INTERNATIONAL ORGANISATION FOR STANDARDISATION
ORGANISATION INTERNATIONALE DE NORMALISATION
ISO/IEC JTC1/SC29/WG11
CODING OF MOVING PICTURES AND AUDIO

ISO/IEC JTC1/SC29/WG11 MPEG2018/m43694
July 2018, Ljubljana, Slovenia

Title	**[MPEG-I Visual] Fast color correction technique for view synthesis**
Status	**Input**
Source	**Poznań University of Technology**
Authors	**Adrian Dziembowski (adziembowski@multimedia.edu.pl), Olgierd Stankiewicz**

1. Introduction

This document presents a simple but effective method of color correction for virtual view synthesis. With the proposed method it is possible to reduce the amount and visibility of color artifacts in the synthesized virtual view, especially in the areas synthesized from only one real input view (e.g. occluded in other input views). Proposed technique adjusts the color characteristics of all the input views to the characteristics of the nearest input view by adding an offset to each color component of each pixel projected from further views.

Problem of color artifacts in virtual view synthesis is presented in Fig. 1.

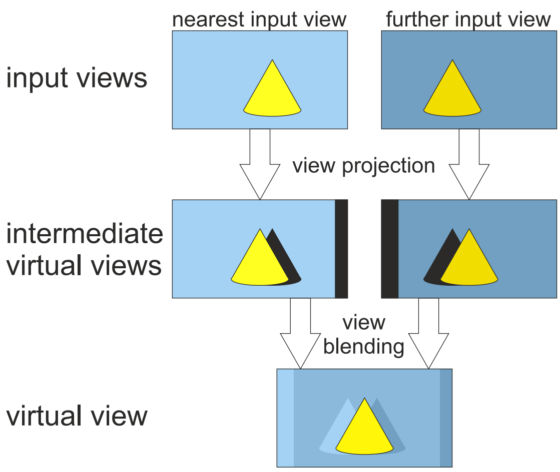

Fig. 1. View synthesis without color correction. In the example, background of the synthesized virtual view is composed of regions coming from different input views, which yields with different shades of background color.

In the first step of virtual view synthesis all the pixels of the input views are projected to the position of the virtual view (view projection step). In this step the disocclusions appear – the areas which are occluded in the input views and should be visible in the virtual one (black areas in Fig. 1). Then, during view blending colors of the pixels of the final virtual view are obtained by blending colors projected from different input views. For disoccluded area blending cannot be performed, so the virtual view is filled by the information from only one input view. It causes the color artifacts in the virtual view in the case where the color characteristics of the input views are different.

2. Color correction

2.1. Algorithm

Before color correction, the nearest input view should be chosen. It can be done manually (by putting the nearest view as the first one in a configuration file) or algorithmically, e.g. by comparing the physical distance between all the real cameras and the virtual one (measuring the Euclidean distance between virtual and real camera's translation vectors). Then, color characteristics of all the views will be adjusted to the nearest view's one.

View synthesis with proposed technique is presented in Fig. 2. The color correction algorithm consists of two phases. In the first one, the offset between the colors of the reference (nearest) and analyzed (further) input views is calculated – orange arrow in Fig.2. Then, estimated offset is added to the color values of all the pixels projected from the analyzed view – magenta arrow in Fig. 2.

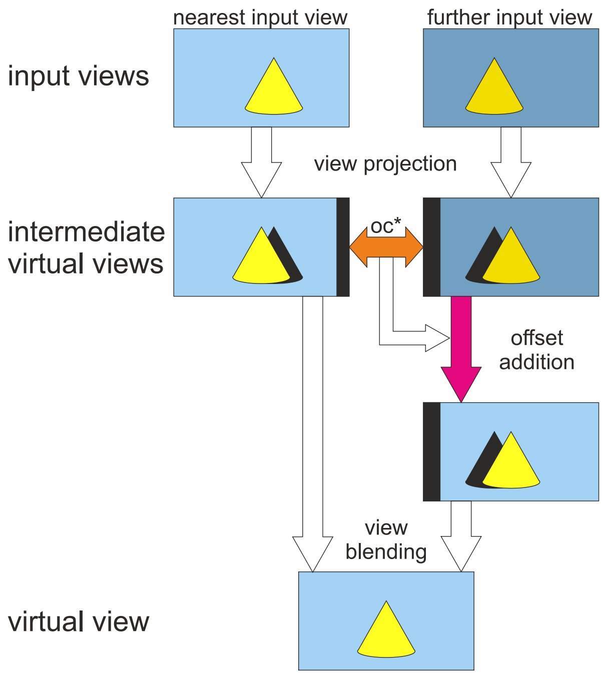

Fig. 2. View synthesis with proposed color correction (oc* – offset calculation)

The color correction offset is calculated for all the pairs of colocated pixels in both intermediate virtual views (Fig. 3).

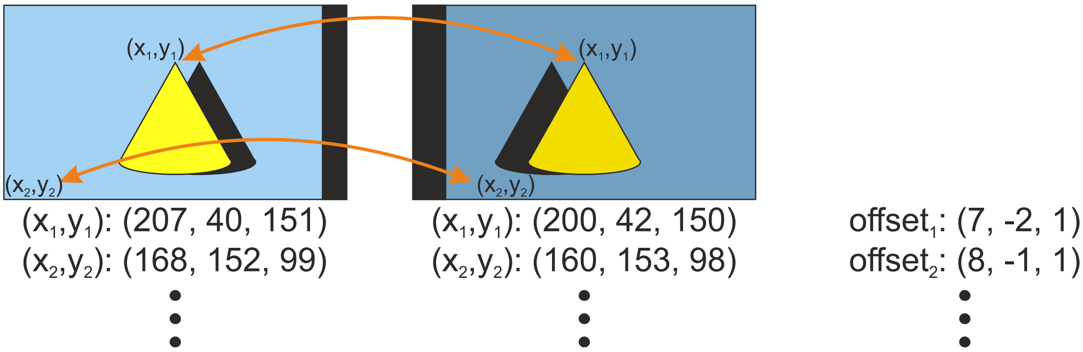

Fig. 3. Offset calculation (offset calculated separately for 3 color components, e.g. YUV)

Offset is calculated for every pixel projected from the nearest input view and the analyzed input view. In presented example, it is estimated for every pixel of the virtual view in the green area (Fig. 4).

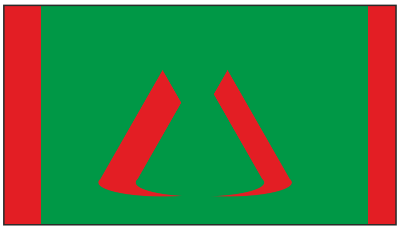
Fig. 4. Mask for offset calculation (offset is calculated for green area)

Optionally, in order to avoid the influence of the depth map artifacts, the color and depth value projected from both the input views can be compared. In that case, points with two significantly different depth or color are omitted during estimation of characteristics difference.

In the next step, all the calculated offsets are averaged in order to produce one mean offset for the entire input view.

In the second phase of proposed color correction algorithm, previously estimated offset is added to the color values of points projected from the further input view. Of course, the calculated color value with added offset may exceed the value range allowed for the pixel format used (e.g. 8-bit). In order to avoid that, saturation arithmetic is used instead of normal bit-overflow arithmetic.

All of the operations presented in this subsection are performed separately for all the input views (except for the nearest input view, which is not modified).

The proposed technique was implemented and tested in VSRS [M40657] and SVS [M42343]. For both algorithms it requires only two simple additional loops (with few additions/subtractions) per input view, what makes proposed technique fast and simple.

2.2. Results

Presented technique was tested on six multiview sequences: *Ballet*, *Breakdancers*, *Poznan_Fencing2*, *Soccer Linear*, *BBB Butterfly Arc* and *BBB Flowers Arc*. For each natural sequence the virtual view at the position of the real view 4 was synthesized using views 3 and 5, for both *BBB* sequences views 32 and 58 were used to synthesize view 45.

Test sequence	PSNR [dB]			
	VSRS		SVS	
	No color correction	With color correction	No color correction	With color correction
Ballet	31.45	**31.96**	31.72	**32.55**
Breakdancers	31.89	31.89	32.09	32.16
Poznan_Fencing2	26.14	**28.04**	26.79	**28.50**
Soccer Linear	34.80	34.80	34.92	34.94
BBB Butterfly	32.06	32.06	33.94	33.91
BBB Flowers	22.71	22.71	22.51	22.48

In the case of sequences, where there are big differences between the color characteristics of the input views, the quality increase is significant. For *Ballet* sequence it is 0.7 dB, for *Poznan_Fencing2*: 1.8 dB.

In the case of sequences, where the input views have similar color characteristics, the quality difference is negligible for SVS and there is practically no difference for VSRS. The reason is the type of data in both algorithms – VSRS operates on integers, SVS on floating point numbers.

Apart from the promising objective PSNR results, what is the most important, the virtual view synthesized using proposed color correction technique is subjectively better – there are less annoying artifacts, especially behind the foreground objects (Figs. 5 and 6).

VSRS, no color correction (31.4 dB)

SVS, no color correction (31.7 dB)

VSRS, with color correction (32.0 dB)

SVS, with color correction (32.6 dB)

Fig. 5. Fragment of the virtual view synthesized using VSRS and SVS, with and without color correction (*Ballet*)

VSRS, no color correction (26.1 dB)

SVS, no color correction (26.8 dB)

VSRS, with color correction (28.0 dB)

SVS, with color correction (28.5 dB)

Fig. 6. Fragment of the virtual view synthesized using VSRS and SVS, with and without color correction (*Poznan_Fencing2*)

3. Conclusions

The proposed color correction technique allows to significantly improve the quality of synthesized virtual views in the situations, where the input views have different color characteristics. For *Ballet* and *Poznan_Fencing2* test sequences, the PSNR of the virtual views synthesized with proposed color correction is significantly higher than without color correction. For *Ballet* it is 0.7 dB higher, for *Poznan_Fencing2*: 1.8 dB. Also the subjective quality of the synthesized views is higher because of the elimination of disturbing color artifacts behind foreground objects.

4. Recommendation

Due to the promising effects and low complexity we recommend to add proposed color correction method to View Synthesis Reference Software.

5. References

[M42343] Sarah Fachada, Daniele Bonatto, Arnaud Schenkel, Gauthier Lafruit, "[MPEG-I Visual] View Synthesis with multiple reference views"

[M40657] Takanori Senoh, Kenji Yamamoto, Nobuji Tetsutani, Hiroshi Yasuda, Krzysztof Wegner, "View Synthesis Reference Software (VSRS) 4.2 with improved inpainting and hole filling"