ISO/IEC JTC 1/SC 29/WG 11
Coding of moving pictures and audio
Convenorship: UNI (Italy)

Document type:	**Approved WG 11 document**
Title:	**Manual of color correction tool**
Status:	**Approved**
Date of document:	**2020-01-31**
Source:	**Video**
Expected action:	
No. of pages:	**5**
Email of convenor:	**leonardo@chiariglione.org**
Committee URL:	**http://mpeg.chiariglione.org**

INTERNATIONAL ORGANISATION FOR STANDARDISATION
ORGANISATION INTERNATIONALE DE NORMALISATION
ISO/IEC JTC 1/SC 29/WG 11
CODING OF MOVING PICTURES AND AUDIO

ISO/IEC JTC 1/SC 29/WG 11 N Error! Reference source not found.

Brussels, Belgium – January 2020

Source:	Video
Title:	Manual of color correction tool
Editor:	Adrian Dziembowski

1 Introduction

This document describes Poznań Color Refinement (PCR), the color correction tool for immersive video purposes. The method allows to improve temporal and inter-view consistency of color characteristics of input views. The PCR method consists of two techniques. In the first one, all views are processed independently in order to equalize their color characteristics over time. Then, color characteristics of all views are unified.

1.1 Temporal consistency improvement

The temporal consistency improvement technique can be divided into three main steps.

In the first step, the background (with highly reduced resolution) for the input view is calculated. At first, the input view is decimated 16 times in both directions (without any filtering, simply by removing another 15 samples). For each pixel of reduced input view the vector containing color values from all frames is created. Then, for each pixel the median value is calculated (separately for each color component). Despite containing also some parts of the foreground objects, the resulting image is called "background" in order to keep the simplicity of the description.

Next two steps are performed independently for all frames of the sequence.

In the second step, a global color difference between background and frame i is calculated. The global color difference is calculated as an average difference for each color component between background and frame i. Each pixel of i^{th} frame of reduced input view is compared to colocated pixel of the background. If the difference is smaller than the fixed threshold (10% of color range for Y, 5% for both chromas) for all 3 color components, it is assumed, that both pixels represent the same object and the difference is caused by temporally changing color characteristics. In this case, analyzed pixel is taken into account when calculating global color difference. Otherwise, it is omitted.

In the third step, the previously calculated global color difference is subtracted from the color component values for all pixels of i^{th} frame of a full resolution input view.

1.2 Inter-view consistency improvement

The second technique used in the proposed color refinement method is an extension of Immersive Video CE-5 response [1]. The inter-view consistency improvement is performed on temporally improved views.

The global color difference between points projected from two real views is calculated as the average difference (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 Y, C_b and C_r color components. In order to equalize colors of points projected from any real view *i*, the global color difference between view *i* and reference view (the view acquired by closest real camera to the virtual one) is subtracted from color component values projected from view *i*.

At first, each real view is projected to the position of the reference view (first view in the configuration file, see section 2.1.2). All real views are processed separately.

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

The differences are aggregated separately for 3 equal ranges of Y intensity: [0, 341), [341, 683) and [683, 1023]. Then, the summarized difference for each range is divided by number of pixels with Y value in that range. As a result of this step, 3 global color differences for each color component are obtained.

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

In order to reduce the possibility of flickering when common area of the scene visible in reference and corrected view, the global color differences are filtered using simple IIR filter. Each global color difference for frame *i* is modified using:

$$GCD'_{c,r}(i) = GCD_{c,r}(i-1) \cdot w_{prev} + GCD'_{c,r}(i) \cdot \left(1 - w_{prev}\right),$$

where $GCD'_{c,r}(i)$ is the global color difference (for frame *i*, color component *c* and intensity range *r*) after filtering and w_{prev} is the weight for $GCD'_{c,r}$ of the previous frame. Weight w_{prev} was set to 0.5.

2 Software manual

PCR tool requires a configuration file. Path to that file should be typed as a command line argument:

```
PCR   config.cfg
```

2.1 Configuration file

Two examples of a configuration file are attached to this manual.

2.1.1 Common parameters

```
NumberOfInputViews            # parameters for each input view should be included (see: 2.1.2)
NumberOfOutputViews           # parameters for each output view should be included (see: 2.1.3)

NumberOfFrames
StartFrame

DepthBlendingThreshold        # if difference between depth values is lower than this threshold,
                              # they are assumed to be the same; recommended: 40 for 10bps

RealCameraParameterFile       # path to camera parameter file (see: section 2.2)
Width
Height
Format                        # Perspective or Omnidirectional

ZNear                         # may be overwritten for each input or output view
ZFar                          # may be overwritten for each input or output view
ViewChromaSubsampling         # 420 or 444, may be overwritten for each input or output view
ViewBitsPerSample             # 8 – 16, may be overwritten for each input or output view
DepthChromaSubsampling        # 400 or 420, may be overwritten for each input or output view
DepthBitsPerSample            # 8 – 16, may be overwritten for each input or output view
```

2.1.2 Input view parameters

```
Input0 {                      # Input 0 is the reference view
    CameraName                # the same as in camera parameters file
    View                      # path to input .yuv file
    Depth                     # path to input .yuv file
    ZNear                     # may be skipped if the same as in section 2.1.1
    ZFar                      # may be skipped if the same as in section 2.1.1
    ViewBitsPerSample         # may be skipped if the same as in section 2.1.1
    ViewChromaSubsampling     # may be skipped if the same as in section 2.1.1
    DepthBitsPerSample        # may be skipped if the same as in section 2.1.1
    DepthChromaSubsampling    # may be skipped if the same as in section 2.1.1
}
Input1 {                      # Inputs 1, 2, … will be refined according to the reference view
    CameraName                # the same as in camera parameters file
```

```
    Depth                      # path to input .yuv file
    View                       # path to input .yuv file
}
…
```

2.1.3 Output view parameters

```
Output0 {
    CameraName                 # the same as in camera parameters file
    View                       # path to output .yuv file
    ViewBitsPerSample          # may be skipped if the same as in section 2.1.1
    ViewChromaSubsampling      # may be skipped if the same as in section 2.1.1
}
Output1 {
    CameraName                 # the same as in camera parameters file
    View                       # path to output .yuv file
}
…
```

2.2 Camera parameters

Current version of PCR tool requires camera parameters in VSRS-style format (intrinsic and extrinsic parameters matrix for each camera):

Camera_name

f_x	0	c_x	
0	f_y	c_y	
0	0	1	
0			
0			
r_{00}	r_{01}	r_{02}	t_0
r_{10}	r_{11}	r_{12}	t_1
r_{20}	r_{21}	r_{22}	t_2

3 Examples

1. Color refinement of IntelFrog sequence (configuration file attached):

```
PCR  SE.cfg
```

2. Color refinement of PoznanFencing sequence (configuration file attached):

```
PCR  SL.cfg
```

4 Software

MPEG Git Repository: http://mpegx.int-evry.fr/software/MPEG/Explorations/6DoF/PCR
Software coordinator: Adrian Dziembowski, adrian.dziembowski@put.poznan.pl

5 References

[1] A. Dziembowski, D. Mieloch, M. Domański, G. Lee, "PUT/ETRI Response to Immersive Video CE-5: Depth and color refinement", ISO/IEC JTC1/SC29/WG11 MPEG/M48092, Jul. 2019, Göteborg, Sweden.