

# LOCAL METHOD OF COLOR-DIFFERENCE CORRECTION BETWEEN STEREOSCOPIC-VIDEO VIEWS

*Sergey Lavrushkin, Vitaliy Lyudvichenko, Dmitriy Vatolin*

Lomonosov Moscow State University, Russian Federation

## ABSTRACT

Many factors can cause color distortions between stereoscopic views during 3D-video shooting. Numerous viewers experience discomfort and headaches when watching stereoscopic videos that contain such distortions. In addition, 3D videos with color differences are hard to process because many algorithms assume brightness constancy.

We propose an automatic method for correcting color distortions between stereoscopic views and compare it with analogs. The comparison shows that our proposed method combines high color-correction accuracy with relatively low computational complexity.

**Index Terms** — Stereoscopic video, color distortions, quality assessment, color correction

## 1. INTRODUCTION

Despite extensive development of technologies for showing 3D video, public interest in stereoscopic content is decreasing [1]. Experts in the field believe visual discomfort, which some viewers experience while watching low-quality stereoscopic content, is the main reason for this decline in consumer demand for 3D video. Stereoscopic video can contain artifacts that directly cause viewer discomfort, such as geometry, sharpness, and color distortion, as well as channel and temporal mismatch. Theoretically, computer processing can correct most of these artifacts during the production stage. Analyses of modern stereoscopic movies [2], however, show that even high-budget movies contain a considerable number of stereoscopic artifacts. Nevertheless, there is an obvious dependence between the budget of a stereoscopic movie and the number of artifacts it contains. This fact suggests that modern instruments for assessing the quality of stereoscopic content are poorly developed: correcting artifacts requires a considerable amount of human and financial resources, and most of this work is insufficiently automated. Thus, the 3D-video industry needs effective tools that reduce the number of artifacts in stereoscopic content.

In addition to visual discomfort, stereoscopic video sequences containing color distortions between views may present extra difficulties when attempting to eliminate other artifacts or when applying visual effects. For example, most stereo-matching algorithms assume the sequence has no color distortions between views. Real video data, however, often contains such distortions, causing these algorithms to deliver unsatisfactory results. Thus, not only do methods for correcting color differences between views reduce visual discomfort, but they also narrow the class of input data, allowing use of more-efficient algorithms for subsequent 3D-video processing.

## 2. RELATED WORK

Methods for correcting color distortions between stereoscopic views are divisible into two categories: local and global. Local methods produce color-distortion models that take into account the spatial position of pixels in the frame, whereas global methods apply to the views a transformation function that does not depend on the spatial position of pixels.

### 2.1. Global methods

Global color-distortion models contain few parameters, and these parameters are easy to calculate. For this reason, global methods have high computational performance and are generally more robust than local methods, meaning they are less likely to produce visible artifacts in the color-correction results. But they are often unable to provide high-quality correction of color distortion, so they usually serve in real-time applications and in the preprocessing phase of local methods.

Histogram-matching-based methods [3, 4] constitute one global approach to color correction. Their parameters are computed to minimize the difference between the color distributions of two images. Although histogram matching has low computational complexity and relatively high accuracy for a global method, it turns out to be impractical because it generates unacceptable artifacts for certain video sequences — especially those containing many occlusions.

In addition, the color-transfer methods [5, 6] are easily adaptable for stereoscopic color correction. Xiao et al. [5] generalize the basic color-transfer idea [6], performing linear color transformations for all image color channels at once rather than for each channel separately. This method has low computational cost and avoids producing visible artifacts because its model has only 12 parameters. But it has lower color-correction accuracy than the histogram-matching algorithm.

### 2.2. Local methods

Because local models of color distortion usually contain many parameters, pixel correspondence between the two views becomes necessary to correctly evaluate those models. Therefore, they can be classified according to the stereo-matching methods and color-distortion models they employ (as well as methods for estimating their parameters).

Methods [7, 8] use SIFT for stereo matching, as it is invariant to small color distortions. The disadvantage of SIFT is that the resulting correspondence vector field is highly sparse — especially over untextured image areas, where it is seldom possible to detect any feature points. Many methods for estimating optical-flow or disparity maps are easily adapted to color-distorted images by replacing the block-matching cost function. Hirschmuller et al. [9] compared 15 simple cost functions for three different matching algorithms. Their modified normalized cross-correlation function

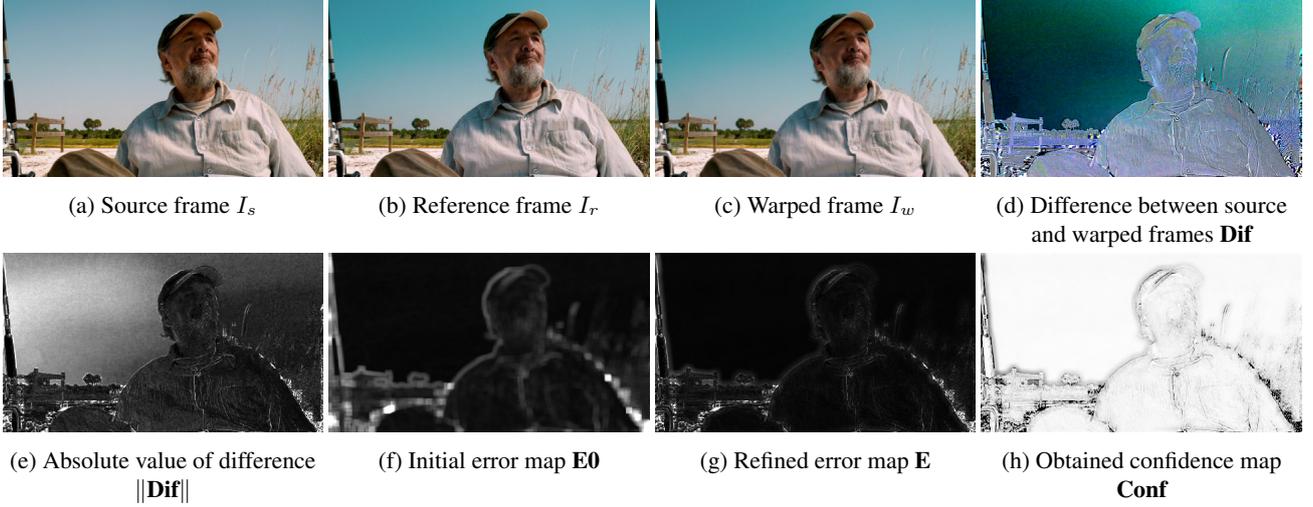


Figure 1: Confidence-map calculation example. The frame is from *Dolphin Tale*.

( $NCC$ ) showed good results in this comparison. The disadvantage of all cost functions based on  $NCC$ , however, is that they show a zero error between a uniform block and any other block. Therefore, they best serve only in matching methods that allow specification of smoothness constraints for the matching function (for example, on the basis of graph cuts). But such methods have a high computational complexity.

As for color-distortion models, methods [7, 8] model color distortions as a function of the difference between the original and corrected images; they estimate its parameters using color differences in the areas of feature points. These parameters are then interpolated for whole image. To exclude information from incorrectly matched pixels, the authors use confidence maps. In method [7], this step is based on the  $NCC$  function; in method [8] it is based on the distance between SIFT descriptors.

### 3. PROPOSED METHOD

To correct the color distortions between stereoscopic views, we transform one stereoscopic view,  $I_r$ , to match the colors of another view,  $I_s$ , using the following approach. First, our method performs stereo matching between  $I_s$  and  $I_r$  using the block-matching algorithm with a special cost function, thereby obtaining the correspondence map  $M$  between views. We then warp  $I_r$  to  $I_s$ :  $I_w = M(I_r)$ . Next, we use  $I_w$  to calculate the color-distortion model, but we first construct a confidence map to exclude information from badly matched pixels.

#### 3.1. Stereo matching

##### 3.1.1. Proposed cost functions

We examined two cost functions,  $NCC$  and  $ZSAD$  [9], that are useful for stereo matching between views with color differences. For example, if blocks  $\mathbf{b}_s, \mathbf{b}_r$  are linearly dependent:

$$\mathbf{b}_r = \alpha_{base} + (1 + \alpha_{contrast})\mathbf{b}_s,$$

where  $\alpha_{base}, \alpha_{contrast}$  are the distortion parameters for the base level of brightness and contrast, respectively, then  $NCC(\mathbf{b}_s, \mathbf{b}_r)$  will be equal to 0, and if  $\alpha_{contrast} = 0$ , then  $ZSAD(\mathbf{b}_s, \mathbf{b}_r)$  will be equal to 0. Thus, stereo matching can yield not only a correspondence map but also the color-distortion parameters  $\mathbf{A}_{base}, \mathbf{A}_{contrast}$ . Unfortunately, these cost functions fail to provide good

stereo matching because for a uniform block  $\mathbf{b}_r$  ( $\mathbf{b}_r(i, j) = const; i, j \in \{1, \dots, blocksize\}$ ),

- $NCC(\mathbf{b}_s, \mathbf{b}_r) = 0$  for any block  $\mathbf{b}_s$ ;
- $ZSAD(\mathbf{b}_s, \mathbf{b}_r) = 0$  for any other uniform block  $\mathbf{b}_s$ .

To address this issue, we take into account the color-distortion strength, which for the  $ZSAD$  cost function is defined by the parameter  $\hat{\alpha}_{base}$  and for the  $NCC$  cost function is defined by the pair  $(\hat{\alpha}_{base}, \hat{\alpha}_{contrast})$ . Here,  $\hat{\alpha}_{base}$  is the mean block difference and  $\hat{\alpha}_{contrast}$  is the block-difference standard deviation. The final cost functions are the following:

$$MZSAD(\mathbf{b}_s, \mathbf{b}_r, \beta, db, dd) = ZSAD^{db}(\mathbf{b}_s, \mathbf{b}_r) + \beta DS_{ZSAD}^{dd},$$

$$MNCC(\mathbf{b}_s, \mathbf{b}_r, \beta, db, dd, \gamma) = NCC^{db}(\mathbf{b}_s, \mathbf{b}_r) + \beta DS_{NCC}^{dd}(\gamma),$$

$$DS_{ZSAD} = |\hat{\alpha}_{base}|,$$

$$DS_{NCC}(\gamma) = (1 - \gamma)|\hat{\alpha}_{base}| + \gamma|\hat{\alpha}_{contrast}|.$$

##### 3.1.2. Selecting cost function

To find optimal parameters for proposed cost functions  $MZSAD$  and  $MNCC$ , we prepared a data set consisting of 5,000 Full HD frames from modern stereoscopic movies. Since obtaining ground-truth correspondence maps for such a data set is difficult, we use an LRC-based metric [9] to evaluate cost-function performance:

$$loss = \sum_p \mathbf{LRC}^2(p),$$

$$\mathbf{LRC}(p) = \|\mathbf{M}_{s \rightarrow r}(\mathbf{M}_{r \rightarrow s}(p)) - p\|,$$

where  $\mathbf{M}_{s \rightarrow r}$  and  $\mathbf{M}_{r \rightarrow s}$  are correspondence maps obtained using stereo matching from  $I_s$  to  $I_r$  and from  $I_r$  to  $I_s$ , respectively. We selected the optimal parameters using a brute-force search of the uniform grid. In addition to determining the cost-function parameters, we selected an optimal scale factor from the set  $\{1, \frac{1}{2}, \frac{1}{3}, \frac{1}{4}\}$  and used it to reduce the frame size before stereo matching. As a result, the smallest loss-metric value came from the following cost function when using a scaling factor of  $\frac{1}{4}$ :

$$BlockMetric(\mathbf{b}_s, \mathbf{b}_r) = ZSAD^2(\mathbf{b}_s, \mathbf{b}_r) + \frac{1}{128} DS_{ZSAD}$$

### 3.2. Correction method for local distortions

#### 3.2.1. Confidence-map construction

We compute the confidence map in two stages:

1. Evaluation of the initial correspondence-error map on the basis of the LRC ( $\mathbf{E}_{LRC}$ ) [9] and the magnitude of the structural differences between views ( $\mathbf{E}_S$ ):

$$\begin{aligned}\mathbf{E}_0 &= \mathbf{E}_S(1 + 0.25 \mathbf{E}_{LRC}), \\ \mathbf{E}_{LRC} &= \min(1, \frac{10}{w} \mathbf{LRC}), \\ \mathbf{E}_S(p) &= \frac{1}{\|\omega_{E_S}\|} \sum_{p \in \omega_{E_S}(p)} \|\mathbf{Dif}(p) - \overline{\mathbf{Dif}}(p)\|^2, \\ \overline{\mathbf{Dif}}(p) &= \frac{1}{\|\omega_{E_S}\|} \sum_{p \in \omega_{E_S}(p)} \mathbf{Dif}(p), \\ \mathbf{Dif} &= I_w - I_s,\end{aligned}$$

where  $w$  is the frame width and  $\omega_{E_S}(p)$  is the square window centered on pixel  $p$  with a radius of nine pixels.

2. Error map refinement. Initial error map  $\mathbf{E}_0$  fails to accurately mark incorrectly mapped blocks. For example, it will mark the whole block as incorrect even though the block has only a partially incorrect mapping. The refinement algorithm redistributes the error  $\mathbf{E}_0$  of each pixel  $p$  among its neighbors  $\omega_{Conf}(p)$ . During redistribution, a neighbor  $k \in \omega_{Conf}(p)$  receives from pixel  $p$  part of its error  $\mathbf{E}_{RE}(p, k)$ ; that part is proportional to the magnitude of the absolute difference  $\mathbf{Dif}(k)$ :

$$\mathbf{E}_{RE}(p, k) = \begin{cases} 0 & , k \notin \omega_{Conf}(p) \\ \mathbf{E}_0(p) \frac{\|\mathbf{Dif}(k)\|}{\sum_{k \in \omega_{Conf}(p)} \|\mathbf{Dif}(k)\|} & , k \in \omega_{Conf}(p) \end{cases}$$

where  $\omega_{Conf}(p)$  is the square window centered on pixel  $p$  with radius of five pixels. The final error value for pixel  $p$  is the sum of the errors  $\mathbf{E}_{RE}(k, p)$  redistributed from each neighbor  $k \in \omega_{Conf}(p)$ :

$$\mathbf{E}(p) = \sum_{k \in \omega_{Conf}(p)} \mathbf{E}_{RE}(k, p).$$

We use scaled and inverted error-map values as our confidence-map values.

#### 3.2.2. Estimation of local distortions

To correct local color distortions, we use the following model, which describes the dependence between the corrected image  $I_c$  and the source image  $I_s$ :

$$I_c(p) = \mathbf{a}(p) I_s(p) + \mathbf{b}(p).$$

We compute the parameters  $\mathbf{a}(p)$  and  $\mathbf{b}(p)$  independently for each pixel  $p$ , using the information from its neighbors  $k \in \omega_C(p)$  in the images  $I_s$  and  $I_w$ , taking into account each pixel confidence  $\mathbf{Conf}(k)$  pixel. For each pixel, we calculate the radius of its neighborhood:

$$r_C(p) = \min \left\{ r : \sum_{k \in \omega(p, r)} \mathbf{Conf}(k) \geq \|\omega(p, 9)\| \right\},$$

where  $\omega(p, r)$  is the square window centered on pixel  $p$  with radius of  $r$  pixels. By doing so, we guarantee the minimum total confidence measure for the pixels we use to estimate  $\mathbf{a}(p)$  and  $\mathbf{b}(p)$ .

To estimate these parameters, two methods are possible: regression and statistical. The regression method selects the parameters for pixel  $p$  that minimize the difference between the corrected and warped images in the neighborhood  $\omega_C(p)$ . To do so, it solves the following linear-regression problem:

$$\alpha, \beta = \underset{\alpha, \beta}{\operatorname{argmin}} \sum_{k \in \omega_C(p)} \left( ((1 + \alpha)I_s(k) + \beta - I_c(k))^2 + \varepsilon \alpha^2 \right) \mathbf{Conf}(k),$$

$$\mathbf{a}(p) = 1 + \alpha, \mathbf{b}(p) = \beta,$$

where  $\varepsilon$  is the regularization parameter. The solution to this problem is the following:

$$\alpha = \frac{\operatorname{cov}(I_c - I_s, I_s, \omega_C(p))}{\operatorname{std}^2(I_s, \omega_C(p)) + \varepsilon},$$

$$\beta = \operatorname{mean}(I_c - (1 + \alpha)I_s, \omega_C(p)),$$

where, for window  $\omega$ ,  $\operatorname{mean}(I, \omega)$  is the expected value of image  $I$ ,  $\operatorname{std}(I, \omega)$  is the standard deviation of image  $I$ , and  $\operatorname{cov}(I, J, \omega)$  is the covariance between images  $I$  and  $J$ , taking into account confidence  $\mathbf{Conf}$ . The statistical method assumes that within the vicinity of  $\omega_C(p)$ , image pixels for each color channel are normally distributed; it also assumes that to eliminate the color differences, it is sufficient to equalize the expected values and standard deviations of image  $I_s$  image in accordance with  $I_w$ :

$$\mathbf{a}(p) = \frac{\operatorname{std}^2(I_c, \omega_C(p)) + \varepsilon}{\operatorname{std}^2(I_s, \omega_C(p)) + \varepsilon}.$$

Next,  $\mathbf{b}(p)$  is calculated in the same way as the regression method. The regression method of parameter estimation provides more accurate correction of color distortions, but it is more inclined to generate artifacts than the statistical method.

## 4. EXPERIMENTAL EVALUATION

To evaluate the correction accuracy of our proposed method, we prepared a data set consisting of 1,000 Full HD frames taken from modern stereoscopic movies produced using only 3D rendering, so they had no initial color distortions. We artificially distorted one view in each frame to add color differences. Our accuracy evaluation used the SSIM metric. To check whether our algorithm preserves the structure of the original images after correction, we prepared a data set consisting of 5,000 Full HD frames, with color distortions, taken from modern stereoscopic movies shot using stereo cameras. To measure the structural similarity of the images, we used the  $\nabla YPSNR$  metric:

$$\nabla YPSNR(I_s, I_c) = -10 \log_{10} \frac{\sum_p \nabla Y^2(p)}{w h},$$

$$\nabla Y = \operatorname{dilate} \left( \left\| \nabla I_s^Y - \nabla I_c^Y \right\| \right),$$

where  $\operatorname{dilate}(I)$  is a dilation operation with a square structural element of size  $9 \times 9$ ,  $\nabla I^Y$  is the Y-channel gradient of image  $I$  (in YUV color space), and  $w$  and  $h$  are the frame width and height, respectively. Additionally, we evaluated the subjective quality of our proposed method using Subjectify.us platform. We created a test set consisting of 18 scenes with color corrected left views and showed them alongside with right views in a chessboard manner. Viewers were asked to choose the best color correction results with least noticeable color differences between left and right views and other visual artifacts. 116 respondents participated in the described subjective evaluation.

	Commercial method A	Commercial method B	Commercial method C	Proposed method
Correction accuracy (SSIM)	0.9981	0.9980	0.9951	0.9992
Structure preservation ( $\nabla Y P S N R$ )	31.786	39.772	33.289	45.707
Subjective quality (Crowd Bradley-Terry rank)	0.797393	—	1.040402	1.264027
Computation time (frames per second)	0.09	0.11	0.25	1.37

Table 1: Comparison of color-correction methods.

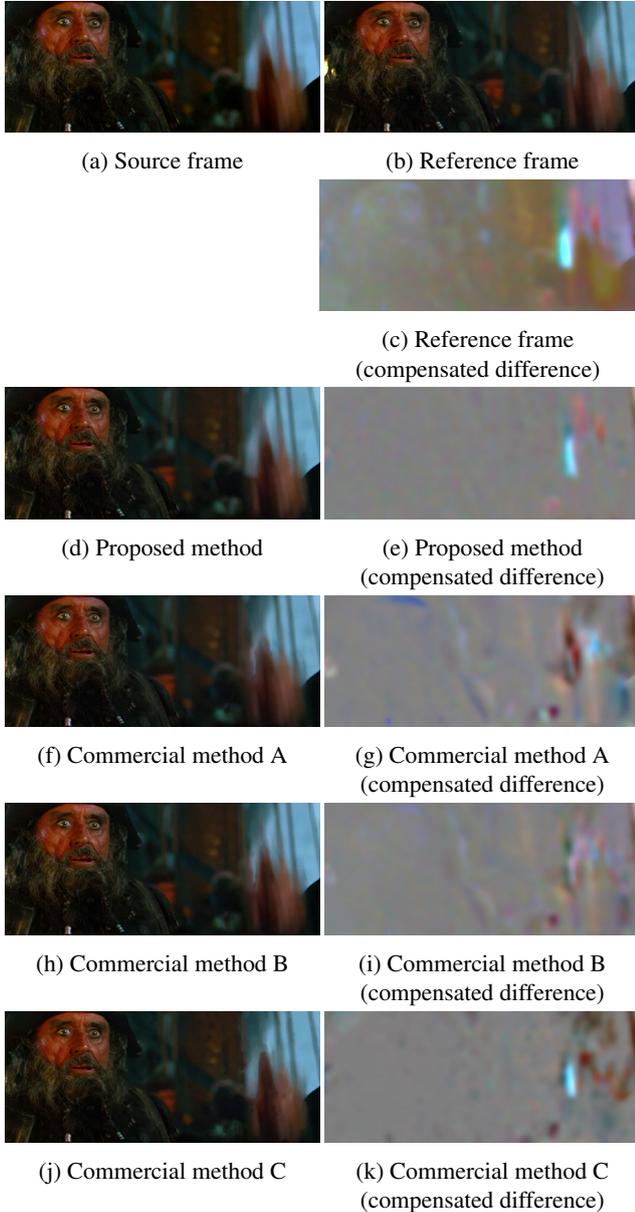


Figure 2: Example results for color-correction methods applied to data with natural distortions. The frame is from a trailer for *Pirates of the Caribbean: On Stranger Tides*.

We compared our proposed method with analogs from commercial software [10, 11]. In addition, we measured the computation time of each method. The final results appear in Table 1. On the basis of this data, we conclude that all of the methods deliver approximately the same accuracy for correcting color distortions, but our method better preserves the image structure, has better subjective quality, and is less computationally expensive than the others. Figure 2 shows a color-correction example.

## 5. CONCLUSION

In this paper, we presented a novel method for correcting color differences between views in stereoscopic videos. We developed a cost function that enables stereo matching of images with color distortions. We also proposed a new color-distortion model that employs stereo-matching confidence maps for better parameter evaluation. We compared our proposed method with commercial analogs and showed it accurately preserves the image structure and has a low computational cost.

## 6. REFERENCES

- [1] Marc Lambooi, Wijnand IJsselsteijn, and Ingrid Heynderickx, “Stereoscopic displays and visual comfort: a review,” *Eindhoven University of Technology Eindhoven University of Technology Philips Research, SPIE Newsroom* 2 April, 2007.
- [2] “VQMT3D (Video Quality Measurement Tool 3D) project page,” <http://www.compression.ru/video/vqmt3d/>, accessed: 2018-03-13.
- [3] Jean-Claude Rosenthal, Frederik Zilly, and Peter Kauff, “Preserving dynamic range by advanced color histogram matching in stereo vision,” in *3D Imaging (IC3D), 2012 International Conference on*. IEEE, 2012, pp. 1–6.
- [4] François Pitié, Anil C Kokaram, and Rozenn Dahyot, “Automated colour grading using colour distribution transfer,” *Computer Vision and Image Understanding*, vol. 107, no. 1-2, pp. 123–137, 2007.
- [5] Xuezhong Xiao and Lizhuang Ma, “Color transfer in correlated color space,” in *Proceedings of the 2006 ACM International Conference on Virtual Reality Continuum and Its Applications*. ACM, 2006, pp. 305–309.
- [6] Erik Reinhard, Michael Adhikhmin, Bruce Gooch, and Peter Shirley, “Color transfer between images,” *IEEE Computer Graphics and Applications*, vol. 21, no. 5, pp. 34–41, 2001.
- [7] Jung-Jae Yu, Hae-Dong Kim, Ho-Wook Jang, and Seung-Woo Nam, “A hybrid color matching between stereo image sequences,” in *3DTV Conference: The True Vision-Capture, Transmission and Display of 3D Video (3DTV-CON), 2011*. IEEE, 2011, pp. 1–4.
- [8] Ran Shu, Ho-Gun Ha, Dae-Chul Kim, and Yeong-Ho Ha, “Integrated color matching using 3D-distance for local region similarity,” in *Color and Imaging Conference*. Society for Imaging Science and Technology, 2013, vol. 2013, pp. 221–226.
- [9] Heiko Hirschmuller and Daniel Scharstein, “Evaluation of stereo matching costs on images with radiometric differences,” *IEEE transactions on pattern analysis and machine intelligence*, vol. 31, no. 9, pp. 1582–1599, 2009.
- [10] “Ocula,” <https://www.foundry.com/products/ocula>, accessed: 2018-05-16.
- [11] “YUVsoft Color Corrector,” <http://www.yuvsoft.com/products/stereo-processing-suite-lite/color-corrector/>, accessed: 2018-05-16.