**INTERNATIONAL ORGANISATION FOR STANDARDISATION**

**ORGANISATION INTERNATIONALE DE NORMALISATION**

**ISO/IEC JTC1/SC29/WG11**

**CODING OF MOVING PICTURES AND AUDIO**

**ISO/IEC JTC1/SC29/WG11 MPEG2019/m52415**

**January 2020, Brussels, Belgium**

|  |
| --- |
| **Source Interdigital** |
| **Status Input contribution** |
| **Title CE2.6-related: Non-Lambertian Light Effect Baking** |
| **Author Julien Fleureau, Renaud Doré, Franck Thudor, Thierry Tapie, Gérard Briand, Bertrand Chupeau** |
|  |

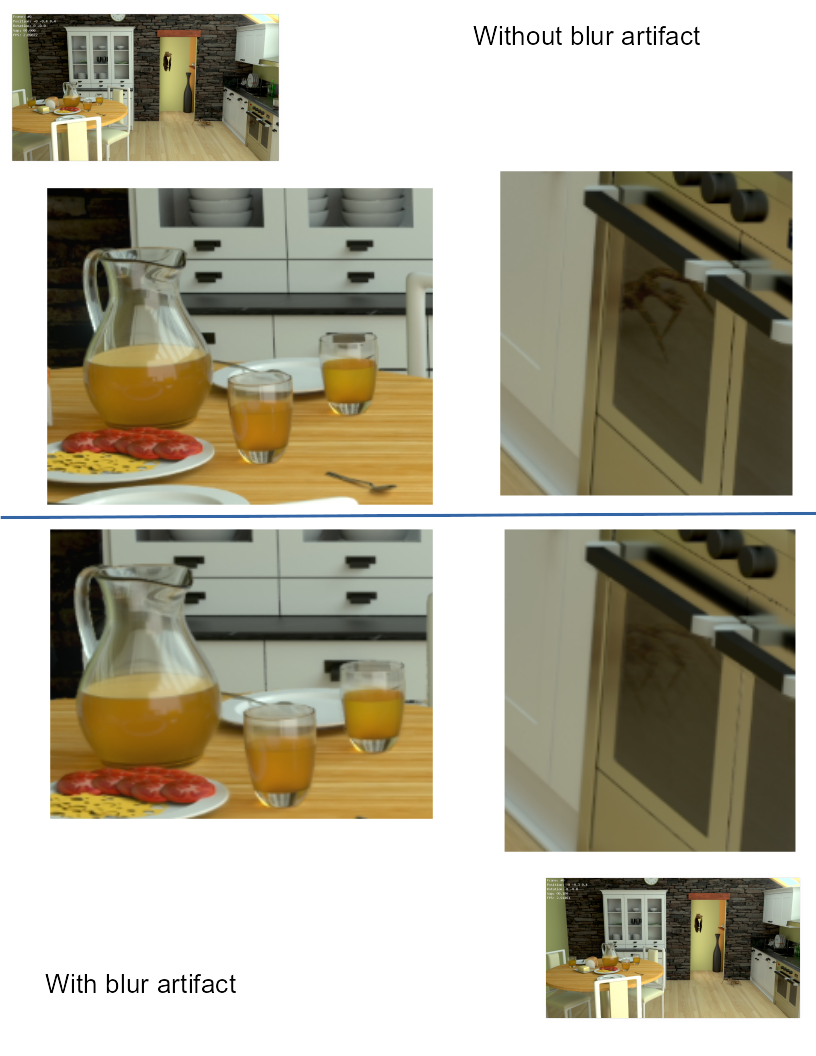
# Context

An important problem faced by the current MIV solution comes from the processing of volumetric scenes containing a lot of reflective, specular or transparency effects (see Figure 1). It is in fact a problem which affects more broadly the view interpolation process of any MVD input. Indeed, recovering a physically true illumination from a MVD input is really complicated because a set of MVD frames does not directly capture sufficient information to recover a proper lightfield. The material properties (Bidirectional Reflectance Distribution Function a.k.a. BRDF) of the objects of the scene as well as the description of the light sources (position, intensity, …) may be only roughly estimated from the set of MVD frames and this process is far to be done real-time.

As a consequence, when synthesizing an interpolated view from several view + depth inputs of a very “non-lambertian” object / scene, the synthesis process is very often too much simple and consists in either a linear combination of the different views or a hard selection of one of these. The problem is even more sensitive with patches where only subparts of the input views are available at the synthesis stage. Very disturbing artifacts may therefore occur including temporal instability, blurring and patchwork effects. The first ones are due to the brutal switch from one reference view to another during a viewport / camera motion for instance, the second one comes from an inappropriate linear combination of views / patches (see Figure 2) and the last one is due to the stitching of patches coming from far views (see Figure 3).



**Figure 1: Example of non-lambertian light effects in OrangeKitchen.**

**Figure 2: Example of blurring artifacts. (left) The jug becomes less translucid, and (right) the spider reflection in the oven is less sharp.**



**Figure 3: Example of patchwork artifacts.**

# Non-Lambertian light effect baking

## Overview

In this contribution, we propose a lightweight strategy to handle this issue. To do so, at the encoding stage one input view (among all the input ones) is selected to convey the viewport dependent light effect and its id is transmitted to the decoder as an extra metadata. At the decoder side, as soon as patches coming from this selected view are “available” for the rendering of the viewport of interest, they are preferentially used regarding the other candidates whatever the view to synthesize position.

The proposed strategy aims at providing to the end-user a visually plausible (and not a physically true) rendering of the scene and take advantage of the limited 6DoF scenario. Indeed, in this latter scenario, the end-user is supposed to experience a limited amount of parallax with “small” motions within a constrained viewing space. In such a scenario it can be empirically shown that a regular viewer is very few sensitive to the variations of specularity / reflection / transparency he could experience when shifting his head position. However, he is very sensitive to the consistency of these light effects over a synthesized view, spatially and temporally.

## Encoding stage

Considering this latter comment, instead of adaptively selecting or weighting / blending patches from different source views at the synthesis stage which may cause artifacts, we suggest, at the encoding stage, to select one unique source view that should be considered at the decoding stage to convey the lighting information. The patches associated to this view will be systematically selected at the decoding stage as the preferred rendering color as soon as this view is covering the current viewport.

Typically, this view should be selected among the views which better cover the volumetric scene in terms of field of view from the supposed average position of the end-user within the viewing space. Ensuring this latter condition reduces the possible clicking / patchwork effects which could occur when the selected view does not cover the user viewport anymore (because of occlusions or too small FoV).

One way to do so is, for instance, to consider among all input views, the one with the most central position (closest from the barycenter of all the input views position) (see Figure 4 - top).

Another option that we prefer in this contribution is to enforce at the encoding stage the synthesis of a central view which will in most cases cover 90% of the scene (see Figure 4 - bottom), and to choose this synthetic view as the reference one.

Doing so, the end-user experiences false but consistent and static non-lambertian light effects which are more pleasant to watch.

Once selected, this camera must be signaled in the set of metadata to be properly used at the decoding stage.

**Figure 4: (top) 5x5 planar rig of cameras with most central camera #33 chosen as reference for the shading and (bottom) a synthesized equirectangular central view example for the TechnicolorMuseum content.**

## Decoding stage

At the decoding stage, the reference camera for the shading / lighting effects is extracted from the metadata bitstream. Each patch associated to this reference camera is used preferentially each time it covers the viewport to synthesize. Doing so, the light effects are baked and do not change whatever the viewing position, as expected.

## Additional Metadata

To signal the specific transport view to the decoder so that this latter may be able to use it properly at the rendering stage, the use of a single 16-bits unsigned integer field **lighting\_confidence\_id** designating the reference view could be considered. A specific value (0xffff or num\_cameras\_minus1 + 1 for instance) could be also chosen to disable the feature (see Table 1).

|  |  |
| --- | --- |
| camera\_params\_list( ) { | **Descriptor** |
| **num\_cameras\_minus1** | u(16) |
| for ( i= 0; i <= num\_cameras\_minus1; i++) { |  |
| **cam\_pos\_x**[ i ] | u(32) |
| **cam\_pos\_y**[ i ] | u(32) |
| **cam\_pos\_z**[ i ] | u(32) |
| **cam\_yaw**[i ] | u(32) |
| **cam\_pitch**[ i ] | u(32) |
| **cam\_roll**[ i ] | u(32) |
| } |  |
| **intrinsic\_params\_equal\_flag** | u(1) |
| for ( i = 0; i <= intrinsic\_params\_equal\_flag ? 0 : num\_ cameras\_minus1; i++ ) |  |
| camera\_intrinsics( [ i ] ) |  |
| **depth\_quantization\_params\_equal\_flag** | u(1) |
| for ( i = 0; i <= depth\_quantization\_equal\_flag ? 0 :  num\_cameras\_minus1; i++ ) |  |
| depth\_quantization( [ i ] ) |  |
| **lighting\_confidence\_id** | u(16) |

**Table 1: Implementation of the specular related metadata as an optional camera id**

## Implementation details

For the purpose of this experiment, a dedicated IDCCViewOptimizer has been used which role is to generate a central view, add this to the list of source views and make it as the unique basic view. This central view is tagged as the reference view to use for the shading at the decoding stage. It is located at the position of the viewport camera (as defined in the JSON files) and its field of view is adapted to fit the field of view of the source cameras. Moreover, this optimizer also allows for source rescaling, a feature which has been especially used for TechnicolorHijack to make it fit in a 4K x 4K atlas (see next section). Lastly, inpainting was disabled for the central view synthesis.

Due to an unexpected bad behavior of the TMIV 3.0 pruner when dealing with a mix of central and regular views, a IDCCPruner module has been used instead which pruning criterion is a geometric only criterion (no color-based pruning) and which behavior is very close from the one originally proposed in TMIV 1.0 and TMIV 2.0.

The group-based module was not used at the encoding and decoding stage and regular Encoder and Decoder modules were preferred.

The IDCCSynthesizer (cf CE2.6.1) was used to perform the simulations of this contribution for ease of implementation.

# Results

Overall results are provided in Table 2 (for comparison with TMIV 3.0) and 3 (for comparison with CE2.6.1), and Figures 6, 7, 8, 9, 10 and 11 show the rate / distortion diagrams for ClassroomVideo, TechnicolorMuseum, TechnicolorHijack, OrangeKitchen and NokiaChess where the study has been conducted and where non-lambertian effects where the most visible. We have also overlaid on the top of these latter graphics, the distortion of the worst view over the different rates.

Note that a single 4K x 4K atlas has been used for the packing of the patches for all contents. TechnicolorHijack source views were rescaled (75%) to make this content packable in a 4K x 4K atlas (see section 2.5). We did not try to optimize the pixel rate further as this resolution was considered as a low pixel rate by the group and as such a frame size can be real-time decoded in most devices included smartphones. However, it is noteworthy to mention that making use of a central view is a very efficient way to save pixel rate because it is, in a sense, a kind of optimal packing of 90% of the scene (see Figure 6). Finally, we also present results obtained for TechnicolorHijack (see Figure 9), at full resolution 8K x 4K without any rescaling for comparison.



**Figure 5: Example of atlases. 4K x 4K frames have been used to do the packing of all contents but the use of a central view allows for a very efficient packing which makes possible to achieve lower resolution if needed.**

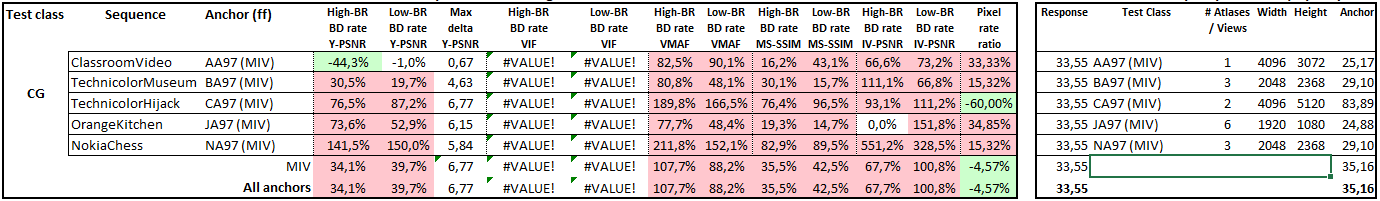
## Objective metrics

As one can observe in Table 2 (for comparison with TMIV3.0) and 3 (for comparison with CE2.6.1) and Figure 6 to 11, our quantitative / objective metrics are generally lower that the one obtained by the TMIV3.0 and CE2.6.1 (except for ClassroomVideo) but this gap is reduced when considering the worst distortion over the views.

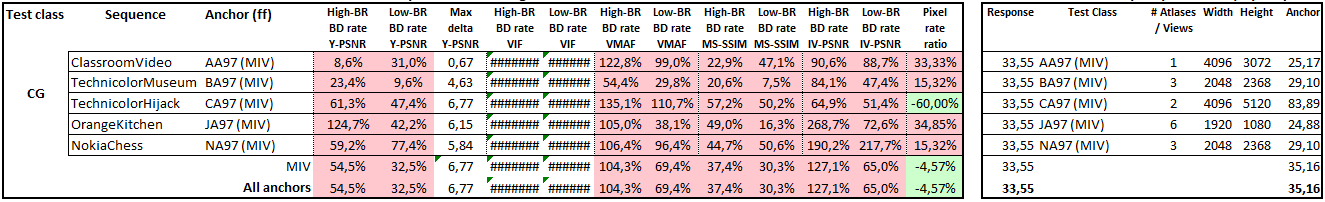
Indeed, the proposed approach synthesizes the basic view and therefore, does not transmit any full source view contrary to the MIV 3.0 anchors. Transmitting these full source views (up to 3 for OrangeKitchen in the MIV anchors because of the presence of 3 groups) tends to artificially increase the average objective metric. But when considering the worst view, things become more on par.

Another important reason which may explain the differences in terms of objective metric is related to the approach itself: whatever the camera position, the central view is preferred for the shading. The reflectivity / specularity / transparency of some source views are therefore not rendered properly which also decreases the objective metrics.

Finally, for TechnicolorHijack in particular, one can observe in Figure 8 and 9 that the downscaling of the input views to make the atlas fit in a 4K x 4K packing is mainly responsible for the loss in PSNR.



**Table 2: Overall results (TMIV3.0 vs proposal)**



**Table 3: Overall results (CE2.6.1 vs proposal)**

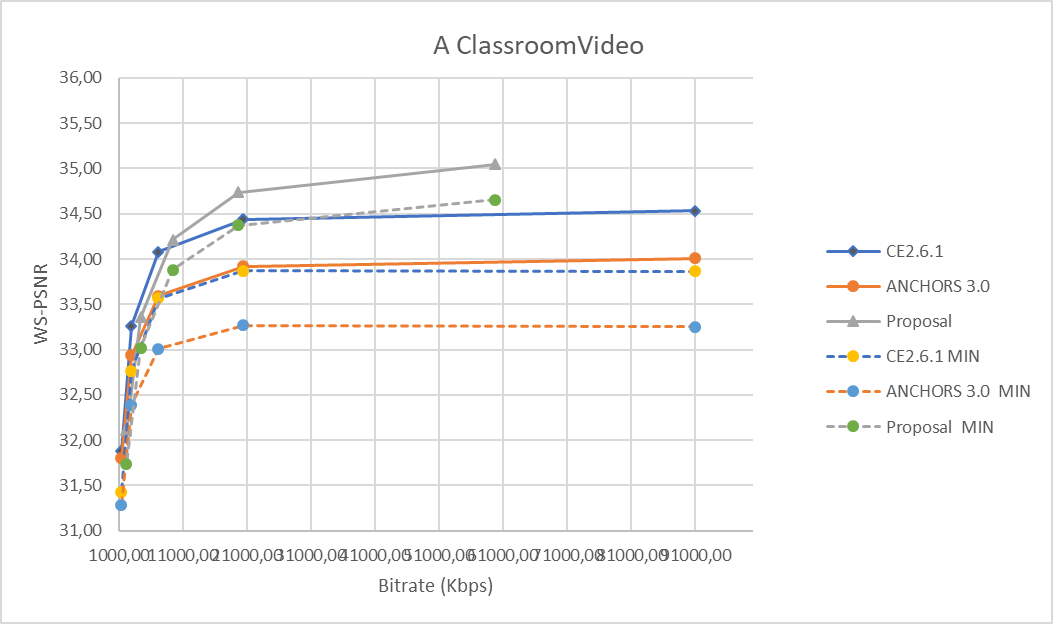
## Subjective results

Things are quite different when considering the subjective results. Some illustrations of the behavior of the proposed approach have been already proposed in Figure 2 and 3 (top). Side-by-side videos will be also proposed for viewing sessions. Baking the non-lambertian effects tends to provide more spatially and temporally consistent synthesis which is of high importance for the final feeling of the end-user.

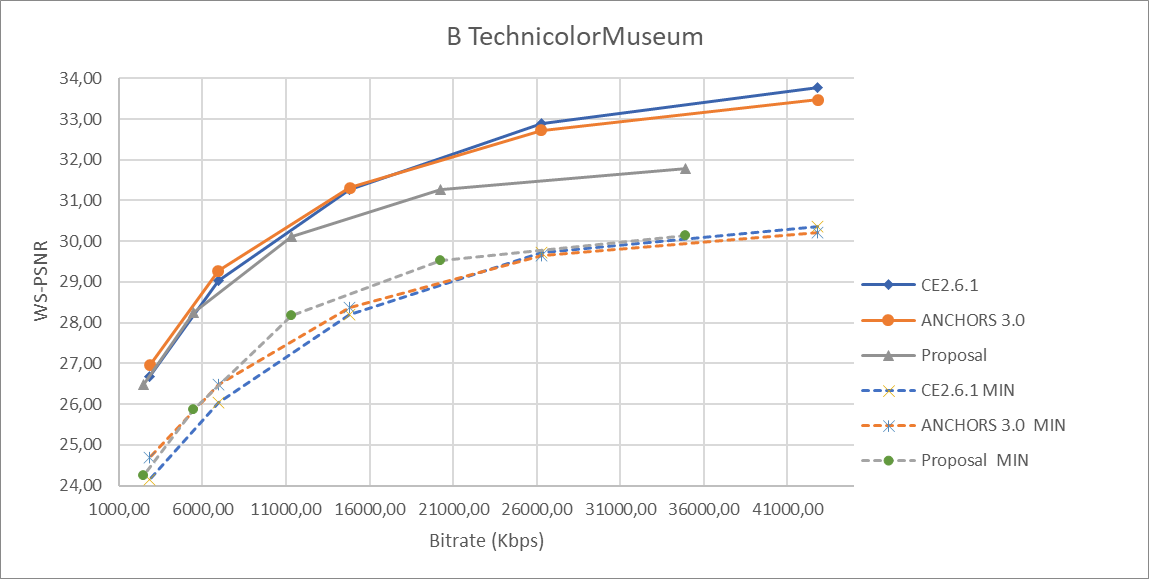
It is especially the case for ClassroomVideo (see Figure 11) and TechnicolorHijack where making use of a single synthetic reference view suppresses the patchwork effect very visible in MIV 3.0 (front tables in ClassroomVideo and roof in TechnicolorHijack) and even suppresses the speckle of ClassroomVideo as a side-effect (due to the use of the central synthetic view which averages the speckle of all views at the encoding stage).

In OrangeKitchen, the very noticeable ‘clicking / switching” effects in MIV 3.0 is really mitigated and makes the visualization more comfortable despite pruning artifacts which are not in the scope of this contribution (see Figure 12).

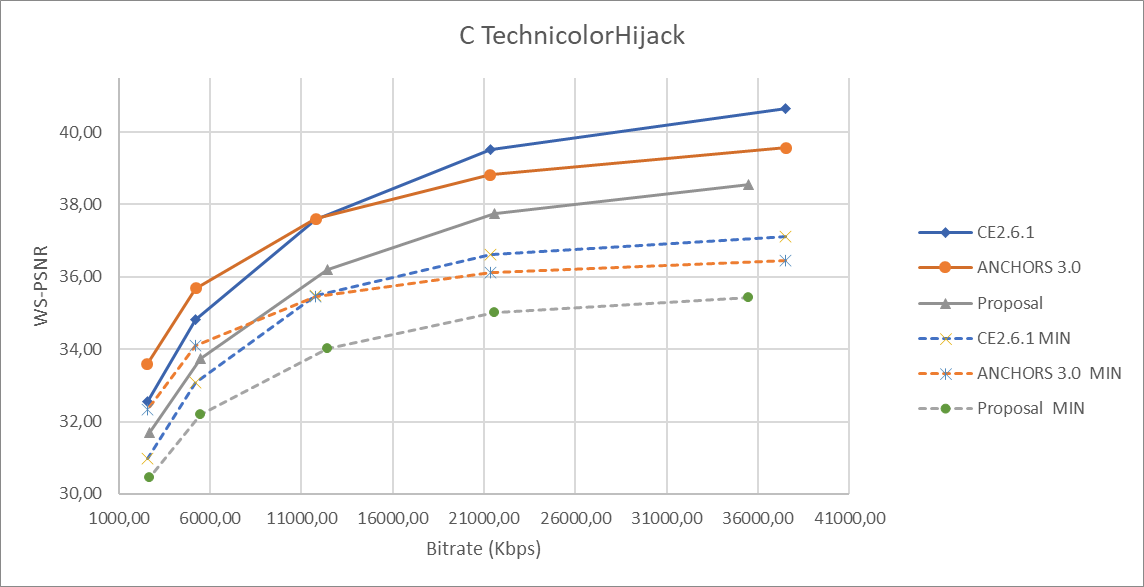
Note that the subjective improvements are mostly visible for the 2 lowest QPs configuration, the compression artifacts making things less clear for higher QPs. For NokiaChess, the quality of both solutions (MIV 3.0 and ours) is very poor which may be explained by an inappropriate depth quantization (see contribution m52424). An inpainting issue on this content also makes the interpretation of the results difficult.



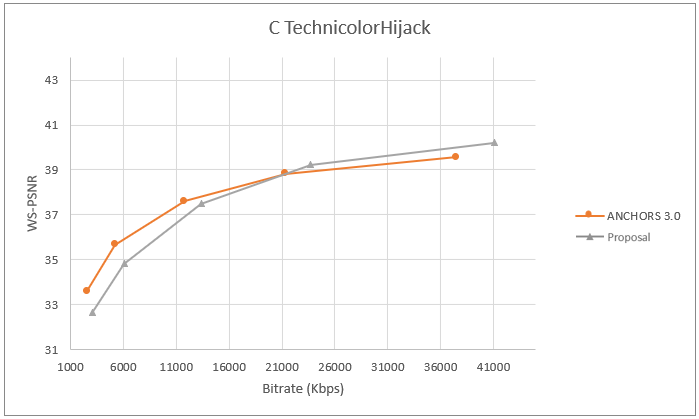
**Figure 6: Rate / Distortion (average and worst) diagram for ClassroomVideo.**



**Figure 7: Rate / Distortion (average and worst) diagram for TechnicolorMuseum.**



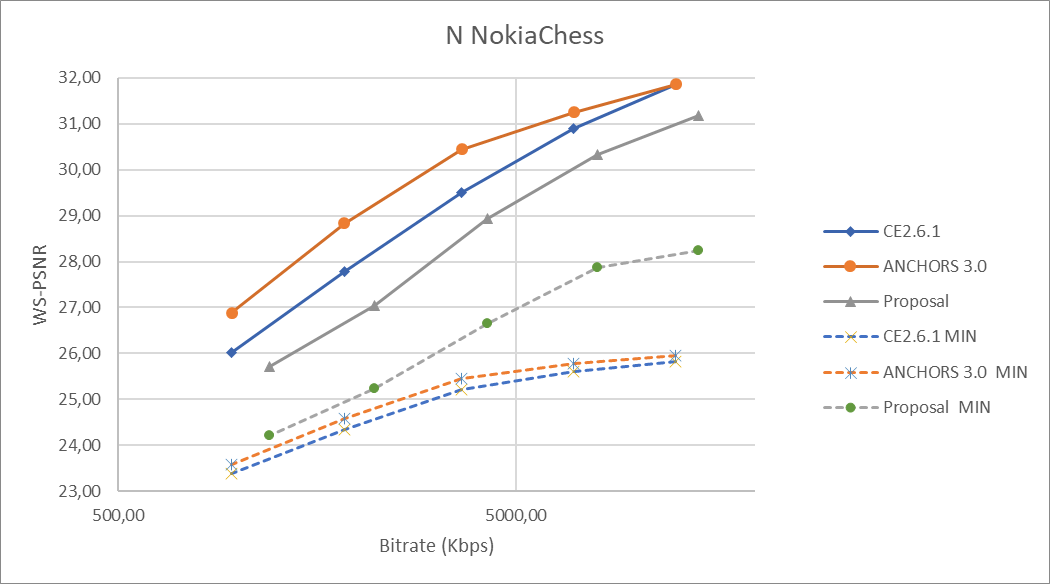
**Figure 8: Rate / Distortion (average and worst) diagram for TechnicolorHijack (4K x 4K configuration).**

****

**Figure 9: Rate / Distortion (average and worst) diagram for TechnicolorHijack (8K x 4K configuration).**



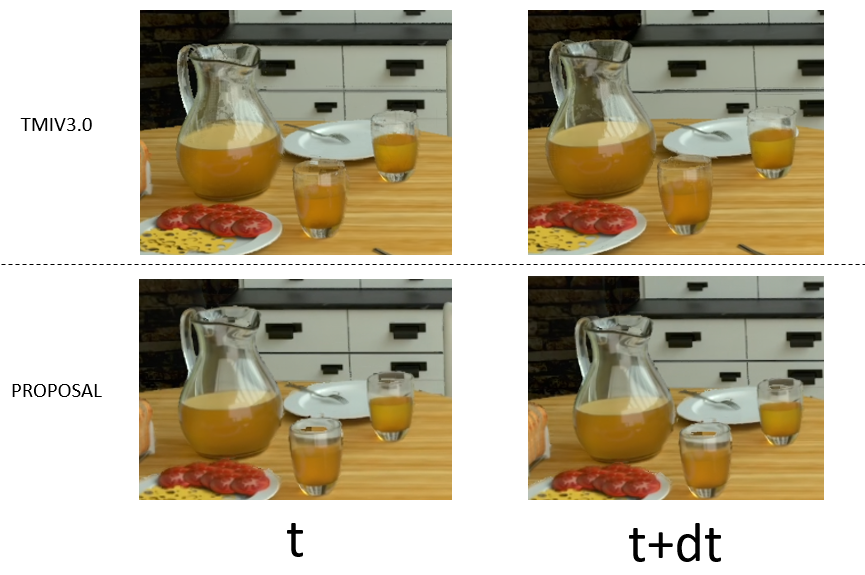
**Figure 10: Rate / Distortion (average and worst) diagram for OrangeKitchen.**



**Figure 11: Rate / Distortion (average and worst) diagram for NokiaChess.**

****

**Figure 12: Patchwork effect and speckle with TMIV3.0 (left) mitigated by our proposal (right) for VideoClassroom.**



**Figure 13: “Clicking” effect (brutal color change) on the jug in TMIV3.0 (top) and proposal (bottom) for OrangeKitchen (more demonstrative in a video than on still images…).**

# Recommendations

We recommend using an additional metadata to signal a reference view dedicated to handle properly the non-lambertian lighting effects at the decoding stage, in the context of a limited 6DoF scenario. One possible implementation is proposed in this contribution. Once again, the subjective gain is not neglectable at all at the cost of a very light additional flag.

We recommend using central views, at least for CGI contents because of the packing efficiency it brings.

We also recommend reconsidering the meaning of objective metrics, and accepting to degrade them from TMIV3.0 when it provides better subjective results. This is also the message of our other contribution for the CTC change [4]. This current proposition to carry the main texture component through a central view for all synthetic content thanks to this light metadata degrades the objective metrics but deliver better subjective results.

Finally, we also recommend adopting the notion of profile for the pixel rate. Thanks to the use of a synthetic views as transport views, the configuration used for this contribution gives a unique value of pixel rate of 4Kx4K for all contents which is consistent with the notion of profile.

# References

1. J. Jung, B. Kroon, J. Boyce, Common Test Conditions for Immersive Video, ISO/IEC JTC1/SC29/WG11 MPEG/N18789, October. 2019, Geneva, Switzerland.
2. A. Dziembowski, “Description of Immersive Video Core Experiments 2: Pixel Pruning”, ISO/IEC JTC1/SC29/WG11, N18933, October 2019, Geneva, Switzerland.
3. J. Fleureau, R. Doré, F. Thudor, T. Tapie, G. Briand, B. Chupeau, “CE2.6.1: Synthesizer”, ISO/IEC JTC1/SC29/WG11, M52414, January 2020, Brussels, Belgium.
4. R. Doré, B. Chupeau, “New QP Depth Change proposal for CTC”, ISO/IEC JTC1/SC29/WG11, M52425, January 2020, Brussels, Belgium.