTM software requires a specific format of camera parameters. Parameters of each view are set in a single line.

Consecutive columns contain:

1. camera id,
2. start frame,
3. number of frames,
4. focal length (only horizontal one, in pixels),
5. camera position (along the horizontal axis only),
6. principal point of the camera matrix (only horizontal one),
7. ZNear,
8. ZFar.

PST.cfg

```
0    0    17    1732.875727 1.2744186    943.231169    3.45064 276.0511
1    0    17    1732.875727 1.1151163    943.231169    3.45064 276.0511
2    0    17    1732.875727 0.955814     943.231169    3.45064 276.0511
3    0    17    1732.875727 0.7965116    943.231169    3.45064 276.0511
4    0    17    1732.875727 0.6372093    943.231169    3.45064 276.0511
5    0    17    1732.875727 0.4779070    943.231169    3.45064 276.0511
6    0    17    1732.875727 0.3186047    943.231169    3.45064 276.0511
7    0    17    1732.875727 0.1593023    943.231169    3.45064 276.0511
8    0    17    1732.875727 0.0000000    943.231169    3.45064 276.0511
```

As shown, the configuration file does not contain any data regarding camera rotation, position other than horizontal, and some intrinsic parameters.

4 Problems, limitations, and workarounds

The 3D-HEVC software has significant limitations regarding camera arrangement. It assumes, that the cameras are arranged linearly and they are rectified.

To allow the 3D-HEVC to work for non-linear content, all information about camera rotation and non-horizontal translation have to be skipped. However, even such a simplification does not help and the encoding cannot be finished because of the assertion fail:

```
Assertion failed: 0, file ..\..\source\Lib\TAppCommon\TAppComCamPara.cpp, line 377
```

The encoder checks positions of all the cameras. If two or more cameras share the same position, the assertion fails.

As a workaround, the position of some cameras was changed by adding the value 0.00001:

K:\3dhevc\SJ_orig.cfg

```
1     0    17    1920.00 -0.20000    960.00  2.24    7.17
4     0    17    1920.00 -0.80000    960.00  2.24    7.17
7     0    17    1920.00 -0.40000    960.00  2.24    7.17
10    0    17    1920.00  0.00000    960.00  2.24    7.17
14    0    17    1920.00 -0.80000    960.00  2.24    7.17
17    0    17    1920.00 -0.40000    960.00  2.24    7.17
20    0    17    1920.00  0.00000    960.00  2.24    7.17
23    0    17    1920.00 -0.60000    960.00  2.24    7.17
```

K:\3dhevc\SJ_1.cfg

```
1     0    17    1920.00 -0.20000    960.00  2.24    7.17
4     0    17    1920.00 -0.80001    960.00  2.24    7.17
7     0    17    1920.00 -0.40000    960.00  2.24    7.17
10    0    17    1920.00  0.00000    960.00  2.24    7.17
14    0    17    1920.00 -0.80000    960.00  2.24    7.17
17    0    17    1920.00 -0.40001    960.00  2.24    7.17
20    0    17    1920.00  0.00001    960.00  2.24    7.17
23    0    17    1920.00 -0.60000    960.00  2.24    7.17
```

However, another error appeared, revealing another limitation of HTM.

```
ERROR: View numbering must be strictly increasing or decreasing from left to right
```

To omit this error, all used views were renumbered:

K:\3dhevc\SJ_1.cfg

```
1     0    17    1920.00 -0.20000    960.00  2.24    7.17
4     0    17    1920.00 -0.80001    960.00  2.24    7.17
7     0    17    1920.00 -0.40000    960.00  2.24    7.17
10    0    17    1920.00  0.00000    960.00  2.24    7.17
14    0    17    1920.00 -0.80000    960.00  2.24    7.17
17    0    17    1920.00 -0.40001    960.00  2.24    7.17
20    0    17    1920.00  0.00001    960.00  2.24    7.17
23    0    17    1920.00 -0.60000    960.00  2.24    7.17
```

K:\3dhevc\SJ_2.cfg

```
2     0    17    1920.00 -0.20000    960.00  2.24    7.17
7     0    17    1920.00 -0.80001    960.00  2.24    7.17
3     0    17    1920.00 -0.40000    960.00  2.24    7.17
1     0    17    1920.00  0.00000    960.00  2.24    7.17
6     0    17    1920.00 -0.80000    960.00  2.24    7.17
4     0    17    1920.00 -0.40001    960.00  2.24    7.17
0     0    17    1920.00  0.00001    960.00  2.24    7.17
5     0    17    1920.00 -0.60000    960.00  2.24    7.17
```

In the final step, the configuration file with camera parameters was sorted by the "new" view numbers, but this step was not necessary:

```
SJ_final.cfg
1  0  0  17  1920.00   0.00001   960.00  2.24   7.17
2  1  0  17  1920.00   0.00000   960.00  2.24   7.17
3  2  0  17  1920.00  -0.20000   960.00  2.24   7.17
4  3  0  17  1920.00  -0.40000   960.00  2.24   7.17
5  4  0  17  1920.00  -0.40001   960.00  2.24   7.17
6  5  0  17  1920.00  -0.60000   960.00  2.24   7.17
7  6  0  17  1920.00  -0.80000   960.00  2.24   7.17
8  7  0  17  1920.00  -0.80001   960.00  2.24   7.17
```

The third limitation of the 3D-HEVC is the camera type – only the perspective cameras are supported. For ERP sequences, there is no focal length, but this value is required in the configuration file. To allow HTM working for ERP content, we have set the focal length to be equal to the horizontal resolution of the view (for SB it is 2048).

With such modifications, HTM does not fail and the encoding is performed.

5 Results

The entire pipeline with 3D-HEVC coding was tested on 3 sequences: P (linear camera arrangement), J (5x5 planar camera array), and B (spherical arrangement of ERP cameras).

The results were compared with TMIV10 A17 anchor, both for WS-PSNR and IV-PSNR:

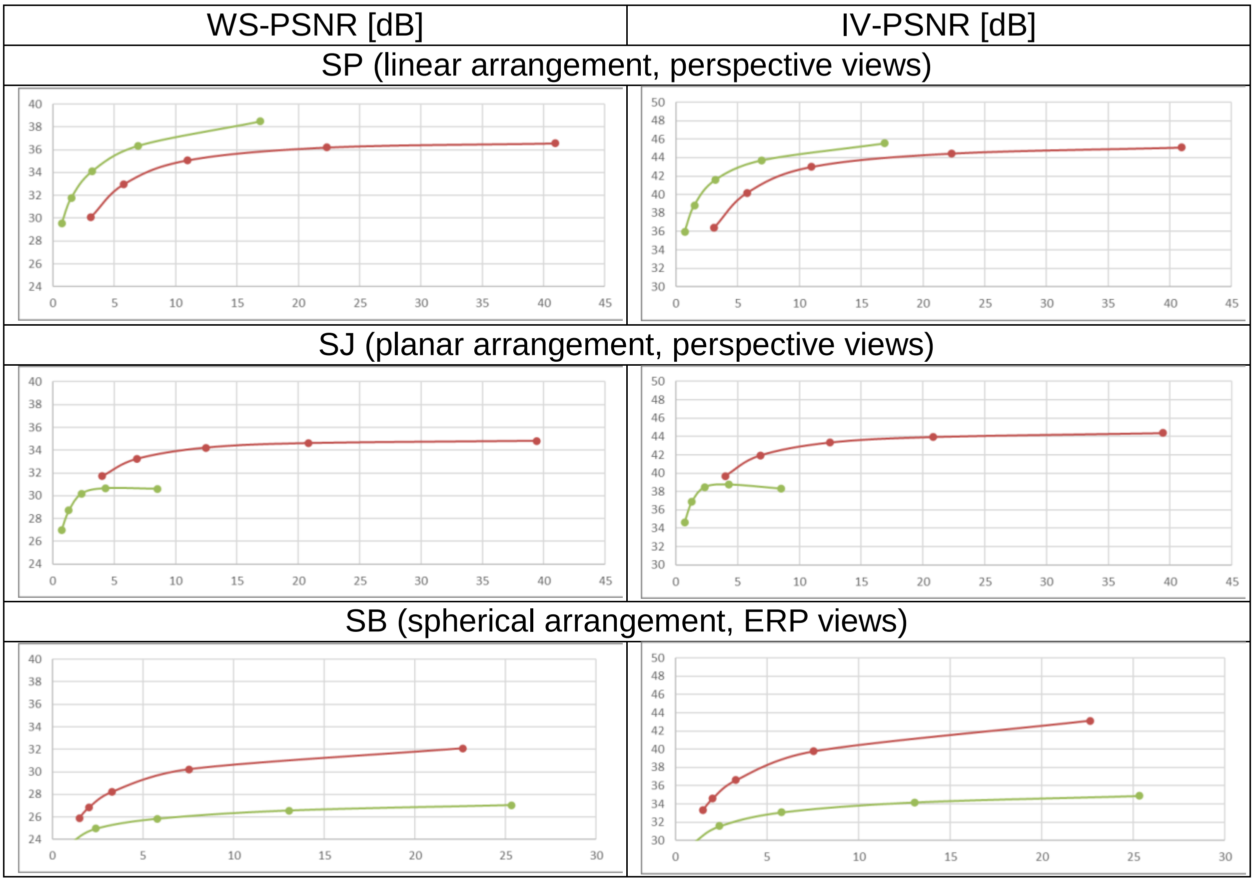

Fig. MIV (red curve) vs. 3D-HEVC (green curve) for three tested sequences.

As presented, 3D-HEVC clearly outperforms MIV for the linear, rectified sequence SP, as that standard was designed for such content.

Fig. MIV (QP2, top) vs. 3D-HEVC (QP25) for SP v4; total bitrate for MIV is higher.

For planar sequence SJ, 3D-HEVC does not allow to achieve reasonable quality, even for lower QP values. However, for drastically low bitrates, 3D-HEVC would probably outperform MIV.

Fig. MIV (QP4) vs. 3D-HEVC (QP25) for SJ v11; total bitrate for 3D-HEVC is higher.

Results for omnidirectional sequence SB indicate, that 3D-HEVC cannot be used for different types of content and MIV outperforms 3D-HEVC in all ways.

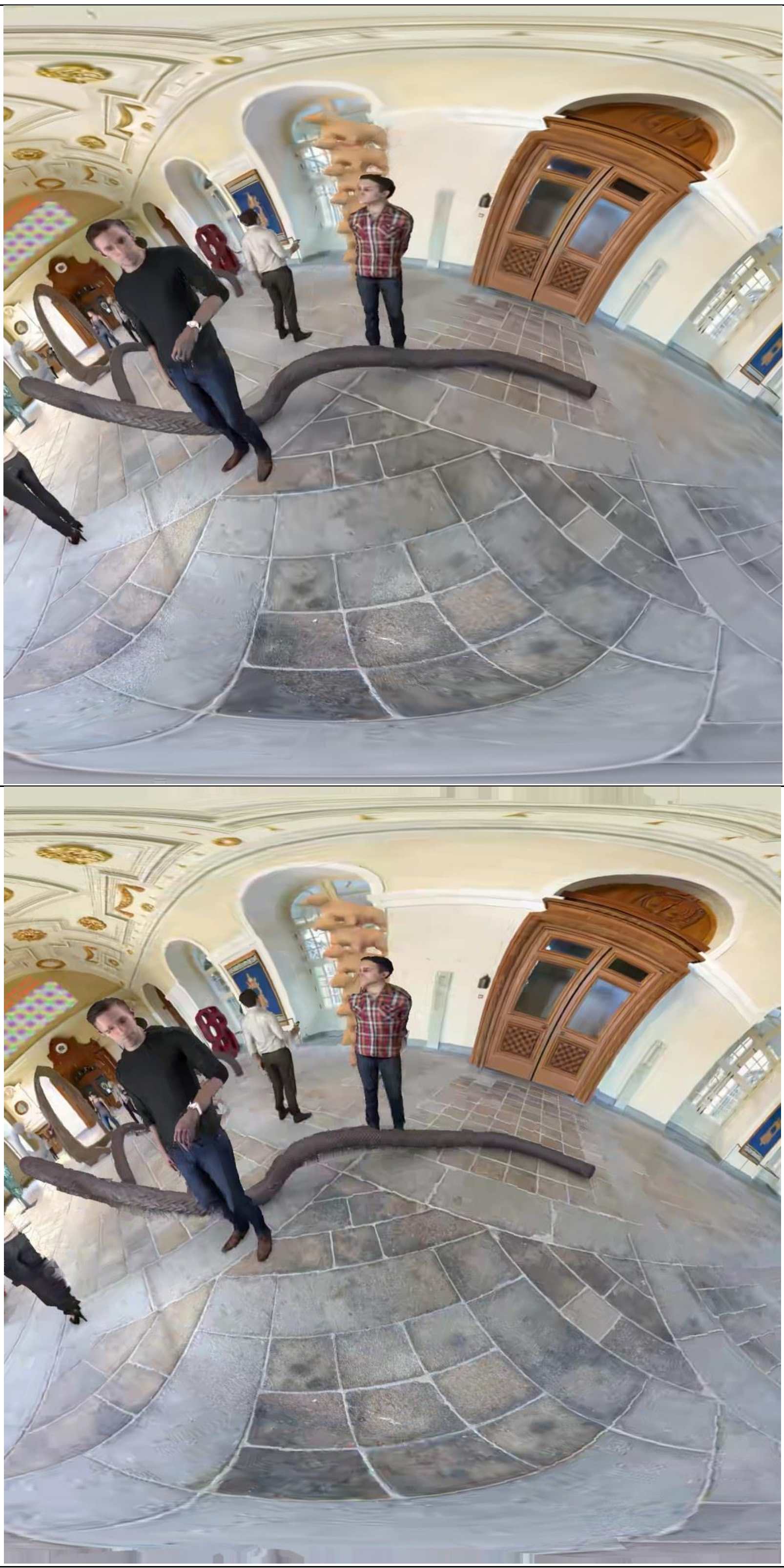

Fig. MIV (QP4) vs. 3D-HEVC (QP40) for SB, v0; similar total bitrate.

6 Recommendations

We recommend:

- using provided configuration files for the crosscheck purposes,
- **not using** the 3D-HEVC encoder in MIV verification tests.

7 Acknowledgement

The research was supported by the Ministry of Science and Higher Education of Republic of Poland.