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/M48092
July 2019, Göteborg, Sweden

Source	**Poznań University of Technology (PUT), Poznań, Poland** **Electronics and Telecommunications Research Institute (ETRI), Daejeon, Republic of Korea**
Status	**Input**
Title	**PUT/ETRI Response to Immersive Video CE-5: Depth and color refinement**
Author	Adrian Dziembowski*, Dawid Mieloch*, Marek Domański*, Gwangsoon Lee** * – Poznań University of Technology, ** – Electronics and Telecommunications Research Institute

1 Introduction

This document presents a technical description of PUT/ETRI experiment on input data refinement (Immersive Video CE-5) [1].

2 Overview of the proposed techniques

2.1 Depth refinement

The proposed depth refinement method is based on a technique described in [3].
The cross-view synthesis is performed in order to project depth values from all N into each of N input depth maps. After this step, for all the points in each depth map there is a list of depth values, projected from various input depth maps.

In order to provide the inter-view consistency, each point is processed in the same way:
1. All the depth values are sorted in descending order.
2. If *n* smallest depth values are similar (difference smaller than a threshold) go to step 6; else go to 3.
3. Remove first (smallest) depth value from the list.
4. If number of the elements in the list is smaller than *n*, go to step 5; else go to step 2.
5. Restore all the removed values to the list, decrement *n* and go to step 2.
6. The new depth value for analyzed point is an average value of these *n* values.

Different values of *n* were tested and *n* = 3 provides best results.

The depth maps after described refinement contains holes – areas without any depth value. These areas are simply inpainted using 8-way, depth-based inpainting method (for each pixel of the hole the depth of nearest non-hole pixel in each direction are compared; then the farthest depth is copied to the analyzed pixel).

In order to provide better consistency, all the described operations are performed twice. In the second iteration, the refined depth maps are treated as input ones.

2.2 Color refinement

Proposed color refinement is based on a color correction method described in [2]. Different views (especially natural ones, i.e. captured using physical cameras) may have inconsistent color characteristics. It results in appearance of color artifacts in the virtual view. In order to reduce this phenomenon, we propose a color refinement technique, which equalizes global color differences between real views. The global color difference between points projected from two real views is calculated as the mean ratio (averaged for the entire image) between color component projected from one view and color component projected from the second one. The algorithm is performed separately for R, G and B color components. In order to equalize colors of points projected from any real view *i*, color component values projected from view *i* are multiplied by the mean ratio between view *i* and reference view (the view acquired by closest real camera to the virtual one).

At first, the reference view is chosen. In our proposal, the most central view is chosen as a reference one. Then, each real view is projected to the position of the reference view. All the real views are processed separately.

In the second step, RGB values of every pixel of the reference view are compared to RGB values of collocated pixel reprojected from the real view. Each color component is processed in the same way, so below we described processing for one component (e.g. R).

The differences are aggregated separately for 3 ranges of R intensity: [0, 341), [341, 683) and [683, 1023]. Then, summarized difference for each range is divided by number of pixels with R value in that range. At the end of this step, we obtain 3 mean ratios for each color component.

In the next step, all the pixels in the real, non-reference views are modified by adding proper mean ratio. In order to avoid color artifacts for pixels with color value close to range boundaries, range overlapping for mean ratios adding was applied: all the R values within an overlap are modified using a weighted average of two mean ratios.

The color refinement is performed using previously refined depth maps.

3 Experimental results

In order to distinguish an influence of both proposed techniques on the quality of synthesized views, we performed three experiments.

3.1 Color refinement

In the first experiment, we used:
- refined input views,
- reference depth maps.

Test class	Sequence	Anchor	High-bitrate BD rate Y-WSPSNR	Low-bitrate BD rate Y-WSPSNR	High-bitrate BD rate VMAF	Low-bitrate BD rate VMAF	High-bitrate BD rate MM-SSIM	Low-bitrate BD rate MS-SSIM	Pixel rate ratio
NC1	TechnicolorPainter	D1 (MIV anchor)	65.5%	36.6%	-19.2%	-11.0%	-3.2%	-2.0%	0.00%
		MIV anchor, reduced input	78.8%	38.2%	-21.7%	-12.0%	-26.0%	-0.5%	0.00%
	IntelFrog	E1 (MIV anchor)	49.0%	18.3%	-43.2%	-24.1%	15.7%	-2.1%	0.00%
		MIV anchor, reduced input	33.9%	24.3%	-34.4%	-17.3%	-46.7%	-1.5%	0.00%
		MIV anchor	57.2%	27.5%	-31.2%	-17.5%	6.3%	-2.0%	0.00%
		MIV, reduced input	56.4%	31.3%	-28.1%	-14.6%	-36.4%	-1.0%	0.00%
		Both anchors	56.8%	29.4%	-29.6%	-16.1%	-15.0%	-1.5%	0.00%

Color refinement modifies luma and chroma values so, when comparing to unrefined reference views, it obviously decreases PSNR value. However, proposed method significantly reduces color inconsistencies (Fig. 1). Also VMAF and SSIM increase prove, that proposed color correction increases virtual view quality.

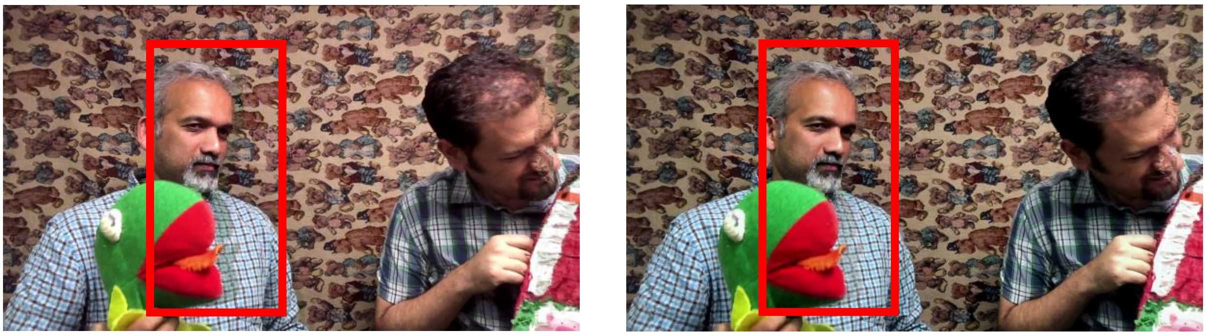

Fig. 1. No color refinement (left) vs. color refinement (right)

In this experiment we used reference depth maps, so the result of pruning and packing was exactly the same as for the anchor. Therefore, the pixel rate values did not change.

3.2 Depth refinement

In the second experiment, we used:

- reference input views,
- refined depth maps.

Test class	Sequence	Anchor	High-bitrate BD rate Y-WSPSNR	Low-bitrate BD rate Y-WSPSNR	High-bitrate BD rate VMAF	Low-bitrate BD rate VMAF	High-bitrate BD rate MM-SSIM	Low-bitrate BD rate MS-SSIM	Pixel rate ratio
NC1	TechnicolorPainter	D1 (MIV anchor)	-28.4%	-22.8%	-30.9%	-24.6%	-19.3%	-17.9%	0.0%
		MIV anchor, reduced input	-21.2%	-23.7%	-32.9%	-29.1%	-24.9%	-25.5%	-50.0%
	IntelFrog	E1 (MIV anchor)	0.0%	0.0%	0.0%	21.7%	0.0%	108.8%	-40.0%
		MIV anchor, reduced input	0.0%	0.0%	0.0%	164.8%	0.0%	0.0%	-20.0%
		MIV anchor	-14.2%	-11.4%	-15.4%	-1.5%	-9.6%	45.5%	-20.0%
		MIV, reduced input	-10.6%	-11.9%	-16.5%	67.9%	-12.5%	-12.8%	-35.0%
		Both anchors	-12.4%	-11.6%	-15.9%	33.2%	-11.0%	16.3%	-27.5%

Proposed depth refinement method increases spatial and temporal consistency of the depth maps thus reduces pixel rate. It also increases the visual, subjective quality of virtual views (Fig. 2).
For TechnicolorPainter, BD-rates for all the objective quality metrics are better, than for anchor. For IntelFrog, obtained BD-rates are worse (0.0% values in the Table above mean, that the whole curve for anchor is above curve for proposed method). The reason of that we commented in the "Remarks" section.

Fig. 2. Anchor (left) vs. depth refinement (right)

3.3 *Color and depth refinement*

In the third experiment, we used:

- refined input views,
- refined depth maps.

Test class	Sequence	Anchor	High-bitrate BD rate Y-WSPSNR	Low-bitrate BD rate Y-WSPSNR	High-bitrate BD rate VMAF	Low-bitrate BD rate VMAF	High-bitrate BD rate MM-SSIM	Low-bitrate BD rate MS-SSIM	Pixel rate ratio
NC1	TechnicolorPainter	D1 (MIV anchor)	15.1%	3.7%	-42.7%	-32.3%	-21.4%	-19.4%	0.0%
		MIV anchor, reduced input	34.3%	3.3%	-45.5%	-36.8%	-25.4%	-26.2%	0.0%
	IntelFrog	E1 (MIV anchor)	0.0%	0.0%	231.7%	-3.0%	0.0%	110.5%	-40.0%
		MIV anchor, reduced input	0.0%	0.0%	0.0%	110.9%	0.0%	0.0%	-20.0%
		MIV anchor	7.5%	1.8%	94.5%	-17.6%	-10.7%	45.6%	-20.0%
		MIV view anchor	17.2%	1.6%	-22.8%	37.1%	-12.7%	-13.1%	-10.0%
		Both anchors	12.3%	1.7%	35.9%	9.7%	-11.7%	16.2%	-15.0%

Combination of depth and color refinement methods joins advantages of both proposed solutions. It significantly increases the consistency of the depth maps, reduces color inconsistencies and decreases pixel rate.

For TechnicolorPainter, refinement significantly increases VMAF and SSIM BD-rates. For IntelFrog, BD-rates for all the quality metrics are worse. The reason of that we commented in the "Remarks" section.

4 Remarks

- For IntelFrog sequence we used wrong .json file with camera parameters. Therefore, all the projections were done incorrectly and objects slightly moved (compared to the reference). It does not reduce the subjective quality, however significantly decrease the objective quality metrics.
- The tests for IntelFrog sequence will be repeated using proper camera parameters; the results will be published in a CE-related document.
- Our depth refinement method performs worse for marginal cameras. Therefore, we think that using 15 input depth maps (including leftmost and rightmost depths: v0 and v14) would allow to additionally increase the quality of refined depth maps.

5 Acknowledgement

This work was supported by Institute of Information & Communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) (No. 2018-0-00207, Immersive Media Research Laboratory).

6 References

[1] A. Dziembowski, “Description of Immersive Video Core Experiment 5”, ISO/IEC JTC1/SC29/WG11 MPEG/N18469, Mar. 2019, Geneva, Switzerland.

[2] A. Dziembowski, “[MPEG-I Visual] Fast color correction technique for view synthesis”, ISO/IEC JTC1/SC29/WG11 MPEG/M43694, July 2018, Ljubljana, Slovenia.

[3] M. Domański, A. Dziembowski, D. Mieloch, O. Stankiewicz, J. Stankowski, A. Grzelka, G. Lee, J. Seo, „Technical description of proposal for Call for Proposals on 3DoF+ Visual prepared by Poznan University of Technology (PUT) and Electronics and Telecommunications Research Institute (ETRI)”, ISO/IEC JTC1/SC29/WG11 MPEG/M47407, Mar. 2019, Geneva, Switzerland.