

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 m 57560

July 2021, Online

Title: Attribute offset modification for unoccupied areas
Source: Adrian Dziembowski (Poznań University of Technology)

Abstract

The document presents the description of the fix of TMIV9.1, which allows proper attribute offset modification for sequences with invalid pixels (i.e. Trio). The fix changes the behavior of TMIV9.1 for Trio sequence, while for others it has a negligible impact.

1 The fix

Only the *Encoder_completeAccessUnit.cpp* file was changed. Left – fix, right – TMIV9.1.

Occupancy array is internally used, even if *vps.vps_occupancy_video_present_flag* is set to false – **constructVideoFrames()**:

```
438 ..... if (m_config.haveTexture) {  
439 ..... frame.texture.resize(frameWidth, frameHeight);  
440 ..... frame.texture.fillNeutral();  
441 ..... }  
442 .....  
443 ..... if (m_config.haveGeometry) {  
444 ..... frame.depth.resize(frameWidth, frameHeight);  
445 ..... frame.depth.fillZero();  
446 ..... }  
447 .....  
448 ..... if (vps.vps_occupancy_video_present_flag) {  
449 ..... int occFrameWidth = frameWidth;  
450 ..... int occFrameHeight = frameHeight;  
451 .....  
452 ..... const auto &asme = m_params.atlas[k].asps.asps_mv_extension();  
453 .....  
454 ..... if (!asme.asme_embedded_occupancy_enabled_flag) &&  
455 ..... asme.asme_occupancy_scale_enabled_flag) {  
456 ..... occFrameWidth /= asme.asme_occupancy_scale_factor_x_minus1() + 1;  
457 ..... occFrameHeight /= asme.asme_occupancy_scale_factor_y_minus1() + 1;  
458 ..... }  
459 ..... frame.occupancy.resize(Common::align(occFrameWidth, 2), Common::align(occFrameHeight, 2));  
460 ..... const auto unoccupiedLevel = computeOccUnoccupiedLevel();  
461 ..... vps.occupancy_information[0].oi_occupancy_2d_bit_depth_minus1();  
462 ..... frame.occupancy.fillValue(unoccupiedLevel);  
463 ..... } else {  
464 ..... frame.occupancy.resize(frameWidth, frameHeight);  
465 ..... frame.occupancy.fillZero();  
466 ..... }
```

The occupancy for each pixel is set to 1. If its input depth is 0 and the input view has embedded occupancy information, its occupancy is set to 0 – **writePatchInAtlas()**:

```
573 .....  
574 ..... // Depth  
575 ..... if (m_config.haveGeometry) {  
576 ..... auto depth = view.depth.getPlane(0)(pview.y(), pview.x());  
577 .....  
578 ..... // TODO(BK): We need to stop using depth == 0 as a special value. This is bug-prone.  
579 ..... if (depth == 0 && !viewParams.hasOccupancy && !outViewParams.hasOccupancy &&  
580 ..... asme.asme_max_entity_id() == 0) {  
581 ..... depth = 1; // Avoid marking valid depth as invalid  
582 ..... }  
583 .....  
584 ..... if (depth == 0 && !viewParams.hasOccupancy) {  
585 ..... atlas.occupancy.getPlane(0)(yOcc, xOcc) = 0;  
586 ..... }  
587 .....  
588 ..... atlas.depth.getPlane(0)(pAtlas.y(), pAtlas.x()) = depth;  
589 ..... if (depth > 0 && m_params.vps.vps_occupancy_video_present_flag(patchParams.atlasId())) {  
590 ..... const auto occupiedLevel = Common::assertDownCast<uint16_t>(  
591 ..... computeOccUnoccupiedLevel()  
592 ..... m_params.vps.occupancy_information(patchParams.atlasId())  
593 ..... oi_occupancy_2d_bit_depth_minus1()) *  
594 ..... 3);  
595 ..... atlas.occupancy.getPlane(0)(yOcc, xOcc) = occupiedLevel;  
596 ..... }  
597 ..... }
```

Additional change – **scaleGeometryDynamicRange()**:

```
129 .....for (size_t f = 0; f < numOffFrames; f++) {  
130 .....for (auto &geometry : m_transportViews[f][v].depth.getPlane(0)) {  
131 .....    geometry = static_cast<uint16_t>(  
132 .....        (static_cast<double>(geometry) - minDepthMapValuethinGOP) /  
133 .....        (static_cast<double>(maxDepthMapValuethinGOP) - minDepthMapValuethinGOP) * 65535.0);  
134 .....    if (lowDepthQuality) {  
135 .....        geometry /= 2;  
136 .....    }  
137 .....}  
138 .....}
```

In TMIV 9.1, the depth was improperly scaled by adding 0.5 to the input geometry value. This change may slightly impact other sequences, c.f. Section 3.

2 Results

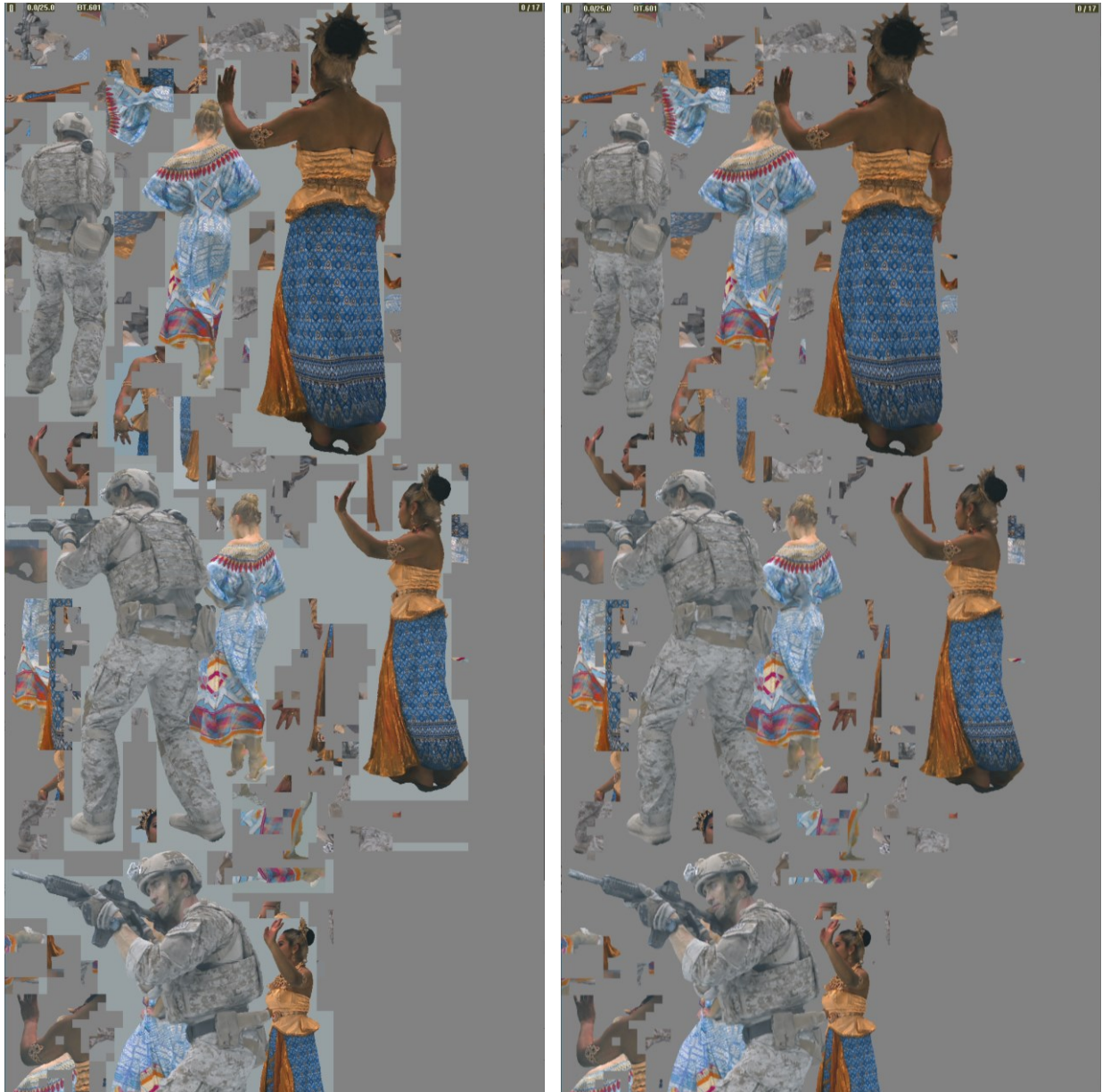


Fig. 1. Fragment of texture atlas 0 from TMIV9.1 (left) and with proposed fix (right).

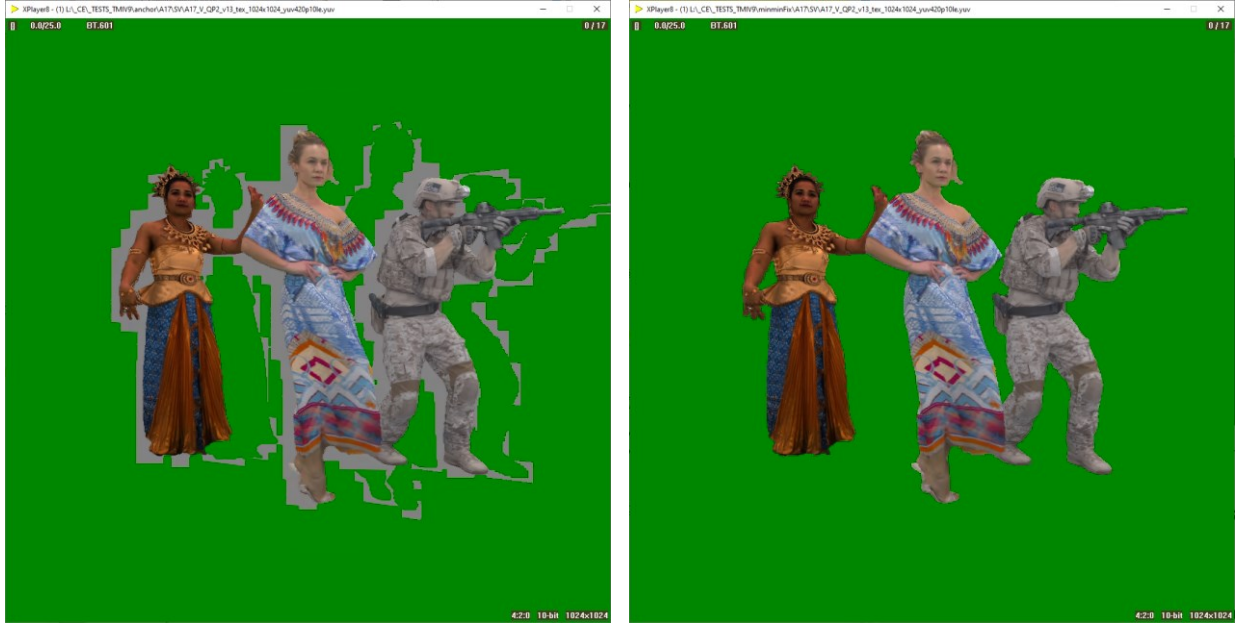


Fig. 2. Synthesized input view from TMIV9.1 (left) and with proposed fix (right); QP2, v13.

The proposed fix was tested under A17 configuration, for all mandatory content and Trio sequence.

Mandatory content - Proposal vs. Low/High-bitrate Anchors						Runtime ratio (%)		
Sequence		High-BR BD rate Y-PSNR	Low-BR BD rate Y-PSNR	Max delta Y-PSNR	High-BR BD rate IV-PSNR	Low-BR BD rate IV-PSNR	Pixel rate [%]	Pixel rate [GP/s]
ClassroomVideo	A	-0.1%	-0.1%	0.98	-0.0%	-0.1%	0%	0.00
Museum	B	-0.2%	-0.0%	9.47	-0.1%	-0.1%	63%	0.67
Fan	O	0.9%	0.9%	8.10	0.3%	0.4%	62%	0.67
Kitchen	J	2.3%	0.6%	14.85	1.5%	0.6%	62%	0.67
Painter	D	-0.0%	-0.0%	8.15	-0.0%	-0.0%	63%	0.67
Frog	E	-0.3%	-0.1%	7.37	-0.0%	-0.0%	62%	0.67
Carpark	P	0.1%	-0.0%	7.10	0.0%	-0.0%	0%	0.00
Chess	N	-3.1%	-0.6%	13.20	0.1%	-0.2%	63%	0.67
Group	R	-2.9%	-2.1%	13.04	-0.2%	-0.4%	62%	0.67
MIV		-0.4%	-0.2%	9.14	0.2%	0.0%	49%	0.52

Atlas encoding	Video encoding	Decoding & Rendering
79.5%	69.0%	57.9%
87.4%	83.4%	94.3%
100.6%	87.7%	81.0%
84.9%	62.8%	46.7%
71.8%	56.0%	41.0%
50.8%	54.7%	39.3%
125.4%	89.6%	36.2%
90.4%	60.0%	42.0%
87.0%	64.3%	76.7%
86.4%	69.7%	57.2%

bitrate	V		
%	WS-PSNR		
	Anchor V	Proposal V	Delta
-1.8%	7.22	7.02	-0.20
-2.8%	7.22	7.02	-0.20
-3.4%	7.22	7.02	-0.21
-3.3%	7.23	7.02	-0.21
-3.1%	7.28	7.04	-0.23

bitrate	V		
%	IV-PSNR		
	Anchor V	Proposal V	Delta
-1.8%	7.43	7.23	-0.20
-2.8%	7.44	7.23	-0.21
-3.4%	7.44	7.24	-0.21
-3.3%	7.45	7.25	-0.21
-3.1%	7.49	7.27	-0.22

The objective quality of synthesized views is lower because they do not contain gray artifacts. The green background at the right is more consistent, but less similar to the input grey background.

3 Influence on other content

The proposed fix corrects the geometry scaling equation.

TMIV9.1:

$$G' = \text{int} \left(\frac{G + 0.5 - G_{\min}}{G_{\max} - G_{\min}} \cdot 65535 \right)$$

With fix:

$$G' = \text{int} \left(\frac{G - G_{\min}}{G_{\max} - G_{\min}} \cdot 65535 \right)$$

Let's consider $G_{\min} = 0$ and $G_{\max} = 5000$, and $G = 0$. Without the fix, rescaled $G' = 6$. After the fix, $G' = 0$. Without the fix, zero values were not zero anymore, resulting in improper patch attribute offset calculation.

When the $G = 5000$ (and $G_{\min}$ and $G_{\max}$ were set as above), rescaled $G' = 65541$ without the fix and $G' = 65535$ after the fix. Without the fix, it was higher than the maximum uint16 value, so the overflow occurred (Fig. 3).

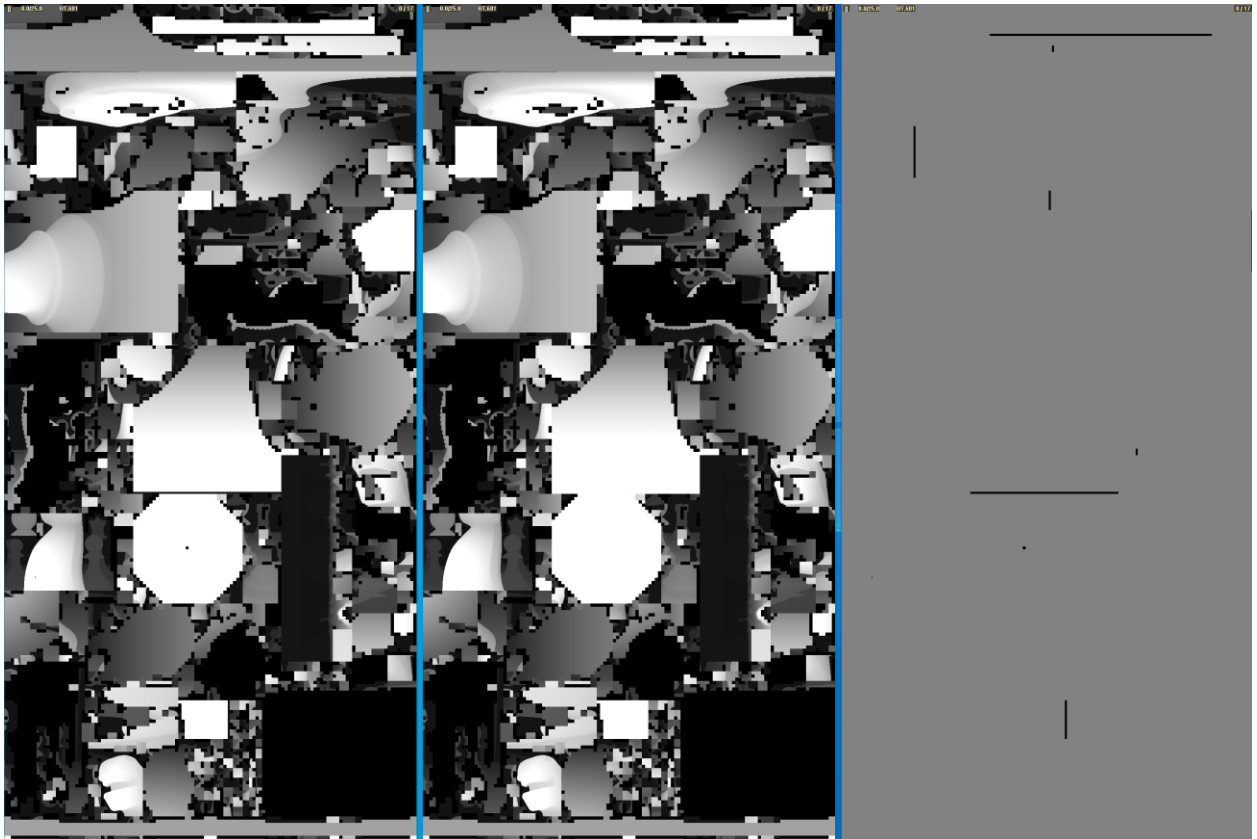


Fig. 3. Second geometry atlas, SN; left: TMIV9.1, middle: with the fix, right: the difference.

4 Recommendation

We recommend to include the proposed fix into TMIV.

Acknowledgement

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