INTERNATIONAL ORGANISATION FOR STANDARDISATION
ORGANISATION INTERNATIONALE DE NORMALISATION
ISO/IEC JTC 1/SC 29/WG 4
MPEG VIDEO CODING

ISO/IEC JTC 1/SC 29/WG 4 m69525
November 2024, Kemer

Title: [INVR][EE3] Report on EE3: MIV DSDE Anchor Generation on VRroom1D
Source: Sungkyunkwan University (SKKU)
Authors: Jong-Beom Jeong, Yeong-Gyu Kim, Reagan Koo, Do-Hun Kim, Jun-Hyeong Park, Jaeyeol Choi, Eun-Seok Ryu (SKKU)

Abstract

This contribution provides the result of MIV DSDE anchor generation for *VRroom1D* sequence[1], as a part of INVR EE3[2]. Experimental results prove that *VRroom1D* sequence raises challenges for 3D INVR, hence it is appropriate for the test sequences. Also, this contribution provides crosscheck results and some discussion points for successful anchor generation.

1 Test Conditions

Common test conditions (CTC) for radiance field representation and compression[3] describes test conditions for INVR. It is worth noting that the CTC recommends using MIV DSDE test conditions using TMIV18[4]. As discussed in the last meeting, four views from *VRroom1D* were chosen as test views: ***v07, v11, v20,*** **and** ***v24***[5]. To include all 26 training views into atlases, options **maxAtlases** and **maxLumaSampleRate** were set to **7** and **9069547520.0**, respectively. As mentioned in offline discussion, test views were excluded from the camera parameter. Based on the CTC, the start frame was set to 16 and a total of 65 frames were evaluated. The following is TMIV encoding command line:

```
TmivEncoder -n 65 -s M-NC1 -f 16 -c config/ctc/miv_dsde_anchor/G_1_TMIV_encode.json \
-p bitDepthTextureVideo 10 -p maxAtlases 7 -p maxLumaSampleRate 9069547520.0\
-p inputTexturePathFmt /TestSequence/VRroom1D/{3}_texture_{4}x{5}_{6}.yuv \
-p inputSequenceConfigPathFmt config/ctc/sequences/VRroom1D_notestview.json \
-p outputDirectory ./ \
> tmiv_enc_log/G65_M-NC1_tmiv_enc_log.txt
```

After TMIV encoding, VVenC encoding is conducted. Following the MIV CTC, the expert mode executable (vvencFFapp) is used with the configuration file that is attached to the TMIV. The following is an example of VVenC encoding command line:

```
vvencFFapp -c config/ctc/miv_dsde_anchor/G_2_VVenC_encode_tex.cfg \
-i G65/M-NC1/RP0/TMIV_G65_M-NC1_RP0_tex_c00_1920x4640_yuv420p10le.yuv \
-b G65/M-NC1/24/TMIV_G65_M-NC1_24_tex_c00_1920x4640_yuv420p10le.266 -s 1920x4640 -q 24 -f 65 -fr 30 \
> G65/M-NC1/24/TMIV_G65_M-NC1_24_tex_c00_1920x4640_yuv420p10le_enc_log.txt
```

VVdeC decoding can be easily done using '-b' and '-o' options. After VVdeC decoding, TMIV decoding is conducted, which splits atlases into packed views (PVs). Below is an example of TMIV decoding command line:

```
TmivDecoder -n 65 -N 65 -s M-NC1 -r RP0 -c config/ctc/miv_dsde_anchor/G_4_TMIV_decode.json \
```

```
-p inputBitstreamPathFmt G{0}/{1}/RP0/TMIV_G{0}_{1}_RP0.bit \
-p inputTextureVideoFramePathFmt G{0}/{1}/{2}/TMIV_G{0}_{1}_{2}_tex_c{3:02}_{4}x{5}_yuv420p10le.yuv \
-p outputMultiviewTexturePathFmt G{0}/{1}/{2}/TMIV_G{0}_{1}_{2}_tex_pv{3:02}_{4}x{5}_yuv420p10le.yuv \
-p outputSequenceConfigPathFmt G{0}/{1}/{2}/TMIV_G{0}_{1}_{3:04}.json \
> G65/M-NC1/RP0/G65_M-NC1_RP0_tmiv_dec_log.txt
```

In MIV DSDE mode, depth estimation using IVDE is conducted. Default configuration file of IVDE can be used. The following is an example of IVDE command line:

```
IVDE config/M-NC1_estimation_params.json G65/M-NC1/RP0/TMIV_G65_M-NC1_0000.json \
config/TMIV_G65_M-NC1_RP0_filenames.json > G65/M-NC1/RP0/G65_M-NC1_RP0_ivde_log.txt
```

Where *M-NC1_estimation_params.json* is defined following the default configuration:

```
{
    "TotalNumberOfFrames": 65,
    "NumOfThreads": 4,
    "NeighboringSegmentsDepthAnalysis": true,
    "NumberOfZSteps": 256,
    "NumberOfSuperpixels": 100000,
    "MatchNeighbors": 4,
    "MatchThresh": 30,
    "Matcher": "Block",
    "MatchingBlockSize": 1,
    "NumberOfCycles": 2,
    "SuperpixelSegmentationType": "SNIC",
    "SuperpixelColorCoeff": 20,
    "TemporalEnhancement": 2,
    "TemporalEnhancementIFramePeriod": 17,
    "TemporalEnhancementThresh": 0.5,
    "NumberOfCyclesInIFrame": 2,
    "StartFrame": 0,
    "AutomaticDepthRange": true,
    "Point2BlockMatching": true,
    "TexturePrefilteringBlockWidth": 5,
    "BitDepthDepth": 10,
    "DepthColorSpace": "YUV420"
}
```

After depth estimation, TMIV rendering is conducted to synthesize test views. The following command line can be used:

```
TmivRenderer -f 0 -n 65 -N 65 -s M-NC1 -r RP0 -v v07 -c config/ctc/miv_dsde_anchor/G_6_TMIV_render.json \
-p inputSequenceConfigPathFmt G{0}/{1}/{2}/TMIV_G{0}_{1}_{3:04}_autoDepthRange.json \
```

```
-p inputViewportParamsPathFmt config/ctc/sequences/VRroom1D.json \
-p inputTexturePathFmt G{0}/{1}/{2}/TMIV_G{0}_{1}_{2}_tex_{3}_{4}x{5}_{6}.yuv \
-p inputGeometryPathFmt G{0}/{1}/{2}/TMIV_G{0}_{1}_{2}_geo_{3}_{4}x{5}_{6}.yuv \
-p outputViewportTexturePathFmt G{0}/{1}/{2}/G{0}_{1}_{2}_{4}_tex_{5}x{6}_{7}.yuv \
> G65/M-NC1/RP0/G65_M-NC1_RP0_v07_tmiv_ren_log.txt
```

Where *VRroom1D.json* contains information of test views. For quality assessment, synthesized test views that were represented using YUVs are converted to PNGs, using the following command line:

```
ffmpeg -s 1920x1080 -pix_fmt yuv420p10le -y \
-i G65/M-NC1/RP0/TMIV_G65_M-NC1_RP0_tex_v07_1920x1080_yuv420p10le.yuv \
-vf "scale=in_range=full:out_range=full,select='between(n,0,64)',format=rgb24" -vsync vfr -c:v png \
-compression_level 0 G65/M-NC1/RP0/png/TMIV_G65_M-NC1_RP0_tex_v07_1920x1080_yuv420p10le_%03d.png
```

Where the options were decided in offline discussion. INVR EE3 experts decided to use QMIV[6] to measure RGB-PSNR and RGB-IV-SSIM. For RGB-SSIM, *loss_utils.py* from 3DGS source code provided by INRIA is recommended. To measure RGB-LPIPS, a pre-trained network using alexnet is used. Below is an example of QMIV command line:

```
QMIV -i0 /TestSequence/VRroom1D/png/v07_texture_1920x1080_yuv420p10le_{:03d}.png \
-i1 G65/M-NC1/RP0/png/TMIV_G65_M-NC1_RP0_tex_v07_1920x1080_yuv420p10le_{:03d}.png \
-ff PNG -ps 1920x1080 -ml PSNR,IVSSIM -v 1 -bd 8 -csi RGB -cwa "1:1:1:0" -cws "1:1:1:0" \
> G65/M-NC1/RP0/ TMIV_G65_M-NC1_RP0_tex_v07_1920x1080_yuv420p10le_qmiv_log.txt
```

In summary, this contribution used experimental conditions as shown in Table 1. Ubuntu 22.04 LTS server with GCC/G++ 11.4.0 and CMake 3.25.2 was used, and the software versions were aligned with CTC.

Table 1. Experimental conditions

Tools	**Version**
TMIV	18.0
VVenC	1.7.0
VVdeC	1.6.0
IVDE	8.0
QMIV	v1.0
Ubuntu	22.04 LTS
GCC/G++	11.4.0
CMake	3.25.2

2 Experimental Results

All QP values ranging from 20 to 50 were encoded using VVenC. Among the QPs, four QPs were selected to meet bitrates about 30.0, 12.0, 7.5, and 4.5 Mbps. Therefore, as listed in Table 2, QPs 24, 31, 35, 40 were selected, to distinguish between rate points.

Table 2. Selected QPs and bitrates

QP	**24**	**31**	**35**	**40**
Bitrate (Mbps)	29.46	11.56	7.54	4.56
Encoding times (s)	13162	5919	4155	2708

Figure 1 shows the RD-curves of *VRroom1D*. The red dotted line represents the results for RP0. As presented in the curves, PSNR, SSIM, LPIPS, and IV-SSIM all improve as the bitrate increases. The values at the lowest and highest bitrates show minimal variation across all four curves, with the PSNR curve exhibiting a difference of only up to 0.11 dB, resulting in a relatively flat curve. One of the reasons is that during the synthesis, there were shifts, which made the objective quality assessment more difficult. Nevertheless, distinct visual differences between different RPs were noticeable in terms of subjective quality.

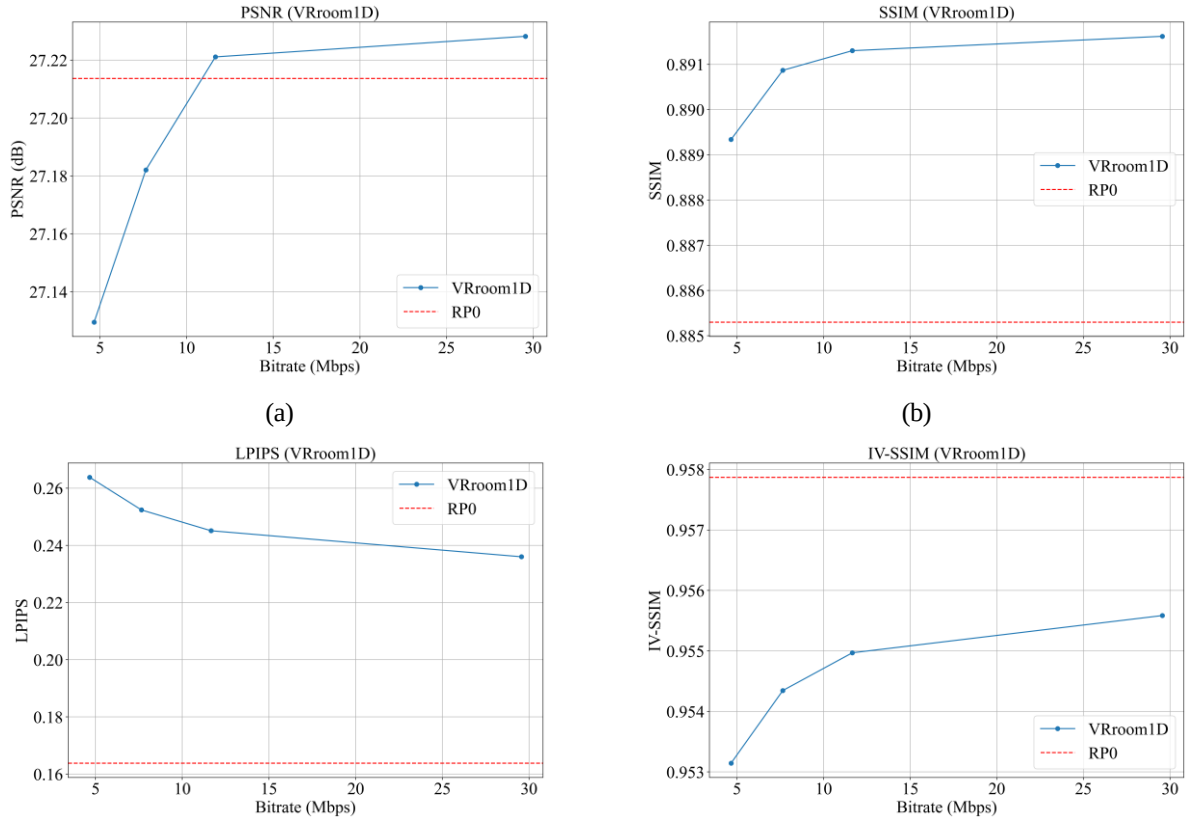


Figure 1. RD-curves of *VRroom1D* sequence, (a) RGB-PSNR↑, (b) RGB-SSIM↑, (c) RGB-LPIPS↓, (d) RGB-IV-SSIM↑. Red dotted lines represent values for RP0 (noncoded).

Figure 2 presents the synthesized test views (v11), including the ground truth and the results for RP0, RP1, RP2, RP3, and RP4. With increasing QPs, the synthesized views exhibit more blurring and more prominent coding artifacts.

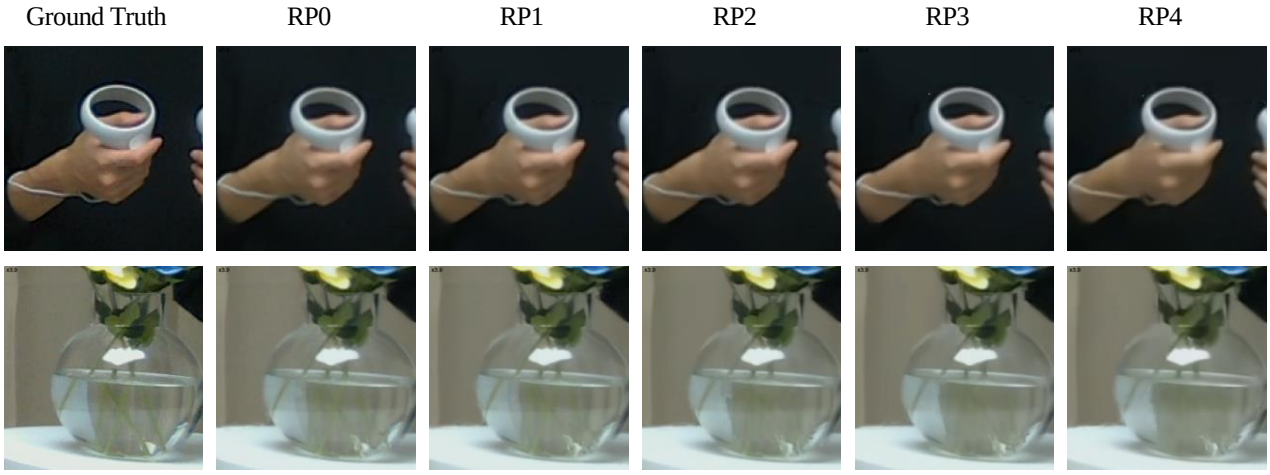
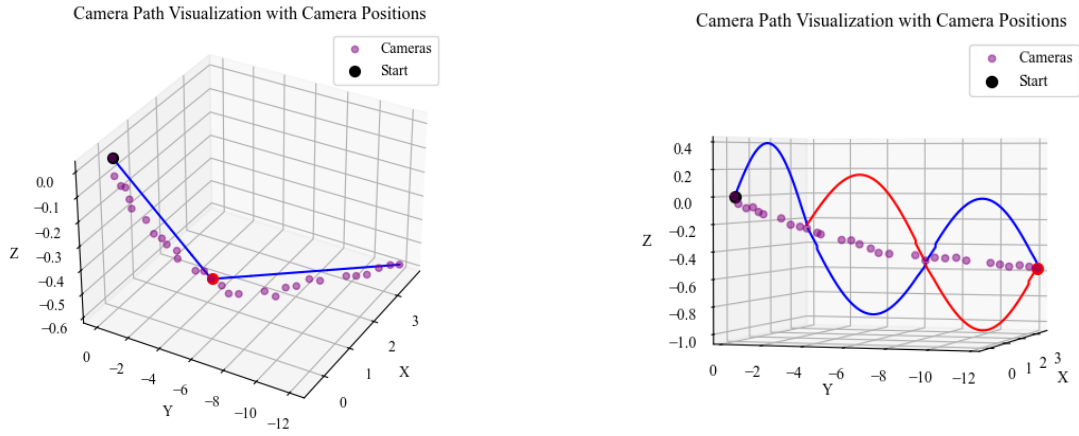


Figure 2. Synthesized test views of v11

This contribution also provides pose traces. Figure 3 shows visualization of two pose traces for *VRroom1D*. Pose trace (a) has a 'V' shape, starting from the left-side position and moving linearly to the middle position, then changing direction to move towards the end position. Pose trace (b) follows a sine wave pattern from the starting point to the far right, then reverses in the form of a negative sine wave at the end. Synthesized pose traces proved that their viewing space did not exceed the viewing range of training views, and they were appropriate for subjective quality assessment.



(a) (b)

Figure 3. Visualization of posetraces. (a) 'V' shape, (b) sine and negative sine

3 Crosscheck

This contribution also provides crosscheck results of *Garage*[7] and *Choreo*[8]. For *Garage*, experimental conditions were verified and aligned. Bitrates were the same, and RGB-PSNR, SSIM, LPIPS values showed very low differences (for RGB-PSNR, no more than 0.01dB).
For *Choreo*, the default 8-bit YUV files were provided. The following script was used to upsample the 8-bit YUVs to 10-bit YUVs:

```
ffmpeg -s 1920x1080 -pix_fmt yuv420p -y -i v0_texture_1920x1080_yuv420p8le.yuv -pix_fmt yuv420p10le \
-vf "scale=in_range=full:out_range=full" v0_texture_1920x1080_yuv420p10le.yuv
```

0.32% of bitrate differences were observed. Crosscheck results showed 0.39dB increase of RGB-PSNR and 0.02 decrease of SSIM. Next section introduces discussion points to minimize these differences.

4 Discussions

To successfully complete anchor generation and crosscheck, the following discussion points are worth discussing.

- **Bitrate measurement.** In MIV activities, adding MIV bitstream bitrates to the atlas bitstream bitrates is recommended. Although the MIV bitstream bitrate is typically under 10 Kbps in MIV DSDE profile, for precise measurement, this should be added.
- **Source YUV file generation.** Generally, 10-bit texture YUVs are given to the TMIV encoder. For *KITTI-360* and *Choreo*, 10-bit YUVs need to be generated; therefore, ffmpeg commands should be the same, including scale option.
- **Quality assessment.** From the previous meeting cycle, QMIV was decided as a tool for RGB-PSNR and IV-SSIM. To measure SSIM, the CTC[3] recommends using 3DGS code provided by INRIA. However, this requires some implementation. Unless the SSIM measurement code is provided, using QMIV can be another solution. Also, when generating png files from YUVs

using ffmpeg, the ‘scale’ option is important. In the last meeting cycle, it was reported that ffmpeg generally uses limited range [16, 235]. To use the full range, the following option can be used: *scale=in_range=full:out_range=full*. Table 3 shows quality differences depending on ‘scale’ option. Generally, not using this option shows higher PSNRs when splitting YUVs into png files. However, using the full range will be more precise experiment.

- Table 3. Quality differences for using ‘scale’ option in ffmpeg to generate png

	W/O Scale				W/ Scale			
	PSNR	SSIM	LPIPS	IV-SSIM	PSNR	SSIM	LPIPS	IV-SSIM
RP0	25.53	0.8662	0.1585	0.9391	27.21	0.8852	0.1638	0.9578
RP1	25.55	0.8750	0.2293	0.9372	27.22	0.8916	0.2360	0.9555
RP2	25.55	0.8750	0.2372	0.9366	27.22	0.8913	0.2451	0.9549
RP3	25.52	0.8747	0.2437	0.9361	27.18	0.8908	0.2523	0.9543
RP4	25.49	0.8733	0.2543	0.9350	27.12	0.889	0.2637	0.9531
Average	**25.52**	0.8728	**0.2246**	0.9368	27.19	**0.8896**	0.2322	**0.9551**

- **IVDE configuration.** Using default configuration of IVDE (*CTC_cfg/estimation_params.json* in IVDE) is recommended. As explained in Section 3.4 in [9], IVDE utilizes parallelization using threads. Therefore, changing *NumOfThreads* can cause different results, and increasing this value does not always decrease the running time, because of thread merging overhead. Regarding bitdepth, the MIV DSDE profile recommends 10-bit depth. Table 4 shows quality differences for 10- and 16-bit depths of *VRroom1D*. Overall, 10-bit depths show better performance; therefore, 10-bit depth is recommended.

Table 4. Quality differences for different bitdepth in IVDE for *VRroom1D*

	10-bit Depth				16-bit Depth			
	PSNR	SSIM	LPIPS	IV-SSIM	PSNR	SSIM	LPIPS	IV-SSIM
RP0	27.21	0.8852	0.1638	0.9578	27.18	0.8848	0.1633	0.9577
RP1	27.22	0.8916	0.2360	0.9555	27.21	0.8911	0.2346	0.9554
RP2	27.22	0.8913	0.2451	0.9549	27.18	0.8908	0.2442	0.9548
RP3	27.18	0.8908	0.2523	0.9543	27.11	0.8900	0.2518	0.9540
RP4	27.12	0.889	0.2637	0.9531	27.07	0.8889	0.2630	0.9527
Average	**27.19**	**0.8896**	0.2322	**0.9551**	27.15	0.8891	**0.2314**	0.9549

5 Conclusion

This contribution outlines the results of MIV DSDE anchor generation for the *VRroom1D* sequence, based on discussions from previous INVR meetings. Four QP values were chosen, considering both bitrate and visual distinguishability, and quality assessments were conducted using PSNR, SSIM, LPIPS, and IV-SSIM metrics. The experimental findings revealed clear visual differences between RPs for *VRroom1D*, suggesting that this content is suitable for INVR experiments. Furthermore, two posetraces were included to evaluate performance from varying perspectives. Crosscheck was successful for *Garage*, and there needs more efforts for *Choreo*. For successful anchor generation, forcing ffmpeg to use full range when splitting YUVs to png files is recommended, for more precise experiment. Also, test conditions should be strictly unified, and it was observed that generating 10-bit depth using IVDE showed better results than 16-bit depth.

6 References

[1] J. Choi, Y. Ryu, Y. Choi, J. -B. Jeong, J. -H. Park, I. Yang, E. -S. Ryu, "[INVR]EE2.1-Related: Report with New Natural INVR Video Contents: SKKU_VRroom", Standard ISO/IEC JTC1/SC29/WG4, input document m64721, October 2023, Hannover.

[2] "Description of exploration experiments on neural representation and compression of video", Standard ISO/IEC JTC1/SC29/WG4, MPEG/N00559, July 2024, Sapporo.

[3] "Common test conditions on radiance field representation and compression", Standard ISO/IEC JTC1/SC29/WG4, MPEG/N00561, July 2024, Sapporo.

[4] A. Dziembowski, B. Kroon, J. Jung, "Common test conditions for MPEG immersive video", Standard ISO/IEC JTC1/SC29/WG4, MPEG/N00406, October 2023, Hannover.

[5] J. -B. Jeong, J. -H. Park, J. Choi, Y. G. Kim, E. -S. Ryu, "Report on EE3: Thoughts on MIV DSDE Anchor Generation", input document m68240, July 2024, Sapporo.

[6] "Software manual of QMIV", Standard ISO/IEC JTC1/SC29/WG4, MPEG/N00535, July 2024, Sapporo.

[7] G. Bang, J. Lee, H. Lee, S. Kim, S. -J. Bae, J. Do, J. W. Kang, "[INVR] EE2.1 report with New INVR Video content", Standard ISO/IEC JTC1/SC29/WG4, input document m64394, July 2023, Geneva.

[8] A. Dziembowski, D. Mieloch, D. Kloska, J. Stankowski, B. Szydeiko, G. Lee, J. Y. Jeong, "[INVR] "Choreo" natural content for INVR applications", Standard ISO/IEC JTC1/SC29/WG4, input document m68221, July 2024, Sapporo.

[9] "Manual of Immersive Video Depth Estimation 3", Standard ISO/IEC JTC1/SC29/WG4, MPEG/N00058, January 2021, Online.