Pattern Recognition 154 (2024) 110575

Contents lists available at ScienceDirect

Pattern Recognition

journal homepage: www.elsevier.com/locate/pr

Color matching in the wild

Raquel Gil Rodríguez [a,1,*], Javier Vazquez-Corral [b,c,1], Marcelo Bertalmío [d], Graham D. Finlayson [e]

[a] Psychology Department, Justus-Liebig University Giessen, Giessen, 35394, Germany
[b] Computer Vision Center, Cerdanyola del Vallès, 08193, Spain
[c] Computer Sciences Department at Universitat Autònoma de Barcelona, Cerdanyola del Vallès, 08193, Spain
[d] Instituto de Óptica, Spanish National Research Council (CSIC), Madrid, 28006, Spain
[e] School of Computing Sciences, University of East Anglia, Norwich, NR4 7TJ, United Kingdom

ARTICLE INFO

Keywords:
Color stabilization
Color matching
Logarithmic encoded images
Gamma-corrected images
HDR encoding

ABSTRACT

We present a method that, given two different views of the same scene taken by two cameras with unknown settings and internal parameters, corrects the colors of one of the images making it look as if it was captured under the other camera settings. Our method is able to deal with any standard non-linear encoded images (gamma-corrected, logarithmic-encoded, or any other) without requiring any previous knowledge of the encoding.

To this end, our method makes use of two important observations. First, the camera imaging pipeline from RAW to sRGB can be well approximated by considering just a per-pixel shading and a color transformation matrix, and second, for correcting the images we only need to estimate a single matrix –that will contain information from both of the original images– and an approximation of the shading term (that emulates the non-linearity).

Our proposed method is fast and the results have no spurious artifacts. The method outperforms the state-of-the-art when compared with other methods that do not require knowledge of the encoding used. It is also able to compete with –and even surpass in some cases– methods that consider information about image encoding.

1. Introduction

Color matching is a well-known problem in color imaging, in which it is necessary to map the colors of one image, called the source, to those of a second image, the reference. Color stabilization, the focus of this paper, is a particular color matching problem. In color stabilization, the source and reference images are instances of the same scene, captured by different cameras or settings and with respect to different view points.

The pipelines of the different digital cameras are proprietary to the camera makers, and different cameras perform different operations (color characterization, white-balance, etcetera) in a different order. This said it is well-accepted [1] that a good approximation for the image values of the core pixels (in fact almost all pixels save those that are very close to the border of the color gamut) is

$$I_{out} = NL\left(I_{lin}A\right), \tag{1}$$

where I_{lin} is the linear image read by the camera sensor after demosaicing, and I_{out} is the output image. Here both I_{lin} and I_{out} denote $n \times 3$ matrices, where n is the number of pixels, A is a 3×3 matrix which carries color information and white balance and the value NL defines a non-linearity (for Low-Dynamic Range images a power-law-gamma correction- and for High-Dynamic Range image either a logarithmic encoded curve or some standard such as HLP or PQ are used). Of course, this is a simplified version of the pipeline, as some other processing techniques, including denoising or contrast enhancement, are also applied. Yet, Eq. (1) is a reasonable approximation and is employed and deployed in different color stabilization methods, including [2–4].

Importantly, Finlayson et al. [5] recently proposed another approximation for the camera processing pipeline based on color homographies. That approach teaches us that we can also approximate the camera processing pipeline as

$$I_{out} = S\,I_{lin}A, \tag{2}$$

where S is a diagonal matrix of size $n \times n$ that represents a pixel-wise shading adjustment, and A, I_{lin} and I_{out} are as previously defined. Broadly, deploying either Eqs. (1) or (5) will well describe the mapping from linear raw to the non-linear final rendered image. That is, in terms of fitting error, the two methods perform equally well.

[*] Corresponding author.
E-mail address: raquel.gil-rodriguez@psychol.uni-giessen.de (R. Gil Rodríguez).
[1] These two authors contributed equally to this work.

https://doi.org/10.1016/j.patcog.2024.110575
Received 5 February 2024; Received in revised form 3 May 2024; Accepted 6 May 2024
Available online 8 May 2024

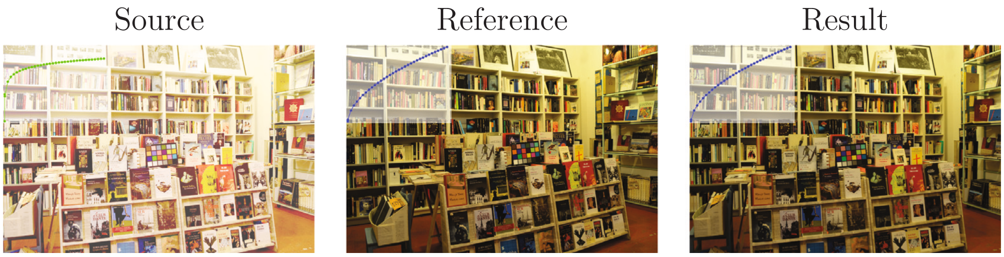

Fig. 1. Schematic of our problem and solution. Given a source image captured under an unknown encoding curve (left) and a reference image of the same scene captured by a different camera and encoded with a different non-linearity (middle), our goal is to modify the source image so it matches the colors and non-linearities of the reference one (right).

In this paper, we present a method that is able to color-stabilize any two images of the same scene that were captured under any combination of color matrices and non-linearities without using any prior information on how the images were processed. Both images do not require to be registered, they just require to have some common region –for example, two views of the same scene that are close enough–. We show how to combine the knowledge given by Eqs. (1) and (5), and how to model this combination. To the best of our knowledge, ours is the first paper to tackle the color stabilization problem *in the wild*. Indeed the prior art [2,6] has mainly focused on the case where images are simply gamma-corrected cases. Moreover, both gamma-encoded and logarithmic-encoding images have been considered [3,4], but knowing the type of encoding being used was required to use those methods. A schematic idea of our problem is shown in Fig. 1.

The paper is organized as follows. In the next section, we discuss the related work. Then, in Section 3 we present our method. After that, we present the different experiments carried out and their results. The paper ends with a short conclusion.

2. Related work

Matching images to make them more color consistent is an important problem encountered both in industry and academia.

In industry, color consistency is, for the most part, assured by skilled manual work; either by colorists performing color grading of movies in post-production houses or by technicians that use camera control units (CCU) [7] in live TV broadcasts. In both cases, these skilled technicians may require either having the cameras properly characterized –for example, by using the ACES framework [8]– or the inclusion of color-charts in the shots.

In academia, two related but slightly different approaches have been used for solving the problem of the color matching images. First, there is the general color transfer problem. Here, we have two images, a source and a reference, and we would like them to match in terms of their color distributions. Posed in the most abstract form the content of the images –i.e. what is in the scene being imaged– can be completely different between the image pair. As an example, we may transfer the colors from one image of a sunset scene (the reference image) to another (of somewhere shot midday, the source image) in the hope of making the second source image look like it was captured at sunset.

The color stabilization problem can be thought of as a special case of color transfer. In color stabilization, the image pair –to be color matched– is acquired for the same scene, and so the underlying content captured in each image is similar. However, in camera pipelines the rendition of a pair of images can change a surprising amount even when they are captured close to each other in time. That is small changes in the content of the images can result in large changes in the color rendition. In color stabilization we would like the color matched image (say the second source image matched to the first reference) to look similar to the first and in particular the colors of the same semantic regions of the scene should almost be the same.

Reinhard et al. [9] presented a seminal work on the color matching problem, where they proposed to first convert both source and reference images to a decorrelated color space, and then transfer the mean and the variance from the reference image to the source one. Pitié et al. [10,11] went one step further and defined the images as probability density functions. Elegantly, in Pitié et al.'s work, the color matching problem was shown to be a particular case of optimal transport (OT), in which the goal is to minimize the cost of transferring probability density distributions of the source image into the reference one.

This idea of using optimal transport as a way of performing color matching was later also considered by Rabin et al. [12] and Ferradans et al. [13]. More recently, Grogan and Dahyot have defined a new model following L2-divergence that shares some components with OT cost function [14]. Other color transfer methods include, Kotera [15] where it was proposed to match the principal components of the two images by rotation and scaling, Pouli and Reinhard [16] in which the authors performed a multi-scale histogram matching, or Xiao and Ma first proposed in [17] to consider different color statistics, and later on in [18] proposed a method that is gradient preserving together with a new evaluation metric for color transfer.

Nguyen et al. [19] proposed to split color constancy and color transfer by first applying color constancy to the input, followed by luminance matching and a final alignment of the color gamut via a linear transformation. Hwang et al. [20] introduced a moving least-squares approach, in which a probability measure was introduced to guarantee robustness against both noise and outliers. Gong et al. [21,22] proposed to color transfer the colors using a projective transformation and a mean intensity mapping. Recently, some methods aiming at photorealistic color transfer based on deep learning architectures have also appeared [23–26]. Still on deep learning, there are new topics that might be applied in the future to this problem, such as distilling datasets in basis and hallucination for efficiency reasons –opening the way to color correct only the basis images [27]–, or knowledge factorization [28] aiming at specializing larger networks to specific sub-tasks.

All the methods mentioned up until now are global, but some local methods also exist. For example, Tai et al. [29] proposed to segment images into different regions, to represent their distribution using Gaussian Mixture Models (GMM) to represent color distributions, and finally match these distributions. Similarly, Xiang et al. [30] also considered first a GMM representation of color areas to later match them.

Color stabilization focuses on the case in which some regions or objects appear in both the reference and the source images, and therefore it is possible to use this extra information. HaCohen et al. [6] presented the first work on this research line, where they propose to compute dense correspondences between the images, combined with a global color mapping model. Vazquez-Corral and Bertalmío [2] proposed a color stabilization algorithm built on the observation that in digital cameras the color encoding can be approximated as a matrix multiplication followed by a power law (gamma correction). Following this observation, they show how to color match the source image to the reference, it is enough to estimate a power law (γ value) for each

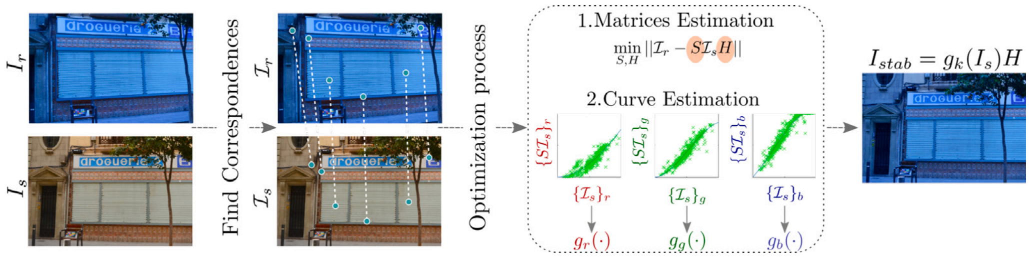

Fig. 2. Flowchart of the proposed method. Given two images, a reference I_r and a source I_s, we first compute a set of correspondences between them, and we denote the pixel values at those correspondences as $\mathcal{I}_s$ and $\mathcal{I}_r$, respectively. Within the optimization process, we estimate the two matrices S and H from Eq. (8), and then we estimate the curve $g(\cdot)$ for each color channel. The final color stabilized image is then computed as $g_k(I_s)H$, $k = \{R, G, B\}$.

of the images, and a single 3 × 3 matrix. Frigo et al. [31] proposed a method focusing on color stabilizing video sequences. This method is based on estimating a non-linearity and channel-based scaling.

To the best of the authors' knowledge, there are only two color stabilization works for logarithmic images. One is the method of Vazquez-Corral and Bertalmío [3], which relies on finding a sufficiently large number of achromatic matches among source and reference. This need to detect achromatic matches may be a challenging limitation, in some cases. The other method [4] builds from the method in [2]. In short, it converts the logarithmic-encoded image into a power-law representation to be able to apply the [2] but also improves the quality of the estimation by allowing the matrix to be a projective 4 × 4 transform. A limitation of both of these methods is that they both need to know how the images are encoded (logarithmic vs. gamma) in order to obtain good results. Moreover, as images are also tone mapped for preference neither the gamma or logarithmic model will account for image encoding in all cases.

Color consistency differs from color stabilization in the sense that in color consistency there are not only two but a set of images from the same scene that need to be color matched. HaCohen et al. [32] extended their color stabilization approach [6], to the case of having more than two images. Later, Park et al. [33] proposed a model in which they estimate two different parameters for the matching: a gamma correction and a white balance constant. Finally, and related to color consistency, different color transfer methods have also been considered for image stitching (see Xu and Milligan [34] for a review). Xia et al. [35] the authors compute parametric curves for each channel under color, gradient and contrast constraints on the overlapping regions.

The method presented in this paper draws from the sources of [2,4], combining these earlier works with the Color Homographies [5]. We will make the simplifying assumption that Eqs. (1) and (5) are, more or less, equivalent –a reasonable one to make as adopting either method accounts for real camera data well [5]– and, in so doing, will use them interchangeably.

3. Our approach

3.1. General idea

Let us start by supposing that we have two rendered images, I_r and I_s –under-script referring to reference and source–, that are broadly of the same scene content and so they have a set of n points $(p_1, \ldots, p_n)$ from the scene in common. Then we can define $\mathcal{I}_r$ and $\mathcal{I}_s$ as the $n \times 3$ matrices corresponding to the RGB values in each of the images for the n corresponding points.

In [2], Vazquez-Corral and Bertalmío stated -following [1]- that the relationship between the RAW image and the two rendered images for the indexed pixels can be well-approximated as:

$$\mathcal{I}_r = NL_r\left(\mathcal{I}_{in} A_r\right)$$
$$\mathcal{I}_s = NL_s\left(\mathcal{I}_{in} A_s\right), \tag{3}$$

Here, the $n \times 3$ matrix $\mathcal{I}_{in}$ represents the RAW image at the points -$(p_1, \ldots, p_n)$-, A_i are 3 × 3 matrices and NL_i are non-linearities (including a power-law, a logarithmic encoding or a tone mapping applied for enhancement). Substituting, we can write:

$$\mathcal{I}_r = NL_r(NL_s^{-1}(\mathcal{I}_s)H), \tag{4}$$

where $H = A_s^{-1} A_r$.

In previous work, the form of Eq. (4) was used to discover the unknown model parameters (NL_1, NL_2 and H) and these parameters were applied to the whole I_s image to obtain the color stabilized output. In [2,4] the corresponding points in the reference and source images are found for using SIFT. Although we chose SIFT due to its simplicity, other matching methods can also be utilized. Then, Eq. (4) is recast as an objective function with the model parameters found to minimize a mean-square error. A necessary constraint in their model is that they need to know the form of the non-linearity. That is, it is assumed to be a gamma or a logarithmic encoding.

In this paper, we will instead use the Homography approximation to the camera processing pipeline given by Finlayson et al. [5] to avoid placing any constraint on the form of the camera non-linearity. Using the same notation as in the previous equations, Finlayson et al. teaches us that

$$\mathcal{I}_r = S_r \mathcal{I}_{in} A_r$$
$$\mathcal{I}_s = S_s \mathcal{I}_{in} A_s, \tag{5}$$

where S_i is a diagonal matrix of size $n \times n$ that represents a pixel-wise shading. Note this model is in actuality approximate (but it is convenient for modeling purposes to assume equality). Importantly, modeling on a large set of real cameras indicates that it fits –accounts for a real camera encoding– as well as Eq. (3). In reality, both representations model real camera data with small errors.

Using Eq. (5), we can rewrite Eq. (4) as:

$$\mathcal{I}_r = S_r S_s^{-1} \mathcal{I}_r A_s^{-1} A_r. \tag{6}$$

It follows that

$$\mathcal{I}_r = S\mathcal{I}_s H, \tag{7}$$

where $H = A_s^{-1} A_r$ (as in Eq. (4)) and $S = S_r S_s^{-1}$.

When two images (or sets of pixels thereof) are related according to Eq. (7) we say that the two images are a *shading* homography apart (see [5] for more details about this term).

With respect to the homography model of image encoding, a rendered image is a shading homography from its linear raw counterpart. Given two images of the same scene encoded differently (the problem at hand in color stabilization), these pair of rendered images must be related by a shading homography (this is the import of Eqs. (5) through (7)).

In other words, we can see that S in Eq. (7) is performing the same type of operation as the combination of the two non-linearities in Eq. (4). We can therefore use S -that is a diagonal, $n \times n$ matrix- to obtain the effect of the non-linearities on the images.

```
Inputs: The reference and the source images $I_r$ and $I_s$.
Output: The color stabilized image $I_{stab}$.
Obtain the $m$ corresponding points for $I_r$ and $I_s$ (e.g. using SIFT) ;
From those corresponding points:
    • Estimate $S$ and $H$ from Eq. (8).
    • Estimate the global non-linearity $g(\cdot)$ from S using Eq. (9).
$I_{stab} = g(I_s)H$.
```

Algorithm 1: Algorithm for our color stabilization approach

Let us rewrite Eq. (7) as an objective function to be minimized to find the best shading terms and the best color correction matrix:

$$\min_{S,H} \|\mathcal{I}_r - S\mathcal{I}_s H\|. \tag{8}$$

Of course, once we have related the two images (e.g. for the n SIFT corresponding points) we need to apply the learned relationship to all the pixels in the source image. Thus, once we obtain H and S, we need to be able to propagate their use to all the pixels in the image. This is trivial for the matrix H as we can directly use it for all the image pixels. However, the matrix S is tied to the pixels at hand (i.e. the n sift points) so cannot be used directly.

Instead, we fit a global curve $g(\cdot)$ that better estimates its behavior for the set of pixels, and we apply that curve to the whole image. The estimation of this curve is performed as

$$\min_{g} \|S\mathcal{I}_s - g(\mathcal{I}_s)\|. \tag{9}$$

Finally, after the fitting of $g(\cdot)$, we carry our a final refinement of H:

$$\min_{H} \|\mathcal{I}_r - g(\mathcal{I}_s)H\|. \tag{10}$$

A detailed schematic of our method is presented in Fig. 2, see also Algorithm 1.

3.2. Implementation

A minimum for Eq. (8) is found by using the (ALS) Alternating Least Squares Method [36]. The ALS method guarantees the finding of a local minima. It iteratively solves for S (holding H fixed) and then H (holding S fixed) until convergence. The ALS method has a number of desirable properties. It is easy to implement, the individual steps are least-squares optimal and the algorithm converges quickly. A detailed explanation of the ALS algorithm is presented in the appendix but also see [36].

Our minimization returns H and S from Eq. (7). But, although H can be directly applied now to the whole image, this is not the case for S. In order to be able to apply S to the full image, we approximate its behavior by solving for a global curve $g(\cdot)$.

The global curve g is computed by fitting a polynomial of order $r-1$ to the data points estimated from S using least squares. Let us denote the jth row of SI_s, $[SI_s]_j = [R_j\ G_j\ B_j]$. Similarly, the corresponding point in the non-shading is denoted $[I_s]_j = [\hat{R}_j\ \hat{G}_j\ \hat{B}_j]$. Let us adopt the notation $q_{k,j}$ and $\hat{q}_{k,j}$ to denote the R, G and B of the jth row in respectively SI_s and I_s ($k \in \{R, G, B\}$). We would like a polynomial expansion of I_s to approximately equal SI_s:

$$q_{k,j} \approx \beta_0 + \beta_1 \hat{q}_{k,j} + \cdots + \beta_r \hat{q}^r_{k,j} \tag{11}$$

for fixed coefficients β_i, $i = 1, 2, \ldots, r$.

Let us now consider how to find the optimal polynomial coefficients. Let the $((k-1)j + j)$th row of an $kN \times r + 1$ matrix A contain the row vector $[\beta_0\ \beta_1\hat{q}_{k,j}\ \cdots\ \beta_r\hat{q}^r_{k,p}]$ and the corresponding $((k-1)j+j)$th element of an $kN \times 1$ vector $\underline{b}$ equal $q_{k,j}$. We would like

$$\underline{b} \approx A\underline{\beta}, \tag{12}$$

We solve for $\underline{\beta}$ in the least-squares sense using the Moore–Penrose inverse. It is these coefficients that define the function $g()$. For our purposes, we found we could fit our data well with $r = 4$ (3rd order polynomial expansion). An important detail is that the curve $g()$ should be monotonic. For our regression this was found, naturally, to be the case. But, monotonicity can also be explicitly enforced by rephrasing the regression as a constrained optimization.

We consider multiple configurations for H and S,

1. H as a 3 × 3 or 4 × 4 matrix: As it was explained in [2], for those pixels that fall inside the core of the gamut of the color space, a linear 3 × 3 matrix suffices for its rendering. However, as it was shown in [4] (and following [5]), the use of a projective 4 × 4 matrix allowed us to obtain better results; as it was able to better correct those pixels in the gamut boundary. Therefore, in this paper, we have decided to try both of these possibilities and compare their results.
2. S as a 1-D or 3-1D curve: In this case, we study how to perform the approximation of S by the global curve $g(\cdot)$. This global curve can be learned in what we call 1D, i.e. we learn the same curve for the three color channels. This case is similar to the one of the 3 × 3 color matrix: It suffices for pixels inside the gamut [2]. However, in order to better deal with pixels in the gamut boundary, that are affected by different operations such as gamut mapping or sharpening, we also introduce a 3 × 1D version for the global curve approximation of S. In this case, we do not obtain a single curve $g(\cdot)$, but one for each of the color channels, i.e. $g_k(\cdot)$ with $k = R, G, B$.

4. Experiments and results

4.1. Methods considered

In this paper, we evaluate our method against those used in [4]: we consider eight state-of-the-art methods for color transfer, stabilization and consistency, Reinhard et al. [9] (Reinhard), Kotera [15] (Kotera), Xiao and Ma [17] (Xiao), Pitié et al. [11] (Pitie), Ferradans et al. [13] (Ferradans), and Park et al. [33] (Park). We also add four more recent methods, Grogan and Dahyot [14] (Grogan), and the deep-learning approaches of Chiu and Gurari [24] (Chiu), Yoo et al. [26] (Yoo), and Ho et al. [25] (Ho). All these methods are general in the sense that they share the common property that their computations –like our approach– do not require “metadata” e.g. knowledge of the form of any curve. We want to emphasize that for the Pitié et al. [11] approach, we focus only on the global part of their method and for Grogan and Dahyot [14], we selected the color transfer approach, no correspondences, using 50 colors for the k-means clustering. In common with the study [4], we do not include the method of Hacohen et al. because it was unable to solve all the different images.

Further, we also compare our approach with the methods of Gil Rodríguez et al. [37] (Gil18) and [4] (Gil20) that **do** require knowledge of the form of the non-linearity applied to the images. In this case, we perform a global comparison at the end of the section to assess how far are we from methods that input extra information.

4.2. Datasets

In our experiments we use the two datasets used in [4]. The **first dataset** consists of 35 image pairs where each pair is repeated 4 times for 4 different encodings. These are where the

1. source and reference are both gamma-corrected images.
2. source and reference are both logarithmically encoded
3. source image is logarithmically-encoded and the reference image is gamma-encoded.
4. source image is gamma-encoded and reference image is logarithmically-encoded.

Table 1
Results for the comparison of the 35 pairs in the two gamma encoded images case.

	ΔE_{00}		PSNR L		CPSNR		CID		RMSE	
	Mean	Median	Mean	Median	Mean	Median	Mean	Median	Mean	Median
Kotera	11.111	7.686	21.122	23.877	19.786	21.040	0.458	0.394	0.145	0.089
Pitie	3.567	3.394	26.162	25.946	25.696	25.769	0.174	0.157	0.055	0.051
Reinhard	4.777	4.652	25.525	25.162	23.904	23.571	0.205	0.191	0.068	0.066
Xiao	4.377	4.232	25.940	26.077	25.183	25.270	0.196	0.160	0.059	0.055
Ferradans	5.522	5.308	23.715	23.874	23.028	22.560	0.260	0.237	0.078	0.074
Park	3.428	3.020	27.604	27.381	26.595	26.384	0.157	0.134	0.051	0.048
Grogan	5.632	4.944	24.174	24.566	23.694	24.290	0.255	0.226	0.074	0.066
Chiu	4.600	4.433	23.853	24.023	23.490	23.591	0.227	0.222	0.071	0.067
Ho	11.487	12.664	21.034	20.552	19.743	20.077	0.519	0.586	0.128	0.117
Yoo	5.466	5.191	22.846	22.777	22.392	22.635	0.277	0.259	0.080	0.075
Proposed 3 × 3 1D	3.700	3.355	26.803	26.631	26.083	25.741	0.160	0.145	0.053	0.052
Proposed 4 × 4 1D	3.461	3.236	26.903	26.876	26.358	26.093	0.153	0.137	0.052	0.051
Proposed 3 × 3 3 × 1D	3.219	2.972	27.000	26.864	26.587	26.588	0.140	0.134	0.050	0.048
Proposed 4 × 4 3 × 1D	3.15	2.993	26.995	27.000	26.613	26.598	0.137	0.13	0.050	0.047

In this way, we are able to evaluate our method for a full set of different non-linearities, proving its generalization. As a cautious remark, let us note here that all the images appearing in the paper have been converted to sRGB for display purposes.

The **second dataset** used in [4] was made by ARRI. It contains HDR videos. From the videos, linear RAW data is obtained by using *ARRIRAW Converter* [38]. In [4] they select three different scenes, and for each scene, they set as a reference image the raw encoded with one of the 3 different options { PQ, HLG, Log C }. In total, there are 9 sets of pairs of images (3 scenes and 3 encodings). Additionally, there is an extra pair comparing two different Log C ARRI curves. Therefore, this dataset has a total of 10 image pairs.

4.3. Metrics

In our experiments, we are predicting the reference image by processing the source. Given the estimated and ground-truth reference images we compute 5 different metrics that quantify, in some way, the error between these images. The metrics we consider are

- *PSNR* for luminance channel (PSNR L),
- color *PSNR* defined as *CPSNR* is the mean of the PSNR of the three color channels,
- *root mean squared error* (RMSE),
- mean ΔE^*_{00} [39] color difference computed in the CIELab color space (ΔE_{00}),
- *CID* [40] is the color extension of SSIM [41].

For each metric, we compute the mean error and the median error for the 35 image pairs. We repeated the process for the second dataset as well, but this time after applying a gamma value of 2.2 to the linear images. This helped us to ensure that we could compare the same metrics for both datasets.

4.4. Results

The results are subdivided into the 4 different combinations of non-linearities present in the dataset.

4.4.1. Gamma to gamma

Results for this case are presented in Table 1. Here, and in the tables that follow, we show in green the best results, and then in respectively blue and yellow, the second and third best results. It is evident that, overall, the best performing algorithm is 4 × 4, 3 × 1D which means the color correction is a projective transform and there is an individual per channel tone curve. This algorithm has the 7 (of 10) best scores for the reported statistics (the mean and median of 5 metrics). The next best method again has per color channel non-linearities, but employs a 3 × 3 color matrix. For the case of PSNR L the method from Park et al. works best with our methods in 2nd and 3rd place.

Qualitative results for this case are shown in Fig. 3. In this Figure, we show from left to right the reference image, the source image, the ground-truth image, the result of processing the source with a competing method and the result of our processing. To re-emphasize, the source and reference images are of *similar* content to the ground-truth but some spatial detail is different (e.g. different camera position or zoom). The source image is in pixel registration to the ground truth. All the algorithms are trying to match the ground truth by learning the tone and color (transfer function) from relating the out of registration source to the reference.

In the first row of Fig. 3, notice that in the output from the method of Grogan et al. there is a clear pinkish cast. There is no cast either in the ground-truth or the output from our method. Notice also how much better we render the graffiti on the door. Regarding the images in the bottom row we see that the color on the floor and the wall are not properly restored by the method of Choi et al., in comparison, our method delivers good color stabilization. For each of these 2 examples, we also display the CID error (which is much lower for our method).

4.4.2. Log to log

Results are presented in Table 2. We see that our proposed method using 3 different curves (one per color channel) is able to improve on all the other methods both when the color matrix H is defined as 3 × 3 or 4 × 4 for all the metrics except the mean PSNR L (again Park's et al. method is best). Our best results are obtained when a 4 × 4 projective color mapping is used.

Qualitative results are shown in Fig. 4. From left to right: the reference, source, ground-truth images, the result of a competing method and our result. In the example in the first row of Fig. 4, we see that the output from Pitie's et al. algorithm has false colors, especially in the floor region. Regarding the bottom image, both our and Reinhard's et al. algorithm work quite well. Though, the color accuracy is significantly better for our method. Quantitatively, the CID metric numbers are consistent with our visual descriptions (our numbers are lower).

4.4.3. Gamma to log

In the previous two cases, both non-linearities were following the same non-linearity equation but with different parameters. In this subsection, this is no longer the case, as the source image is encoded using a gamma curve and the reference image is encoded using a logarithmic curve. Results for this asymmetric case are presented in Table 4. Save for the mean Delta E metric, our approach (per channel tone mapping and a projective color map) works best overall. For the mean Delta E score, the Pitie method is slightly better but the difference is small and is likely not visually significant. Evidently, the second best method is our proposed method where we use per color tone curves and a 3 × 3 color transform matrix.

Qualitative results for this case are shown in Fig. 5. Regarding the comparisons in the top row, colors from the Kotera algorithm are

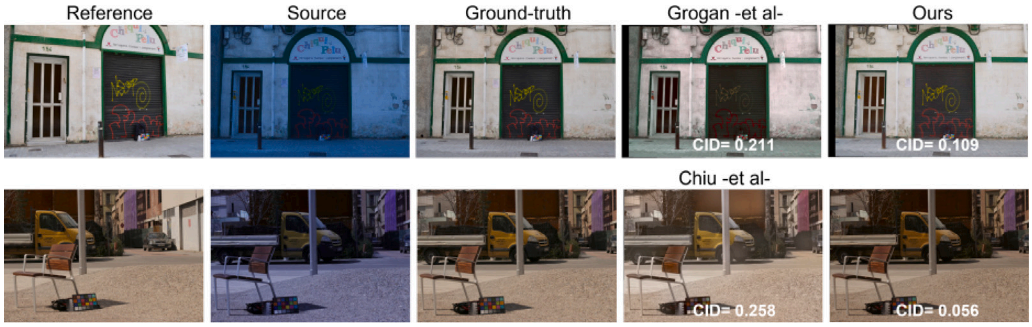

Fig. 3. Results for the Gamma to Gamma case. Please note the color of the wall and the color of the graffiti on the door in the top image, and the color of the floor and the wall in the bottom image. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Table 2
Results for the comparison of the 35 pairs in the two logarithmic encoded images case.

	ΔE_{00}		PSNR L		CPSNR		CID		RMSE	
	Mean	Median	Mean	Median	Mean	Median	Mean	Median	Mean	Median
Kotera	14.234	8.381	18.586	21.081	17.615	19.676	0.551	0.481	0.179	0.104
Pitie	3.978	4.044	25.797	25.369	25.119	25.099	0.207	0.201	0.059	0.056
Reinhard	7.878	7.916	22.656	22.512	19.899	19.369	0.364	0.368	0.107	0.108
Xiao	5.632	5.599	24.330	23.910	23.199	23.190	0.272	0.264	0.072	0.069
Ferradans	8.587	7.047	19.351	20.831	18.925	20.25	0.395	0.325	0.128	0.097
Park	6.768	4.548	26.217	26.162	23.961	24.196	0.296	0.210	0.083	0.062
Grogan	6.161	5.995	23.402	23.899	22.740	22.952	0.303	0.281	0.079	0.077
Chiu	5.314	5.397	22.614	22.605	22.232	22.068	0.278	0.284	0.082	0.081
Ho	12.137	11.590	19.121	19.676	18.221	18.128	0.548	0.597	0.146	0.152
Yoo	14.409	13.553	14.777	14.535	14.797	14.388	0.628	0.648	0.193	0.192
Proposed 3 × 3 1D	4.465	4.622	25.682	25.746	24.558	24.904	0.204	0.181	0.065	0.060
Proposed 4 × 4 1D	4.041	4.000	25.396	25.855	24.946	25.466	0.185	0.172	0.060	0.054
Proposed 3 × 3 3 × 1D	3.78	3.512	25.939	26.576	25.545	25.887	0.172	0.155	0.056	0.052
Proposed 4 × 4 3 × 1D	3.760	3.450	25.825	26.674	25.455	25.952	0.172	0.151	0.057	0.052

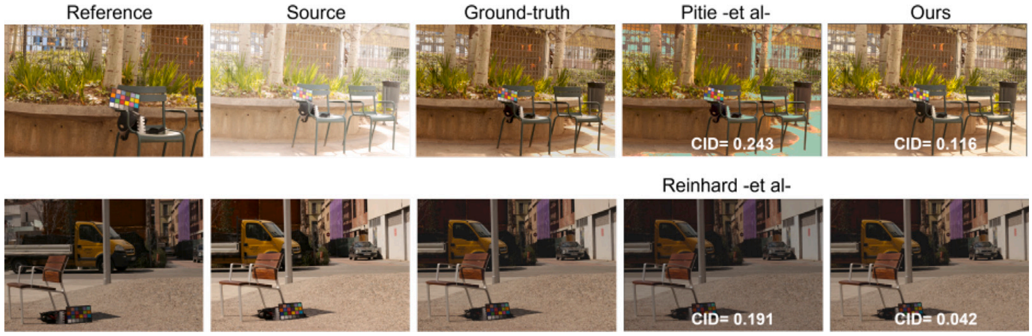

Fig. 4. Results for the case of two logarithmic encoded images. Please, focus on the general look and the color checker in the top image. Also, please look at the pink color on the wall, the yellow color of the truck, and the color of the floor in the bottom image. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

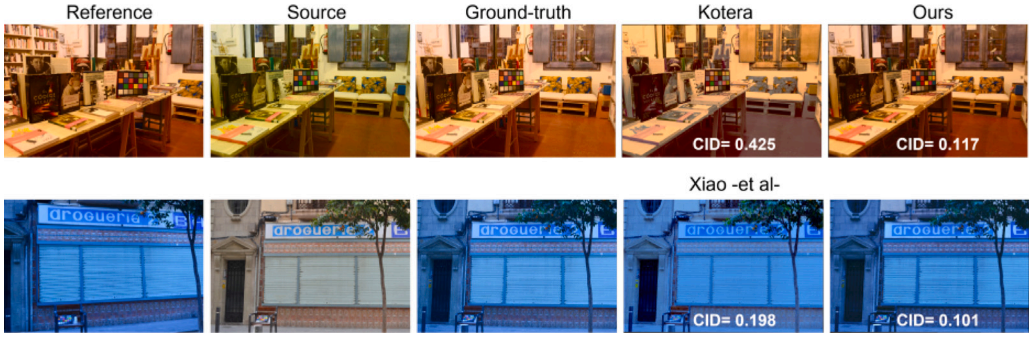

Fig. 5. Results for the case in which the source image is gamma encoded and the reference image is logarithmically encoded. Please, focus on the color of the books and the sofas in the top image, and on the gray and yellow of the walls in the bottom one. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Table 3
Results for the comparison of the 35 pairs in the case where the source image is gamma encoded and the reference image is logarithmically encoded.

	ΔE_{00}		PSNR L		CPSNR		CID		RMSE	
	Mean	Median	Mean	Median	Mean	Median	Mean	Median	Mean	Median
Kotera	12.658	9.202	18.629	20.748	17.893	20.089	0.538	0.430	0.162	0.099
Pitie	3.752	3.903	25.957	25.538	25.378	25.217	0.184	0.173	0.057	0.055
Reinhard	6.438	6.246	22.861	22.642	21.776	21.666	0.291	0.291	0.084	0.083
Xiao	6.794	5.734	23.023	22.77	22.097	22.215	0.322	0.314	0.081	0.077
Ferradans	6.317	6.165	22.222	21.826	21.577	21.357	0.318	0.298	0.089	0.086
Park	12.808	9.620	20.746	22.593	18.779	19.351	0.510	0.454	0.147	0.108
Grogan	9.450	8.775	19.243	19.329	18.967	18.464	0.418	0.453	0.127	0.122
Chiu	6.345	6.039	21.083	20.888	20.800	20.769	0.326	0.317	0.096	0.093
Ho	10.331	11.233	20.930	20.889	20.131	19.418	0.509	0.543	0.140	0.139
Yoo	8.105	7.948	19.431	19.685	19.204	19.442	0.436	0.442	0.120	0.107
Proposed 3 × 3 1D	7.031	4.804	22.626	24.081	22.158	23.737	0.282	0.239	0.095	0.065
Proposed 4 × 4 1D	4.419	4.213	25.112	25.342	24.662	25.109	0.195	0.183	0.061	0.056
Proposed 3 × 3 3 × 1D	4.007	3.752	25.867	26.091	25.384	25.386	0.180	0.171	0.057	0.055
Proposed 4 × 4 3 × 1D	3.911	3.710	25.898	26.094	25.458	25.489	0.174	0.160	0.056	0.054

Table 4
Results for the comparison of the 35 pairs in the case where the source image is logarithmically encoded and the reference image is gamma encoded.

	ΔE_{00}		PSNR L		CPSNR		CID		RMSE	
	Mean	Median	Mean	Median	Mean	Median	Mean	Median	Mean	Median
Kotera	15.704	12.405	17.017	18.864	15.970	16.625	0.631	0.586	0.199	0.148
Pitie	3.909	3.830	25.796	25.498	25.225	25.020	0.200	0.201	0.059	0.056
Reinhard	7.928	7.516	21.260	21.056	18.883	18.687	0.393	0.392	0.117	0.116
Xiao	7.926	7.554	21.446	20.539	20.438	20.059	0.403	0.416	0.100	0.099
Ferradans	8.578	7.954	19.654	19.163	19.172	18.518	0.381	0.369	0.122	0.119
Park	5.895	5.242	24.038	23.352	22.972	22.294	0.305	0.290	0.078	0.077
Grogan	9.864	9.217	18.714	18.901	18.486	18.663	0.473	0.466	0.135	0.123
Chiu	5.929	6.064	21.481	21.246	21.240	21.285	0.313	0.322	0.094	0.093
Ho	10.647	10.138	20.155	19.855	18.674	19.263	0.481	0.479	0.135	0.128
Yoo	5.471	5.207	22.794	22.726	22.341	22.307	0.273	0.256	0.080	0.077
proposed 3 × 3 1D	7.351	5.882	21.841	22.723	21.394	22.296	0.330	0.296	0.104	0.079
proposed 4 × 4 1D	4.916	4.763	24.491	24.462	24.014	24.082	0.237	0.221	0.069	0.065
Proposed 3 × 3 3 × 1D	4.244	3.866	25.099	24.862	24.709	24.662	0.198	0.166	0.062	0.061
Proposed 4 × 4 3 × 1D	4.248	4.106	24.967	24.886	24.634	24.886	0.196	0.168	0.062	0.062

everywhere a little wrong whereas our output matches the ground truth. In the second example (bottom row) focus on the window of the blinds in the shop. We correctly render these as slightly yellowish where as they are bluer in the method of Xiao et al. Again the CID scores tell the same story: our method is performing better than the state of the art.

4.4.4. Log to gamma

Results for this case are presented in Table 3. In this case, the method of Pitie is able to overcome us in some metrics. This said we need to remember that the method of Pitie showed a quite bad performance in the cases of two gamma-encoded and two logarithmic-encoded images, therefore proving that our method is much more robust to different non-linearities than that of Pitie et al.. We believe it is also interesting to point out that in this particular setting, our method with 3 different curves (one per color channel) and a 3 × 3 matrix seems to slightly improve over the 4 × 4 matrix case.

Qualitative results are shown in Fig. 6. In the top row, we compare our output to those of Park et al. for one image. Clearly, our method delivers good fidelity with the original but the Park et al. output is wrong (especially the ground in the image). The model of Ferradans et al. is considered in the second example. Here our method again delivers good results with the Ferradans method delivering an image that has an erroneous color cast. As for the other 3 qualitative examples, the CID numbers are consistent with the qualitative comparisons.

4.5. Comparisons against methods that require encoding information

In this subsection, we want to compare the results obtained by our agnostic method against those obtained by the methods of Gil Rodríguez et al. [37] (Gil18) and [4] (Gil20). We want to note that the results of these two methods can be understood as upper-bounds for our method, as they consider the information about the type of curve used in the encoding. In Table 5 we present the results as the average for all the previous four cases. In our method, we consider the case of a 4 × 3 matrix H and three different tone-mapping curves.

Table 5
Comparison between the results of our method and the results of methods inputting extra information for the 140 comparisons (35 comparisons per non-linearity pair). Our method is able to compete well, especially for ΔE_{00}, CID, and RMSE without the need for any extra information.

	ΔE_{00}	PSNR L	CPSNR	CID	RMSE
Ours	3.767	25.921	25.540	0.169	0.056
Gil18	3.928	27.186	25.852	0.179	0.056
Gil20	3.371	27.515	26.442	0.154	0.050

As was to be expected, our method is not able to outperform these methods which use extra information. This said, we want to remark on how our method gets close to them for ΔE_{00} (only a difference of 0.39 versus Gil20, and better than Gil18), CID (only a difference of 0.15 versus Gil20, and better than Gil18), and RMSE (only a difference of 0.006 versus Gil20, and equal to Gil18)

4.6. HLG/PQ/LogC arri dataset

Results for this dataset are shown in Table 6, where we present the mean over the 10 image pairs. We show our results for the case of a 4 × 3 matrix H and considering three different curves. We can see how our method outperforms all the other methods that do not require extra

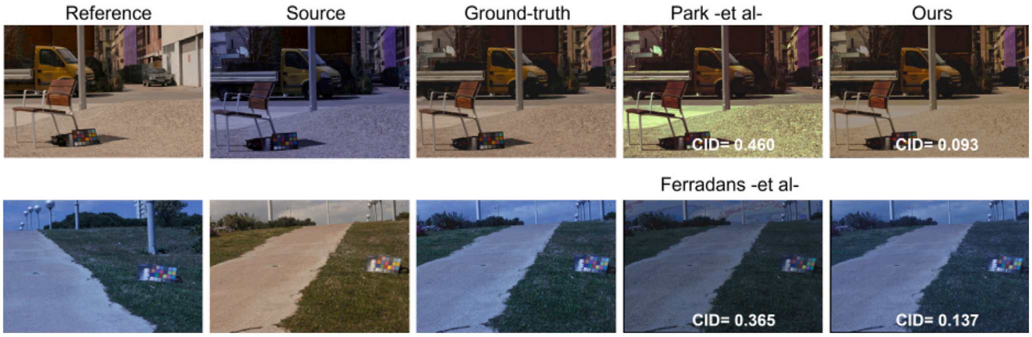

Fig. 6. Results for the case in which the source image is logarithmically encoded and the reference image is gamma encoded. Please note the yellow of the truck and the floor color in the top image. Also, please realize the bluish color that is present in both our result and the ground-truth, in comparison to the method by Ferradans et al., which presents a grayish cast. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

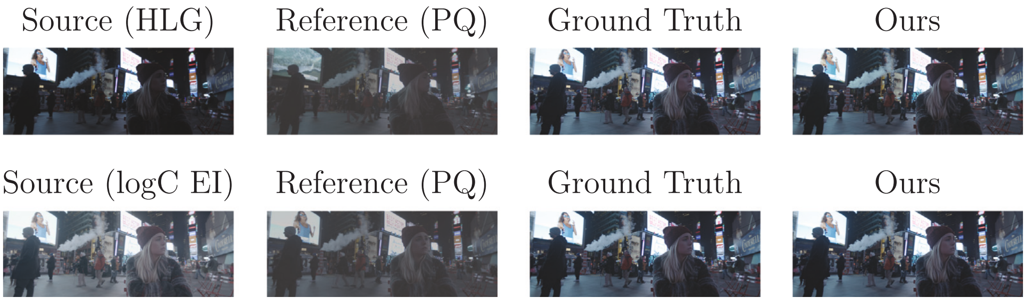

Fig. 7. Examples with PQ, HLG and Log C curves. From left to right, reference, source, GT and our result. Each row represents a different comparison. The GTs and our results are tone mapped using [42]. Images from ARRI [43]. In the case of PQ curve, we set up the absolute luminance of the display to 1000 cd/m^2.

Table 6
Results show mean averages over 10 pairs, where reference and source images are encoded using HLG, PQ and logarithmic curves. In the case of PQ curve, we set up the absolute luminance of the display to 1000 cd/m^2.

		ΔE_{00}	PSNR L	CPSNR	CID	RMSE
No info	Kotera	3.344	32.505	30.567	0.110	0.045
	Pitie	1.022	40.047	40.134	0.035	0.021
	Reinhard	1.861	35.311	35.020	0.062	0.040
	Xiao	1.891	32.965	32.789	0.061	0.032
	Ferradans	4.820	24.692	24.624	0.183	0.073
	Park	1.624	38.250	36.795	0.044	0.029
	Grogan	3.615	28.981	28.558	0.143	0.052
	Chiu	2.559	27.082	26.765	0.212	0.050
	Ho	5.444	25.135	24.861	0.254	0.075
	Yoo	3.334	26.006	25.713	0.266	0.056
	Ours	0.780	51.982	51.550	0.016	0.011
Info	Gil18	1.775	36.086	35.636	0.068	0.029
	Gil20	0.310	48.324	47.649	0.002	0.005

Table 7
Average running time (s) on 35 images of size 768 × 1152.

	Kotera	Pitie	Reinhard	Xiao	Ferradans	Park	Grogan	Ours
Time (s)	8.867	9.107	2.153	2.059	53.790	316	9.440	5.198

information and also the one of Gil 18 [37] –which makes use of the additional knowledge about the form of the non-linearity–. Also, our method is able to outperform the method of Gil Rodríguez et al. [4] –another one using extra information– for both PSNR L and CPSNR.

In Fig. 7 we show two examples of the outputs generated by our method. Our results are very close to the ground truth.

4.7. Time comparison

Table 7 presents the execution times for methods that do not require any learning. Processing times are computed for images of size 768 × 1152. We see that our method is faster than competing methods that deliver similar good performance to our method—Park and Pitie. Simpler methods that perform worse on the Tables –Reinhard and Xiao– are slightly faster. When possible, we used the code provided by the original authors. The executions were performed on a desktop computer with Ubuntu 22.04.4 LTS as the operating system and the following specifications: Intel(R) Core(TM) i5-8400 CPU at 2.80 GHz processor, 16 GB RAM, and Nvidia GeForce GTX 1060 graphics card. We use Matlab version R2019a.

4.8. Realistic scenarios

Fig. 8 demonstrates the ability of our method to work with different cameras without knowing their characteristics. In detail, we show how we can convert between two Smartphone cameras: an iPhone 12 (Phone 1) and a Realme 8 Pro (Phone 2). Fig. 9 goes one step beyond and shows how our method is also able to work with images downloaded from the internet, where we do not know any information about the capture conditions.

4.9. Limitations

Our method relies on having good enough pixel-to-pixel correspondences between the source and the reference images. For this reason, it may present limitations when the viewpoints from the two images are too different to detect enough correspondences in the shared regions, when one of the images is a large zoom-in of the other, or when one of the images is captured under extreme lighting conditions such as low-light. These limitations might be reduced if replacing SIFT by some other methods such as [44–46].

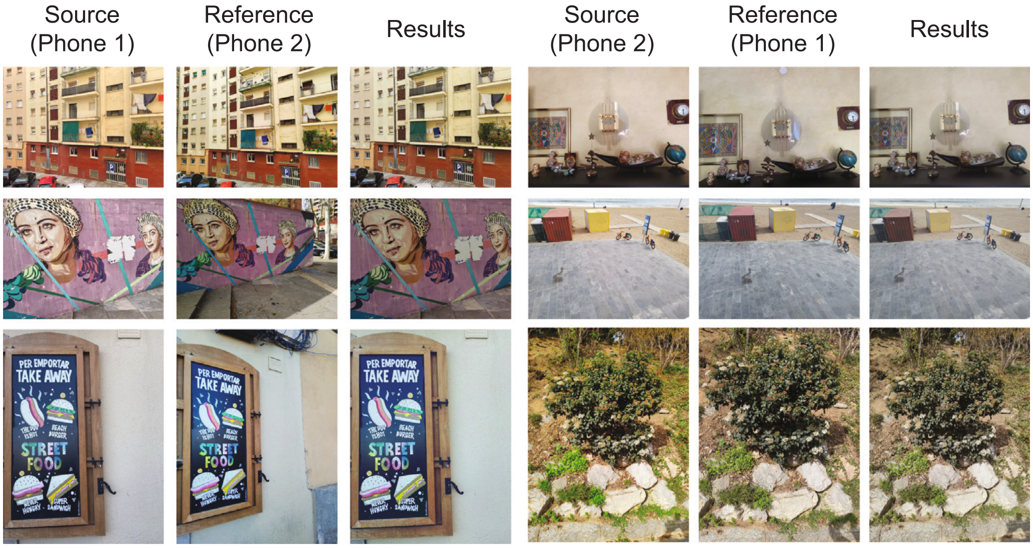

Fig. 8. Examples with images captured under realistic scenarios. Our method is able to color match the images between two Smartphone cameras: an iPhone 12 (Phone 1) and a Realme 8 Pro (Phone 2).

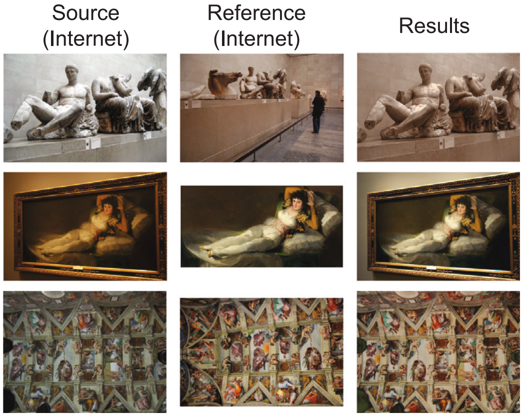

Fig. 9. Examples with images captured under realistic scenarios. Our method is able to color correct images downloaded from internet, without any further information about the capture conditions.

5. Conclusions

In this paper, we have introduced a color matching method able to deal with any encoded images (gamma-corrected, logarithmic-encoded, or any other) that share some content but are out-of-registration. To this end, we aimed to understand the color transform from source to reference –in a known pixel specific way– so we can apply the learned transform to the source so it looks like the reference when transformed.

Our method has 4 steps. First, we find common points in the source and reference images using the SIFT methodology. Then we relate the corresponding source and reference points using a shading homography (a global color transform and a per pixel shading term). Third, we approximate the shading change using an optimized tone-curve. Fourth we apply the tone-curve and the color transform to the source. A key strength of our method is that we make no a priori assumption about the shape of the tone function or that it belongs to a particular mathematical family (e.g. gamma curves)

Extensive experiments demonstrate that our method advances the state of the art. Moreover, we also compete with or even improve over methods that employ additional information about the type of non-linearities used in encoding the images.

Further work will aim at generalizing this approach to other color-related tasks such as color characterization, exposure correction, image harmonization, or HDR imaging.

CRediT authorship contribution statement

Raquel Gil Rodríguez: Methodology, Conceptualization, Writing – review & editing. **Javier Vazquez-Corral:** Methodology, Conceptualization, Writing – review & editing, Writing – original draft. **Marcelo Bertalmío:** Writing – review & editing. **Graham D. Finlayson:** Writing – review & editing.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

Acknowledgments

This work has received funding from different sources. RGR was supported by ERC Advanced Grant, Color3.0. JVC was supported by Grant PID2021-128178OB-I00 funded by MCIN/AEI/10.13039/501100011033 and by ERDF "A way of making Europe", and also by the Departament de Recerca i Universitats from Generalitat de Catalunya with reference 2021SGR01499. MB was supported by grant PID2021-127373NB-I00 from the Spanish Ministry of Science, Innovation and Universities (MICINN). GF was funded by EPSRC, United Kingdom Grant EP/S028730/1.

Appendix. Minimization of the alternating least squares

Our goal is to minimize:

$$\min_{S,H} \|\mathcal{I}_r - S \cdot \mathcal{I}_s \cdot H\| \tag{A.1}$$

where $\mathcal{I}_r$, $\mathcal{I}_s$ represent the RGB values of the corresponding pixels in the reference image and the source image respectively. We will minimize this equation iteratively for H and S. To do so, three steps in each iteration need to be performed —let us consider as an example the first iteration, $\mathcal{I}^0 = \mathcal{I}_s$-:

```
Inputs: The corresponding pixels of the reference image $\mathcal{I}_r$ and
  the corresponding pixels of the source image $\mathcal{I}_s$.
Output: $H^i$, $S^{final}$.
Initialize $\mathcal{I}^0 = \mathcal{I}_s$, $i = 0$, and $S^{final} = Id$ ;
repeat
    $\min_{H^i} ||\mathcal{I}_r - H^i\mathcal{I}^i||$ ;
    $\min_{S^i} ||\mathcal{I}_r - H^i\mathcal{I}^i S^i||$;
    Update $S^{final} \leftarrow S^{final} \cdot S^i$;
    Update $\mathcal{I}^{i+1} = \mathcal{I}^i S^i$;
    Update $i \leftarrow i + 1$.
until $||\mathcal{I}^{i+1} - \mathcal{I}^i|| < \epsilon$, where $\epsilon$ is a predefined tolerance value;
```

Algorithm 2: ALS algorithm for color stabilization

1. Minimizing for H (assuming S is the identity):

$$\min_{H^1} \|\mathcal{I}_r - H^1\mathcal{I}^0\|. \tag{A.2}$$

2. Minimizing for S:

$$\min_{S^1} \|\mathcal{I}_r - H^1\mathcal{I}^0 S^1\|. \tag{A.3}$$

3. Updating the image for the next iteration:

$$\mathcal{I}^1 = H^1\mathcal{I}^0 S^1. \tag{A.4}$$

We keep iterating over these operations until $\|\mathcal{I}^{i+1} - \mathcal{I}^i\| < \epsilon$.

Let us look in more detail into the minimization of the first two steps. The first step is a standard least squares minimization and can therefore be solved by

$$H^1 = \mathcal{I}_r(\mathcal{I}^0)^+, \tag{A.5}$$

where $^+$ denotes the pseudo-inverse.

The second step is more intricate. We should remember that S is a diagonal $n \times n$ image. Therefore, we need to force the solution of the minimization to be diagonal. We can do that in closed form as:

$$S^1_{(jj)} = (V_{(j)} \cdot \mathcal{I}_{r(j)})/\|V_{(j)}\|^2, \tag{A.6}$$

where $V = H^1 I^0$, $V_{(j)}$ and $\mathcal{I}_{r_{(j)}}$ are 1-by-3 vectors (the values of the column j in V and $\mathcal{I}_r$), and $\cdot$ denotes the scalar product. An algorithm presenting this minimization is shown in Algorithm 2.

References

[1] S. Bianco, A. Bruna, F. Naccari, R. Schettini, Color space transformations for digital photography exploiting information about the illuminant estimation process, J. Opt. Soc. Am. (JOSA) 29 (3) (2012) 374–384.

[2] Javier Vazquez-Corral, Marcelo Bertalmío, Color stabilization along time and across shots of the same scene, for one or several cameras of unknown specifications, IEEE Trans. Image Process. (TIP) 23 (10) (2014) 4564–4575.

[3] Javier Vazquez-Corral, 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, Marcelo Bertalmío, Color matching images with unknown non-linear encodings, IEEE Trans. Image Process. 29 (2020) 4435–4444.

[5] Graham Finlayson, Han Gong, Robert B. Fisher, Color homography: theory and applications, IEEE Trans. Pattern Anal. Mach. Intell. (TPAMI) 41 (1) (2019) 20–33.

[6] Yoav HaCohen, Eli Shechtman, Dan B. Goldman, Dani Lischinski, Non-rigid dense correspondence with applications for image enhancement, ACM Trans. Graph. 30 (4) (2011) 70.1–70.10.

[7] MediaCollege, CCU (camera control unit) operations, 2012.

[8] P. Postma, B. Chorley, Colour grading with colour management, in: SMPTE15: Persistence of Vision - Defining the Future, 2015, pp. 1–8.

[9] Erik Reinhard, Michael Ashikhmin, Bruce Gooch, Peter Shirley, Color transfer between images, IEEE Comput. Graph. Appl. 21 (5) (2001) 34–41.

[10] F. Pitié, R. Dahyot, A.C. Kokaram, N-dimensional probablility density function transfer and its application to colour transfer, in: IEEE International Conference on Computer Vision, ICCV, Vol. 2, 2005, pp. 1434–1439.

[11] François Pitié, Anil C. Kokaram, Rozenn Dahyot, Automated colour grading using colour distribution transfer, Comput. Vis. Image Underst. 107 (1) (2007) 123–137.

[12] J. Rabin, S. Ferradans, N. Papadakis, Adaptive color transfer with relaxed optimal transport, in: IEEE International Conference on Image Processing, ICIP, 2014, pp. 4852–4856.

[13] S. Ferradans, N. Papadakis, G. Peyré, J. Aujol, Regularized discrete optimal transport, SIAM J. Imaging Sci. 7 (3) (2014) 1853–1882.

[14] Mairéad Grogan, Rozenn Dahyot, L2 divergence for robust colour transfer, Comput. Vis. Image Underst. 181 (2019) 39–49.

[15] H. Kotera, A scene-referred color transfer for pleasant imaging on display, in: IEEE International Conference on Image Processing, ICIP, Vol. 2, 2005, pp. 5–8.

[16] Tania Pouli, Erik Reinhard, Progressive color transfer for images of arbitrary dynamic range, Comput. Graph. (2011).

[17] Xuezhong Xiao, Lizhuang Ma, Color transfer in correlated color space, in: PACM International Conference on Virtual Reality Continuum and Its Applications, in: VRCIA, ACM, New York, NY, USA, 2006, pp. 305–309.

[18] Xuezhong Xiao, Lizhuang Ma, Gradient-preserving color transfer, Comput. Graph. Forum 28 (7) (2010) 1879–1886.

[19] R.M.H. Nguyen, S.J. Kim, M.S. Brown, Illuminant aware gamut-based color transfer, Comput. Graph. Forum 33 (7) (2014) 319–328.

[20] Y. Hwang, J.Y. Lee, I.S. Kweon, S.J. Kim, Color transfer using probabilistic moving least squares, in: IEEE Conference on Computer Vision and Pattern Recognition, CVPR, 2014, pp. 3342–3349.

[21] Han Gong, Graham Finlayson, Robert Fisher, Recoding color transfer as a color homography, in: Proceedings of the British Machine Vision Conference, BMVC, BMVA Press, 2016, pp. 17.1–17.11.

[22] Han Gong, Graham D. Finlayson, Robert B. Fisher, Fufu Fang, 3D color homography model for photo-realistic color transfer re-coding, Vis. Comput. (2017) 1–11.

[23] Yijun Li, Ming-Yu Liu, Xueting Li, Ming-Hsuan Yang, Jan Kautz, A closed-form solution to photorealistic image stylization, in: Proceedings of the European Conference on Computer Vision, ECCV, 2018, pp. 453–468.

[24] Tai-Yin Chiu, Danna Gurari, PhotoWCT2: Compact autoencoder for photorealistic style transfer resulting from blockwise training and skip connections of high-frequency residuals, in: IEEE/CVF Winter Conference on Applications of Computer Vision, WACV, 2022, pp. 2978–2987.

[25] Man M. Ho, Jinjia Zhou, Deep preset: Blending and retouching photos with color style transfer, in: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, WACV, 2021, pp. 2113–2121.

[26] Jaejun Yoo, Youngjung Uh, Sanghyuk Chun, Byeongkyu Kang, Jung-Woo Ha, Photorealistic style transfer via wavelet transforms, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2019, pp. 9036–9045.

[27] Songhua Liu, Kai Wang, Xingyi Yang, Jingwen Ye, Xinchao Wang, Dataset distillation via factorization, Adv. Neural Inf. Process. Syst. 35 (2022) 1100–1113.

[28] Xingyi Yang, Jingwen Ye, Xinchao Wang, Factorizing knowledge in neural networks, in: European Conference on Computer Vision, Springer, 2022, pp. 73–91.

[29] Yu-Wing Tai, Jiaya Jia, Chi-Keung Tang, Local color transfer via probabilistic segmentation by expectation-maximization, in: IEEE Computer Society Conference on Computer Vision and Pattern Recognition, CVPR, 2005, pp. 747–754.

[30] Yao Xiang, Beiji Zou, Hong Li, Selective color transfer with multi-source images, Pattern Recognit. Lett. 30 (7) (2009) 682–689.

[31] O. Frigo, N. Sabater, J. Delon, P. Hellier, Motion driven tonal stabilization, IEEE Trans. Image Process. (TIP) 25 (11) (2016) 5455–5468.

[32] Yoav HaCohen, Eli Shechtman, Dan B. Goldman, Dani Lischinski, Optimizing color consistency in photo collections, ACM Trans. Graph. 32 (4) (2013) 85.1–85.9.

[33] J. Park, Y.W. Tai, S.N. Sinha, I.S. Kweon, Efficient and robust color consistency for community photo collections, in: IEEE Conference on Computer Vision and Pattern Recognition, CVPR, 2016, pp. 430–438.

[34] W. Xu, J. Mulligan, Performance evaluation of color correction approaches for automatic multi-view image and video stitching, in: IEEE Computer Society Conference on Computer Vision and Pattern Recognition, CVPR, 2010, pp. 263–270.

[35] M. Xia, J.Y. Renping, X.M. Zhang, J. Xiao, Color consistency correction based on remapping optimization for image stitching, in: IEEE International Conference on Computer Vision Workshops, ICCVW, 2017, pp. 2977–2984.

[36] Graham D Finlayson, Maryam Mohammadzadeh Darrodi, Michal Mackiewicz, The alternating least squares technique for nonuniform intensity color correction, Color Res. Appl. 40 (3) (2015) 232–242.

[37] Raquel Gil Rodríguez, Javier Vazquez-Corral, Marcelo Bertalmío, Color-matching shots from different cameras having unknown Gamma or logarithmic encoding curves, in: SMPTE Annual Technical Conference & Exhibition, 2017, pp. 1–15.

[38] ARRI, ARRIRAW converter, 2018, http://www.arri.com/camera/alexa/tools/arriraw_converter/. Accessed 2018.

[39] G. Sharma, W. Wu, E.N. Dalal, The CIEDE2000 color-difference formula: implementation notes, supplementary test data, and mathematical observations, Color Res. Appl. 30 (1) (2005) 21–30.

[40] Ingmar Lissner, Jens Preiss, Philipp Urban, Matthias Scheller Lichtenauer, Peter Zolliker, Image-difference prediction: From grayscale to color, IEEE Trans. Image Process. (TIP) 22 (2) (2013) 435–446.

[41] Zhou Wang, A.C. Bovik, H.R. Sheikh, E.P. Simoncelli, Image quality assessment: from error visibility to structural similarity, IEEE Trans. Image Process. (TIP) 13 (4) (2004) 600–612.

[42] Praveen Cyriac, Trevor Canham, David Kane, Marcelo Bertalmío, Vision models fine-tuned by cinema professionals for high dynamic range imaging in movies, Multimedia Tools Appl. 80 (2) (2021) 2537–2563.

[43] Harald Brendel, ALEXA Log C Curve- Usage in VFX, Technical Report, ARRI, 2011.

[44] Gucan Long, Laurent Kneip, Jose M Alvarez, Hongdong Li, Xiaohu Zhang, Qifeng Yu, Learning image matching by simply watching video, in: Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, the Netherlands, October 11-14, 2016, Proceedings, Part VI 14, Springer, 2016, pp. 434–450.

[45] Jiayi Ma, Xingyu Jiang, Aoxiang Fan, Junjun Jiang, Junchi Yan, Image matching from handcrafted to deep features: A survey, Int. J. Comput. Vis. 129 (1) (2021) 23–79.

[46] Matheus Machado Dos Santos, Giovanni G. De Giacomo, Paulo L.J. Drews, Silvia S.C. Botelho, Matching color aerial images and underwater sonar images using deep learning for underwater localization, IEEE Robot. Autom. Lett. 5 (4) (2020) 6365–6370.

Raquel Gil Rodriguez obtained her PhD from Universitat Pompeu Fabra (UPF), Barcelona in 2018. She is a research scientist in computational color at the Psychology Department, Justus-Liebig Universität, Giessen. Her main interest is in understanding the way we perceive the world, especially color.

Javier Vazquez-Corral received the Ph.D. degree in computer science from the Universitat Autònoma de Barcelona (UAB), Bellaterra, Spain, in 2011. He worked as Postdoctoral researcher at different universities, such as the Universitat Pompeu Fabra (UPF) in Barcelona and the University of East Anglia in Norwich. He is currently an Associate Professor at the Universitat Autònoma de Barcelona (UAB). His research interests are related to the use of color in image processing and computer vision problems. He is also interested in bridging the gap between color in the human brain and its use in computer-vision applications.

Marcelo Bertalmí,o is a Scientific Researcher with the Spanish National Research Council (CSIC) in Madrid Spain. He received the B.Sc. and M.Sc. degrees in electrical engineering from the Universidad de la República, Uruguay, and the Ph.D. degree in electrical and computer engineering from the University of Minnesota, USA, in 2001. His current research interests are in developing image processing algorithms for cinema that mimic neural and perceptual processes in the visual system, and to investigate new vision models based on the efficient representation principle.

Graham Finlayson (Member, IEEE) is currently a Professor with the School of Computing Sciences, University of East Anglia, U.K. His research interests include color, physics-based computer vision, image processing, and the engineering required to embed technology in devices. He is a fellow of the Royal Photographic Society, the Society for Imaging Science and Technology, and the Institution for Engineering Technology.