COLOR STABILIZATION FOR MULTI-CAMERA LIGHT-FIELD IMAGING

Olivier Vu Thanh[†] *Trevor Canham*[⋆] *Javier Vazquez-Corral*[⋆]
Raquel Gil Rodríguez[§] *Marcelo Bertalmío*[⋆]

[†] INP Grenoble, Grenoble, France
[⋆]Universitat Pompeu Fabra, Barcelona, Spain
[§]Justus-Leibig Universität, Giessen, Germany

ABSTRACT

By capturing a more complete rendition of scene light than standard 2D cameras, light-field technology represents an important step towards closing the gap between live action cinematography and computer graphics. Light-field cameras accomplish this by simultaneously capturing the same scene under different angular configurations, providing directional information that allows for a multitude of post-production effects. Among the practical challenges related to capturing multiple images simultaneously, a very important problem is the fact that the different images do not perfectly match in terms of color, which severely complicates all further processing. In this work we adapt and extend to the light-field scenario a color stabilization method previously proposed for standard multi-camera shoots, and demonstrate experimentally that it provides an improvement over the state-of-the-art techniques for light-field imaging.

Index Terms— light field imaging, color stabilization, color matching, camera post-processing

1. INTRODUCTION

Light-field imaging is a promising new capture platform which adds an additional two dimensions of directional information to the 2D scene projection provided by standard capture methods. This additional data can be taken advantage of through post processing schemes which provide a number of relevant scene enhancements and reconstructions. These include but are not limited to post-capture re-focus, change of viewpoint, 3D data extraction, change of focal length, super-resolution, as well as any other practical application which could benefit from a more complete sampling of scene light information. However, with the introduction of this extra information comes additional challenges related to the high quality capture, storage, manipulation, transmission, and display of assets.

One of the currently open problems in light-field capture is that of the homogenization of color rendering between various directional capture elements. In all physical light-field capture architectures, views of the scene are captured from various angles. This can be accomplished via the placement of a lenslet array on a single sensor, the use of multiple independent cameras arranged in an array, or by the physical shifting of the camera point of view between exposures. In all cases, variations can occur in the color rendering of the scene between views, requiring methods which are capable of equalizing them. To avoid extending the labor intensive light-field capture process, a method should be proposed which accomplishes this homogenization without additional camera calibration steps or user input in post processing.

A pioneering work in color stabilization was presented by Hacohen *et al.* in [1], where the authors introduced a model to compute dense correspondences between the images, and combined it with a global color mapping model. Vazquez-Corral and Bertalmío [2] took inspiration from the color camera processing pipeline and presented an algorithm that estimates a power-law for each image and a matrix that converts one of the images into the other. This model was later extended for the case of logarithmic encoded images [3, 4, 5]. Also, following a similar line, Frigo *et al.* [6] presented a method to color stabilize video sequences in which they estimated a non-linearity and a channel-based scaling.

None of the previous approaches, however, were tested for light-field data. To the best of the authors' knowledge, the only work dealing with the color correction of light-fields is the one of Matysiak *et al.* [7] where a full pipeline for lenslet light-field image enhancement is proposed. In that work, the authors propose to use a Gaussian Mixture Model for color stabilization, as explained in [8, 9]. The authors also propose three different propagation schemes for the case of light-field images. The first scheme aims at stabilizing all the images towards the center one. The second stabilizes all the images of the center column towards the center one, and then the rest of images of each row are color stabilized with respect the center column. The third scheme is just an average of the previous two.

The contribution of this paper is to introduce an effective color-stabilization method specifically tailored to the light-field case, and that outperforms the state-of-the-art in light-field imaging. Our method adapts and extends to the light-field scenario the color stabilization method of [2], as will be explained next.

2. PROPOSED COLOR STABILIZATION METHOD

This section consists of three parts. We will start by recapping the technique of [2], which was formulated and validated for two-camera shooting scenarios. Secondly, we explain how we modify the optimization process of [2] and we will propose a spreading method for applying the color stabilization technique, where the idea is to color stabilize the images in a concentric manner starting from a reference in the center of the camera array. And finally, we will introduce a novel approach that avoids the problems that the original method presents in colors close to the gamut boundary.

2.1. Color stabilization for two cameras

In [2], the color processing pipeline of a digital camera is approximated by

$$\begin{bmatrix} R \\ G \\ B \end{bmatrix}_{out} = \left(A \cdot \begin{bmatrix} R \\ G \\ B \end{bmatrix}_{in} \right)^{\frac{1}{\gamma}} \tag{1}$$

where A is a 3 × 3 matrix representing both white balance and color encoding, RGB_{in} is the input raw triplet and γ characterizes a power law function.

Let us now suppose we have two different cameras. Then, according to the previous equation, the outputs (RGB_1 and RGB_2) of the two cameras for the same pixel of the scene RGB_{in} will be

$$\begin{bmatrix} R \\ G \\ B \end{bmatrix}_{1} = \left(A_1 \cdot \begin{bmatrix} R \\ G \\ B \end{bmatrix}_{in} \right)^{\frac{1}{\gamma_1}} ; \begin{bmatrix} R \\ G \\ B \end{bmatrix}_{2} = \left(A_2 \cdot \begin{bmatrix} R \\ G \\ B \end{bmatrix}_{in} \right)^{\frac{1}{\gamma_2}} \tag{2}$$

Then, solving for RGB_{in} we obtain

$$\begin{bmatrix} R \\ G \\ B \end{bmatrix}_{1}^{\gamma_1} - H \begin{bmatrix} R \\ G \\ B \end{bmatrix}_{2}^{\gamma_2} = 0 \tag{3}$$

where $H = A_1 {A_2}^{-1}$. Therefore H represents the color transformation between the linearized version of both images.

Let us now consider that we have obtained a set of correspondence points between the two images. Each of these correspondences will provide us with an equation like Eq. 3, therefore providing an over-determined system. Then, the three unkowns of the equation H, γ_1 and γ_2 can be inferred by minimizing the error of Eq. 3.

Once H, γ_1 and γ_2 are found, they can be applied to the whole image. In this sense, let us define the images I_1 and I_2 as $3 \times N$ matrices where N is the number of pixels. Then, if our goal is to stabilize I_2 according to I_1, we apply the following equation

$$I_2^{'} = (H \cdot I_2^{\gamma_2})^{\frac{1}{\gamma_1}} \tag{4}$$

where $I_2^{'}$ is the color stabilized image.

2.2. Optimization modifications, center and spreading method

Let us note that in a light-field pipeline, the camera output is likely to be encoded linearly. For this reason, in this specific situation the gamma of the reference image γ_1 (image 1 acts as the reference in the previous explanation) is fixed to unity. For non-linear adjustment purposes, γ_2 will still be approximated during the optimization process. Another modification with respect to [2] is that here the optimum values for γ_2 and H are found simultaneously instead of sequentially. Also, since our method aims at being included in a pipeline for light-fields, the RANSAC method proposed to remove outliers in [2] will not be used as it is too computationally expensive. Instead, outliers are detected considering their deviation from the whole cluster of correspondences.

The initial natural idea for applying the color stabilization proposed by [2] to a light-field would be to take the center of the light-field as a reference. Then, each view would be color stabilized according to the reference view. The problem is that if a view is too far from the reference there will be less correspondences than if the view and the reference were neighbours. To counter this, a second natural idea would be to spread the color stabilization in the light-field in a concentric manner, as in Figure 1. The color stabilization starts at the center. Each color stabilized view will be considered as a reference for the nearest view in the upper circle of views. In return for an increased number of correspondences, the spreading method brings the risk of propagating correction error. Let us note that both the center and spreading method will need the same number of source/reference comparisons to be made.

2.3. On dealing with the gamut boundaries

A weakness of [2] is its lack of consideration for gamut boundary pixels. It comes from the fact the Eq. 1 is a good approximation of the core pixels of the image, but does not consider further processes such as gamut mapping. This leads to problems when some bright image regions become clipped after the in-camera processing, as the real color of the region is lost. A clear example of this is shown in Figure 2. In the source image (left) the over-exposed pixels in the blanket have been clipped. For this reason, the correction from the original method would introduce a color cast to these regions.

In this section we propose a solution for this problem. In particular, we propose to detect the over exposed pixels of the view and "smooth" the application of the original matrix H

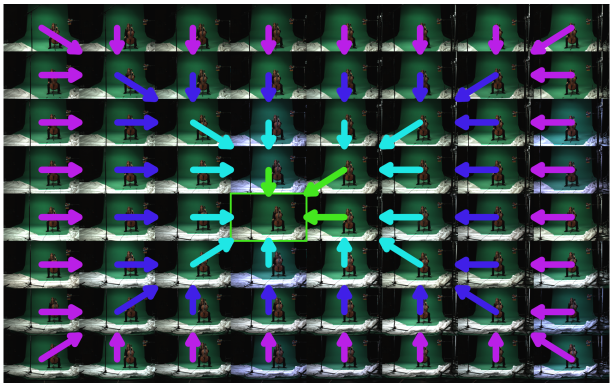

Fig. 1. Color stabilization with a spreading method. Each frame is color stabilized to a neighbour reference which has been color stabilized before.

(Eq. 3) on these regions.

To explain how this smoothing is done, let i be a pixel whose coordinates in L*a*b* color space are (L_i, a_i, b_i). In [10], the following equation is proposed to detect over exposed pixels :

$$\mathcal{M}_i = \frac{1}{2}\left(\tanh\left(\delta \cdot \left((L_i - L_T) + (C_T - C_i)\right)\right) + 1\right) \quad (5)$$

where C_i denotes the chroma component, i.e. $C_i = \sqrt{a_i^2 + b_i^2}$, L_T and C_T denote the boundary value of the overexposure region, and δ controls how fast $\mathcal{M}_i$ grows. As in [10], we will choose $\delta = 1/60$, $L_T = 80$ and $C_T = 40$.

Then, our smoothing applied to matrix H will depend on the value of $\mathcal{M}_i$. In order to determine in which range we should perform the smoothness, we found that $\mathcal{M}_i$ for RGB=(111) is 0.88. For this reason, we decided to start the smoothing of the matrix H from those pixels having $\mathcal{M}_i$ higher than 0.5.

Our process is as follows. We perform a cubic interpolation between H and a weighted identity according to $\mathcal{M}_i$. The weighted identity is found so as the maximum value of the source image corresponds to the maximum value of the reference image (i.e. a maxRGB transformation).

3. RESULTS

3.1. Scoring the results

Without a ground truth comparison, the objective measurement of light-field color stabilization poses a challenging problem. Since each view is taken from a different angle and has a different focus, it is not possible to directly compare them. Our proposed solution is to use the Macbeth ColorChecker present in our example case. In each view, each patch of the Macbeth ColorChecker will be localized with the ColorChecker finder proposed by [11]. A local mean of each patch is then computed and compared to the ColorChecker of the central view. We choose to use CIEDE2000 [12] as an indicator of the color difference. This gives a total of $N \times 24$ color difference values for a single multi-view frame, with N being the number of views minus 1 (for the central reference view). We then compute the mean, the median and the maximum of these values to provide a general score for the method.

We also propose in this paper a second measure that illustrates the general stability among different views. In particular, to compute the standard deviation among the views for each color checker patch and for each color channel:

$$\sigma = \sqrt{\frac{1}{N+1}\sum_{i=1}^{N+1}(x_i - \overline{x})^2} \quad (6)$$

where $\{x_1, x_2, \ldots, x_{N+1}\}$ are the measured values for one color patch and for one color channel in all the different light-field views. This provides us with 24×3 (24 patches and 3 color channels) standard deviations, that are later averaged to obtain the final results.

We took into consideration that the results might be influenced by the presence of correspondences in the Macbeth ColorChecker. For this reason, and in order to not influence our results, the color stabilization of our method is computed without any correspondences in the Macbeth ColorChecker. Also, let us note that the scaling ratio shown in Table 1 only concerns the views during the SIFT phase of [2]. The pixel correspondences are determined between down-scaled views, but resulting transformations are applied to the views in their original scaling.

Regarding the implementation of [7], we use the code the authors have provided online. In this case, the scaling is performed as with our method, i.e. the parameters are learned in the scaled image, but applied to the full size one. This method is allowed to consider color matches in the Macbeth ColorChecker.

3.2. Interpretation

Considering the results in Table 1, the results of the center and spreading methods are nearly the same. With this example case, the center method has slightly better results. The difference between these two methods are not likely to be perceptible. The results presented in Table 1 indicate that our method succeeds in stabilizing the color of the views according to the reference. Intuitively, the spreading method is likely more effective for low resolution and wide angle light-fields. In these two cases, the disparity between the views is greater so finding correspondences between adjacent views may be more relevant. Let us finally note that the non-monotonic evolution of the color differences for our method as a function of image scaling is probably due to SIFT finding different correspondences.

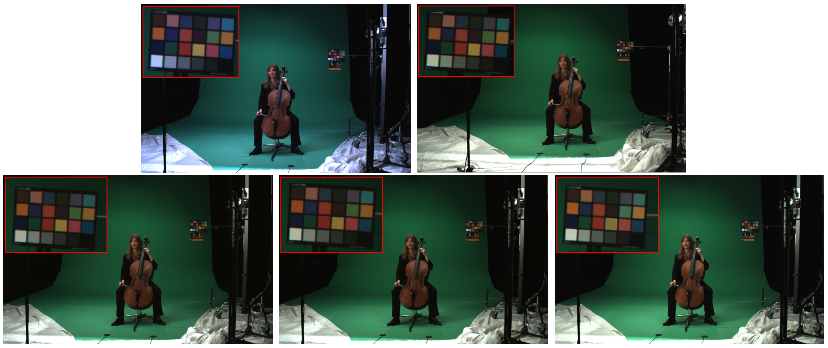

Fig. 2. Top: source (left) and reference (right), with zoomed-in detail of color chart. Bottom, from left to right: proposed method (reference in the center), proposed method (spreading reference), result of [7]. It can be seen that the proposed method makes a better match to the reference than [7] does, particularly in its rendering of saturated colors.

	resizing	20 %	40 %	60 %	100 %
Ours-center	Mean	1.16	1.09	1.16	0.91
	Med	0.87	0.88	0.89	0.75
	Max	8.07	6.95	8.27	6.83
Ours-spreading	Mean	1.26	1.18	1.35	1.18
	Med	1.01	0.94	1.10	0.94
	Max	8.55	7.35	8.24	7.41
Matysiak et al.	Mean	8.50	7.18	2.48	1.53
	Med	2.87	2.09	1.42	1.13
	Max	65.33	65.28	36.18	25.33

Table 1. Color differences with different resizings in terms of ΔE_{00}. The smaller the better.

We show the results for the stability measure in Table 2. In there, we can see that the results for a resizing of 20%, 40% and 60% are better for the spreading method than for the center method. This result supports the hypothesis that the spreading method is superior at decreasing color fluctuations among the views when the light-field's resolution is low.

From both Tables, we can see that the consistency of color difference values as a function of resize ratio indicates that our method can be optimized in terms of computation time -by reducing the size of the images- without noticeable differences on the quality of the color stabilization on the light-field.

Furthermore, by looking at the results in Table 1 and in Table 2, we can conclude that our method outperforms the method proposed in [7] in terms of overall error and consistency across the multi-view array. Regarding implementation, our method also outperforms the one proposed in [7] in terms of processing time. While the method of [7] takes in the order of hours to obtain the results for the full size images in a Intel® Core^{TM} i5-8400 CPU @ 2.80GHz 6 with 16GB of ram, our method is computed in less than 5 minutes.

resizing	20 %	40 %	60 %	100 %
Ours-center	0.0425	0.0211	0.0290	0.0092
Ours-spread	0.0124	0.0120	0.0237	0.0322
Matysiak et al.	0.0884	0.0916	0.0670	0.0136

Table 2. Stability results with different resizings. The smaller the better.

4. CONCLUSIONS

We developed a quick and reliable method to decrease the color disparity that might occur among the views in a light-field. Since our algorithm is fast and can export the outputs as 3D LUTs, it can easily be integrated into a practical light-field pipeline. In addition, the method's sole reliance on common scene content between views allows it to function without additional on-set calibration steps. While only a limited validation was conducted here, the method should be tested with a data set featuring a greater number of capture platforms and scene content moving forward. Of particular interest are scenes with wider color variation and high dynamic range. Additionally, the center and spreading methods should be further tested with light-field data containing larger scene content or lighting disparities between central and outer views.

5. ACKNOWLEDGEMENTS

This work has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement number 761544 (project HDR4EU) and under grant agreement number 780470 (project SAUCE), and by the Spanish government and FEDER Fund, grant ref. PGC2018-099651-B-I00 (MCIU/AEI/FEDER, UE).

6. REFERENCES

[1] Yoav HaCohen, Eli Shechtman, Dan B. Goldman, and Dani Lischinski, "Non-rigid Dense Correspondence with Applications for Image Enhancement," *ACM Transactions on Graphics*, vol. 30, no. 4, pp. 70.1–70.10, July 2011.

[2] Javier Vazquez-Corral and Marcelo Bertalmío, "Color Stabilization Along Time and Across Shots of the Same Scene, for One or Several Cameras of Unknown Specifications," *IEEE Transactions on Image Processing (TIP)*, vol. 23, no. 10, pp. 4564–4575, Oct 2014.

[3] Javier Vazquez-Corral and Marcelo Bertalmío, "Log-encoding Estimation for Color Stabilization of Cinematic Footage," in *IEEE International Conference on Image Processing (ICIP)*, 2016, pp. 3349–3353.

[4] Raquel Gil Rodríguez, Javier Vazquez-Corral, and Marcelo Bertalmío, "Color-matching shots from different cameras having unknown gamma or logarithmic encoding curves," in *SMPTE Annual Technical Conference & Exhibition*, 2017.

[5] Raquel Gil Rodríguez, Javier Vazquez-Corral, and Marcelo Bertalmío, "Color matching images with unknown non-linear encodings," *IEEE Transactions on Image Processing*, pp. 1–1, 2020.

[6] Oriel Frigo, Neus Sabater, Julie Delon, and Pierre Hellier, "Motion Driven Tonal Stabilization," *IEEE Transactions on Image Processing (TIP)*, vol. 25, no. 11, pp. 5455–5468, Nov 2016.

[7] Pierre Matysiak, Mairéad Grogan, Mikaël Le Pendu, Martin Alain, and Aljosa Smolic, "A pipeline for lenslet light field quality enhancement," in *IEEE International Conference on Image Processing (ICIP 2018)*, 2018.

[8] Mairead Grogan, Rozenn Dahyot, and Aljosa Smolic, "User interaction for image recolouring using l2," in *14th European Conference on Visual Media Production*, London, UK, 2017.

[9] Mairead Grogan and Rozenn Dahyot., "L2 divergence for robust colour transfer," *Computer Vision and Image Understanding*, vol. 181, pp. 39–49, 2019.

[10] Dong Guo, Yuan Cheng, Shaojie Zhuo, and Terence Sim, "Correcting over-exposure in photographs," in *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, June 2010, pp. 515–521.

[11] Keigo Hirakawa, "Colorchecker finder," .

[12] Gaurav Sharma, Wencheng Wu, and Edul N. Dalal, "The ciede2000 color-difference formula: Implementation notes, supplementary test data, and mathematical observations," *Color Research & Application*, vol. 30, no. 1, pp. 21–30, 2005.