ORIGINAL ARTICLE

CrossMark

3D color homography model for photo-realistic color transfer re-coding

Han Gong[1] · Graham D. Finlayson[1] · Robert B. Fisher[2] · Fufu Fang[1]

Abstract

Color transfer is an image editing process that naturally transfers the color theme of a source image to a target image. In this paper, we propose a 3D color homography model which approximates photo-realistic color transfer algorithm as a combination of a 3D perspective transform and a mean intensity mapping. A key advantage of our approach is that the re-coded color transfer algorithm is simple and accurate. Our evaluation demonstrates that our 3D color homography model delivers leading color transfer re-coding performance. In addition, we also show that our 3D color homography model can be applied to color transfer artifact fixing, complex color transfer acceleration, and color-robust image stitching.

Keywords Color transfer · Color grading · Color homography · Tone mapping

1 Introduction

Color palette modification for pictures/frames is often required in professional photograph editing as well as the video postproduction. Artists usually choose a desired target picture as a reference and manipulate the other pictures to make their color palette similar to the reference. This process is known as photo-realistic color transfer. Figure 1 shows an example of photo-realistic color transfer between a target image and a source image. This color transfer process is a complex task that requires artists to carefully adjust for multiple properties such as exposure, brightness, white point, and color mapping. These adjustments are also interdependent, *i.e.*, the alignment for a single property can cause the other previously aligned properties to become misaligned. Some artifacts due to nonlinear image processing (*e.g.*, JPEG block edges) may also appear after color adjustment.

One of the first photo-realistic color transfer methods was introduced by Reinhard et al. [23]. Their method proposed that the mean and variance of the source image, in a specially chosen color space, should be manipulated to

match those of a target. More recent methods [1,16,19–21] might adopt more aggressive color transfers—*e.g.*, color distribution force matches [19,20]—and yet, these aggressive changes often do not preserve the original intensity gradients and new spatial type artifacts may be introduced into an image (*e.g.*, JPEG blocks become visible or there is false contouring). In addition, the complexity of a color transfer method usually leads to longer processing time. To address these issues, previous methods [11,13,18] were proposed to approximate the color change produced by a color transfer, such that an original complicated color transfer can be re-formulated as a simpler and faster algorithm with an acceptable level of accuracy and some introduced artifacts.

In this paper, we propose a simple and general model for re-coding (approximating) an unknown photo-realistic color transfer which provides leading accuracy and the color transfer algorithm can be decomposed into meaningful parts. Our model is extended from a recent planar color homography color transfer model [11] to the 3D domain, as opposed to the original 2D planar domain. In our improved model, we decompose an unknown color transfer into 3D color perspective transform and mean intensity mapping components. Based on [11], we make two new contributions: (1) a 3D color mapping model that better re-codes color change by relating two homogeneous color spaces and (2) a monotonic mean intensity mapping method that prevents artifacts without adding unwanted blur. Our experiments show significant improvements in color transfer re-coding accuracy. We

✉ Han Gong
  h.gong@uea.ac.uk, gong@fedoraproject.org

1 School of Computing Sciences, University of East Anglia, Norwich, UK

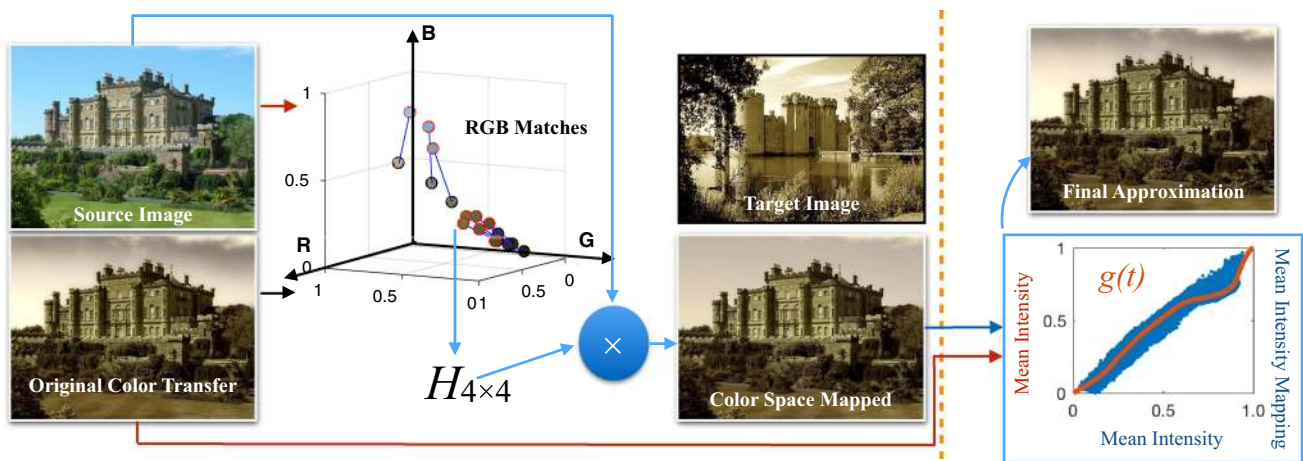2 School of Informatics, University of Edinburgh, Edinburgh, UK

Fig. 1 Pipeline of our color transfer decomposition. The "target" image was used by the "original color transfer" algorithms to produce the "original color transfer" output image, but it is not used by the proposed color transfer re-coding algorithm. The orange dashed line divides the pipeline into two steps: (1) color space mapping: The RGBs of the source image (drawn as red edge circles) and the original color transfer image (by [16] with the target image as the reference, black edge cir-cles) are matched according to their locations (*e.g.*, the blue matching lines), from which we estimate a 3D homography matrix H and use H to transfer the source image colors and (2) mean intensity mapping: The image mean intensity values (mean values of R, G, and B) are aligned by estimating the per-pixel shading between the color space-mapped result and the original color transfer result by least squares. The final result is a visually close color transfer approximation

demonstrate three applications of the proposed method for color transfer artifact fixing, color transfer acceleration, and color-robust image stitching.

Throughout the paper, we denote the source image by I_s and the original color transfer result by I_t. Given I_s and I_t, we re-code the color transfer with our color homography model which approximates the original color transfer from I_s to I_t. Figure 1 shows the pipeline of our color transfer decomposition.

Our paper is organized as follows. We review the leading prior color transfer methods and the previous color transfer approximation methods in Sect. 2. Our color transfer decomposition model is described in Sect. 3. We present a color transfer re-coding method for two corresponding images in Sect. 4. In addition, we demonstrate its applications in Sect. 5. Finally, we conclude in Sect. 6.

2 Background

In this section, we briefly review the existing work on photo-realistic color transfer, the methods for re-coding such a color transfer, and the concept of Color Homography.

2.1 Photo-realistic color transfer

Example-based color transfer was first introduced by Reinhard et al. [23]. Their method aligns the color distributions of two images in a specially chosen color space via 3D scaling and shift. Pitie et al. [19,20] proposed an iterative color transfer method that distorts the color distribution by random 3D rotation and per-channel histogram matching until the distributions of the two images are fully aligned. This method makes the output color distribution exactly the same as the target image's color distribution. However, the method introduces spatial artifacts. By adding a gradient preservation constraint, these artifacts can be mitigated or removed at the cost of more blurry artifacts [20]. Pouli and Reinhard [21] adopted a progressive histogram matching in L*a*b* color space. Their method generates image histograms at different scales. From coarse to fine, the histogram is progressively reshaped to align the maxima and minima of the histogram, at each scale. Their algorithm also handles the difference in dynamic ranges between two images. Nguyen et al. [16] proposed an illuminant-aware and gamut-based color transfer. They first eliminate the color cast difference by a white-balancing operation for both images. A luminance alignment is later performed by histogram matching along the "gray" axis of RGB. They finally adopt a 3D convex hull mapping to limit the color-transferred RGBs to the gamut of the target RGBs. Other approaches (*e.g.*, [1,25,28]) solve for several local color transfers rather than a single global color transfer. As most non-global color transfer methods are essentially a blend of several single color transfer steps, a global color transfer method is extendable for multi-transfer algorithms.

2.2 Photo-realistic color transfer re-coding

Various methods have been proposed for approximating an unknown photo-realistic color transfer for better speed

and naturalness. Pitie et al. [18] proposed a color transfer approximation by a 3D similarity transform (translation + rotation + scaling) which implements a simplification of the earth mover's distance. By restricting the form of a color transfer to a similarity transform model, some of the generality of the transfer can be lost such that the range of color changes it can account for is more limited. In addition, a color transfer looks satisfying only if the tonality looks natural and this is often not the case with the similarity transformation. Ilie and Welch [13] proposed a polynomial color transfer which introduces higher-degree terms of the RGB values. This encodes the non-linearity of color transfer better than a simple 3×3 linear transform. However, the nonlinear polynomial terms may over-manipulate a color change and introduce spatial gradient artifacts. Similarly, this method also does not address the tonal difference between the input and output images. Gong et al. [11] proposed a planar color homography model which re-codes a color transfer effect as a combination of 2D perspective transform of chromaticity and shading adjustments. Compared with [18], it requires less parameters to represent a non-tonal color change. The model's tonal adjustment also further improves color transfer re-coding accuracy. However, the assumption of a 2D chromaticity distortion limits the range of color transfer it can represent. Their [11] tonal adjustment (mean intensity-to-shading mapping) also does not preserve image gradients and the original color rank. Another important work is probabilistic moving least squares [12] which calculates a largely parameterized transform of color space. Its accuracy is slightly better than [13]. However, due to its high complexity, it is unsuitable for real-time use. In this paper, we only benchmark against the color transfer re-coding methods with a real-time performance.

2.3 2D color homography

The color homography theorem [7,8] shows that chromaticities across a change in capture conditions (light color, shading, and imaging device) are a homography apart. Suppose that we have an RGB vector $\rho = [R, G, B]^\mathsf{T}$, its r and g chromaticity coordinates are written as $r = R/B$, $g = G/B$ which can be interpreted as a homogeneous coordinate vector $\underline{c}$ and we have:

$$\underline{c} \propto \begin{bmatrix} r & g & 1 \end{bmatrix}^\mathsf{T}. \tag{1}$$

when the shading is uniform and the scene is diffuse, it is well known that across a change in illumination or a change in device, the corresponding RGBs are, to a reasonable approximation, related by a 3×3 linear transform $H_{3\times3}$:

$$\underline{\rho}'^\mathsf{T} = \underline{\rho}^\mathsf{T} H_{3\times3} \tag{2}$$

where $\underline{\rho}'$ is the corresponding RGBs under a second light or captured by a different camera [14,15]. Due to different shading, the RGB triple under a second light is written as

$$\underline{c}'^\mathsf{T} = \alpha \underline{c}^\mathsf{T} H_{3\times3} \tag{3}$$

where $H_{3\times3}$ here is a color homography color correction matrix, α denotes an unknown scaling. Without loss of generality, let us interpret $\underline{c}$ as a homogeneous coordinate, *i.e.*, assume its third component is 1. Then, rg chromaticities $\underline{c}^\mathsf{T}$ and $\underline{c}'^\mathsf{T}$ are a homography apart.

The 2D color homography model decomposes a color change into a 2D chromaticity distortion and a 1D tonal mapping, which successfully approximates a range of physical color changes. However, the degrees of freedom of a 2D chromaticity distortion may not accurately capture more complicated color changes applied in photograph editing.

3 3D color homography model for photo-realistic color transfer re-coding

The color homography theorem reviewed in Sect. 2.3 states that the same diffuse scene under an illuminant (or camera) change will result in two images whose chromaticities are a 2D (planar) homography apart. The 2D model works best for RAW-to-RAW linear color mappings and can also approximates the nonlinear mapping from RAW to sRGB [8]. In this paper, we extend the planar color homography color transfer to the 3D spatial domain. A 3D perspective transformation in color may be interpreted as a perspective distortion of color space (*e.g.*, Fig. 2). Compared with the 2D homography model, the introduction of a 3D perspective transform can model a higher degree of non-linearity in the color mapping. We propose that a global color transfer can be decomposed into a 3D perspective transform and a mean intensity alignment.

We start with the outputs of the previous color transfer algorithms. We represent a 3D RGB intensity vector by its
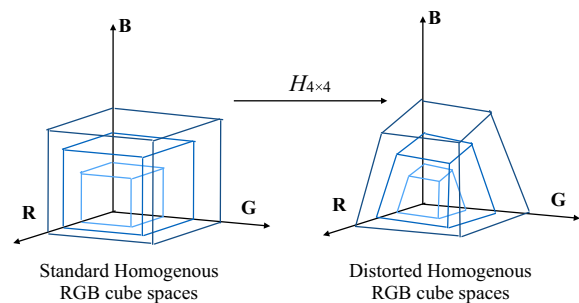

Fig. 2 Homogeneous color space mapping. The left 3 homogeneous RGB cubes are equivalent (up to a scale). The left RGB space cubes can be mapped to the right by a 4×4 3D homography transform H

4D homogeneous coordinates (*i.e.*, appending an additional element "1" to each RGB vector). Assuming we relate I_s to I_t with a pixel-wise correspondence, we represent the RGBs of I_s and I_t as two $n \times 3$ matrices R_s and R_t, respectively, where n is the number of pixels. We also denote their corresponding $n \times 4$ homogeneous RGB matrices as $\dot{R}_s$ and $\dot{R}_t$. For instance, $\dot{R}_s$ can be converted to R_s by dividing its first 3 columns by its 4th column. Our 3D color homography color transfer model is proposed as:

$$\dot{R}_t \approx D\dot{R}_s H_{4\times 4} \tag{4}$$

$$R_t \approx D'\mathbf{h}(\dot{R}_s H_{4\times 4}) \tag{5}$$

where D is a $n \times n$ diagonal matrix of scalars (*i.e.*, exposures, but only used for estimating $H_{4\times 4}$) and $H_{4\times 4}$ is a 4×4 perspective color space mapping matrix, $\mathbf{h}()$ is a function that converts a $n \times 4$ homogeneous RGB matrix to a $n \times 3$ RGB matrix (*i.e.*, it divides the first 3 columns of $\dot{R}_s$ by its 4th column and removes the 4th column such that $R_s = \mathbf{h}(\dot{R}_s)$), D' is another $n \times n$ diagonal matrix of scalars (*i.e.*, shadings) for mean intensity mapping. A color transfer is decomposed into a 3D homography matrix $H_{4\times 4}$ and a mean intensity mapping diagonal matrix D'. The effect of applying the matrix H is essentially a perspective *color space distortion* (*e.g.*, Fig. 2). That is, we have a homography for RGB colors (rather than just chromaticities – (R/B, G/B)). D' adjusts mean intensity values (by modifying the magnitudes of the RGB vectors) to cancel the tonal difference between an input image and its color-transferred output image (*e.g.*, the right half of Fig. 1).

4 Image color transfer re-coding

In this section, we describe the steps for decomposing a color transfer between two registered images into the 3D color homography model components.

4.1 Perspective color space mapping

We solve for the 3D homography matrix $H_{4\times 4}$ in Eq. 4 by using alternating least squares (ALS) [9] as illustrated in Algorithm 1 where i indicates the iteration number. In each

---

1   $i = 0$, $\arg\min_{D^0} \left\| D^0 \dot{R}_s - \dot{R}_t \right\|_F$, $\dot{R}_s^0 = D^0 \dot{R}_s$;
2   **repeat**
3     $i = i + 1$;
4     $\arg\min_{H_{4\times 4}} \left\| \dot{R}_s^{i-1} H_{4\times 4} - \dot{R}_t \right\|_F$;
5     $\arg\min_D \left\| D\dot{R}_s^{i-1} H_{4\times 4} - \dot{R}_t \right\|_F$;
6     $\dot{R}_s^i = D\dot{R}_s^{i-1} H_{4\times 4}$;
7   **until** $\left\| \dot{R}_s^i - \dot{R}_s^{i-1} \right\|_F < \epsilon$ OR $i > n$;

Algorithm 1: Homography from alternating least squares

---

iteration, Step 4 (Algorithm 1) keeps D fixed (which was updated in Step 5 of a previous iteration) and finds a better $H_{4\times 4}$. Step 5 finds a better D using the updated $H_{4\times 4}$ fixed. The minimization in Steps 1, 4, and 5 are achieved by linear least squares. After these alternating estimation steps, we get a decreasing evaluation error for $\left\| \dot{R}_s^i - \dot{R}_s^{i-1} \right\|_F$. The assured convergence of ALS has been discussed in [26]. Practically, we may limit the number of iterations to $n = 20$. (Empirically, the error is not significant after 20 iterations.)

4.2 Mean intensity mapping

The tonal difference between an original input image and the color-transferred image is caused by the nonlinear operations of a color transfer process. We cancel this tonal difference by adjusting mean intensity (*i.e.*, scaling RGBs by multiplying D' in Eq. 5). To determine the diagonal scalars in D', we first propose a universal mean intensity-to-mean intensity mapping function $g()$ which is a smooth and monotonic curve fitted to the per-pixel mean intensity values (*i.e.*, mean values of RGB intensities) of the two images. As opposed to the unconstrained mean intensity-to-shading mapping adopted in [11], we enforce monotonicity and smoothness in our optimization which avoids halo-like artifacts (due to sharp or non-monotonic tone change [6]). The mapping function g is fitted by minimizing the following function:

$$\underset{g}{\arg\min} \quad \left\| \underline{y} - g(\underline{x}) \right\|^2 + p \int \left\| g''(t) \right\|^2 dt \tag{6}$$

$$\text{subject to} \quad g'(t) \geq 0 \quad \text{and} \quad 0 \leq g(t) \leq 1.$$

where the first term minimizes the least-squares fitting error and the second term enforces a smoothness constraint for the fitted curve. $\underline{x}$ and $\underline{y}$ are assumed in [0, 1]. The curve is smoother when the smoothness factor p is larger ($p = 10^{-5}$ by default). $\underline{x}$ and $\underline{y}$ are the input and reference mean intensity vectors. The mapping function $g()$ is implemented as a lookup table which is solved by quadratic programming [4]. (See "Appendix" 1.) Figure 1 shows an example of the computed function $g()$.

Let the operator $\mathbf{diag}(\underline{x})$ return a diagonal matrix with components of $\underline{x}$ along its diagonal. Given the estimated mean intensity mapping function $g()$, the diagonal scalar matrix D' (in Eq. 5) is updated as follows:

$$\underline{b} = \frac{1}{3}\mathbf{h}(\dot{R}_s H_{4\times 4})\begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} \tag{7}$$

$$D' = \mathbf{diag}(g(\underline{b}))\mathbf{diag}^{-1}(\underline{b})$$

where $\underline{b}$ is the input mean intensity vector of the 3D color space-mapped RGB values (*i.e.*, $\mathbf{h}(\dot{R}_s H_{4\times 4})$). Because this
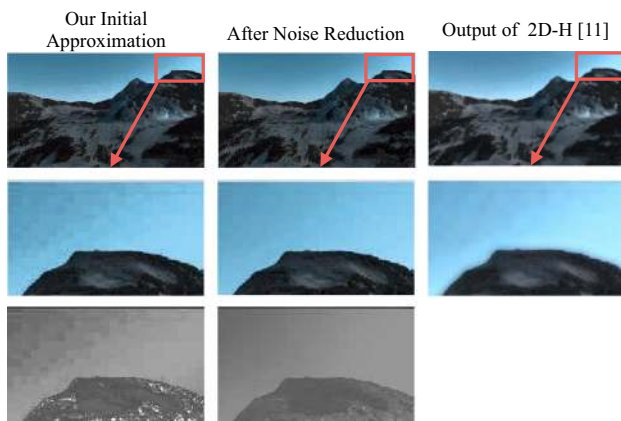
Our Initial Approximation   After Noise Reduction   Output of 2D-H [11]

Fig. 3 Minor noise reduction: Some JPEG block artifacts of our initial approximation are reduced via mean intensity mapping noise reduction step. Compared with our final approximation result (middle), the output of [11] contains significant blur artifacts at the boundary of the hill. Row 2 shows the magnified area in Row 1. Row 3 shows the shading image $I_{D'}$ of the magnified area

step only changes the magnitude of an RGB vector, the physical chromaticity and hue are preserved.

4.3 Mean intensity mapping noise reduction

Perhaps because of the greater compression ratios in images and videos, we found that, even though the mean intensity is reproduced as a smooth and monotonic tone curve, some minor spatial artifacts could—albeit rarely—be introduced. We found that the noise can be amplified in the mean intensity mapping step and the component D' (in Eq. 5) absorbs most of the spatial artifacts (*e.g.*, JPEG block edges). To reduce the potential gradient artifacts, we propose a simple noise reduction. Because D' scales each pixel individually, we can visualize D' as an image denoted as $I_{D'}$. We remove the spatial artifacts by applying a joint bilateral filter [17] which spatially filters the scale image $I_{D'}$ guided by the source image I_s such that the edges in $I_{D'}$ are similar to the edges in the mean intensity channel of I_s. Figure 3 shows the effect of the monotonic smooth mean intensity mapping and its mapping noise reduction. Although it is often not necessary to apply this noise reduction step, we have always included it as a "safety guarantee."

4.4 Results

In our experiments, we assume that we have an input image and an output produced by a color transfer algorithm. Because the input and output are matched (*i.e.*, they are in perfect registration), we can apply Algorithm 1 directly.

In the experiments which follow, we call our method—3D homography color transform $+$ mean intensity mapping— "3D-H." Similarly, the 2D homography approach for color

transfer re-coding [11] is denoted as "2D-H." We first show some visual results of color transfer approximations of [16, 20, 21, 23] in Fig. 4. Our 3D color homography model offers a visually closer color transfer approximation.

Although a public dataset for color transfer re-coding was published in [11], it contains a limited sample size. In this work, we use a new dataset[1] In Table 1, we quantitatively evaluate the approximation accuracy of the 3 state-of-the-art algorithms [11,13,18] by the error between the approximation result and the original color transfer result. The results are the averages over the 200 test images. The method [13] is tested by using a common second-order polynomial (which avoids over-fitting). We adopt PSNR (peak signal-to-noise ratio) and SSIM (structural similarity) [27] as the error measurements. Acceptable values for PSNR and SSIM are usually considered to be, respectively, over 20 dB and 0.9. Table 1 shows that 3D-H is generally better than the other compared methods for both PSNR and SSIM metrics. To further investigate the statistical significance of the evaluation result, we run a one-way ANOVA to verify that the choice of our model has a significant and positive effect on the evaluation metrics (*i.e.*, PSNR and SSIM). In our test, we categorize all evaluation numbers into four groups according to associated color transfer re-coding method. Table 2 shows the post hoc tests for one-way ANOVA where the choice of color transfer approximation method is the only variable. We obtained the overall p-values < 0.001 for both PSNR and SSIM which indicate the choice of color transfer re-coding method has a significant impact on the color transfer approximation result. In addition, we run a post hoc analysis and found near 0 p-values when comparing 3D-H with all 3 other methods. This further confirms that the difference in performance of 3D-H is significant. Our test also shows that the difference between 2D-H [11] and Poly [13] is not significant. Our 3D color homography model produces the best result overall.

5 Applications

In this section, we demonstrate three applications of our color transfer re-coding method.

5.1 Color transfer acceleration

More recent color transfer methods usually produce higher-quality outputs, however, at the cost of more processing

---

[1] The dataset will be made public for future comparisons. with a significant larger size—200 color transfer images—so that the quality of color transfer re-coding can be thoroughly evaluated. Each color transfer image pair also comes with the color transfer results of 4 popular methods [16,20,21,23].

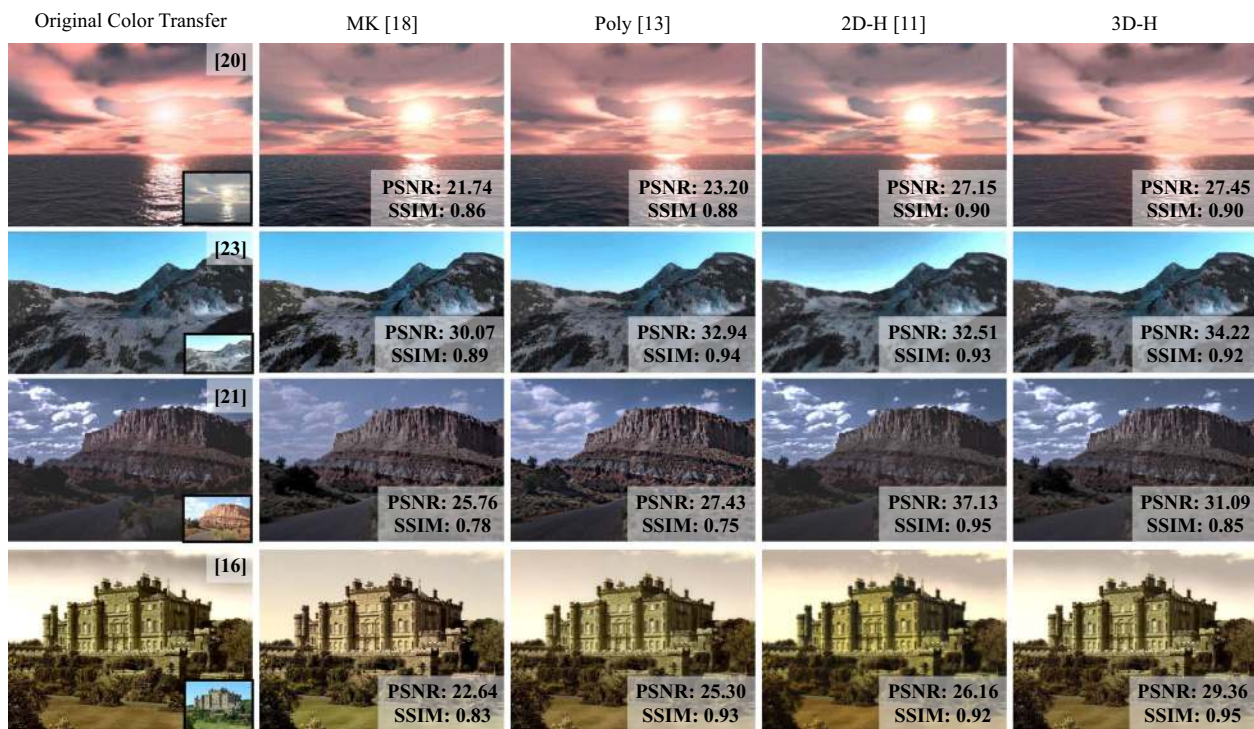**Fig. 4** Visual result of 4 color transfer approximations (rightmost 4 columns). The original color transfer results are produced by the methods cited at the top right of the images in the first column. The original input images are shown at the bottom right of the first column. Please also check our supplementary material for more examples (http://goo.gl/n6L93k)

Table 1 Mean errors between the original color transfer result and its approximations by 4 popular color transfer methods

	Nguyen [16]	Pitie [20]	Pouli [21]	Reinhard [23]
PSNR (peak signal-to-noise ratio)				
MK [18]	23.24	22.76	22.41	25.21
Poly [13]	25.54	25.08	27.17	28.27
2D-H [11]	24.59	25.19	27.22	28.24
3D-H	**27.34**	**26.65**	**27.55**	**30.00**
SSIM (structural similarity)				
MK [18]	0.88	0.85	0.81	0.85
Poly [13]	0.91	0.89	0.85	0.88
2D-H [11]	0.86	0.86	**0.90**	0.92
3D-H	**0.93**	**0.90**	0.89	**0.93**

The best results are made bold

time. Methods that produce high-quality images and are fast include the work of Gharbi et al. [10] who proposed a general image manipulation acceleration method—named transform recipe (TR)—designed for cloud applications. Based on a downsampled pair of input and output images, their method approximates the image manipulation effect according to changes in luminance, chrominance, and stack levels. Another fast method by Chen et al. [3] approximates the effect of many general image manipulation procedures by convolutional neural networks (CNNs). While their approach significantly reduces the computational time for some complex operations, it requires substantial amounts of samples

for training a single image manipulation. In this subsection, we demonstrate that our re-coding method can be applied as an alternative to accelerate a complex color transfer by approximating its color transfer effect at a lower scale. We approximate the color transfer in the following steps: (1) We supply a thumbnail image (40×60 in our experiment) to the original color transfer method and obtain a thumbnail output; (2) Given the pair of lower-resolution input and output images, we estimate a color transfer model that approximates the color transfer effect; (3) We then process the higher-resolution input image by using the estimated color transfer model and obtain a higher-resolution output which looks very

Table 2 Post hoc tests for one-way ANOVA on errors between the original color transfer result and its approximations

Method A	Method B	p-value
PSNR overall p-value < 0.001		
MK [18]	Poly [13]	< 0.001
MK [18]	2D-H [11]	< 0.001
MK [18]	3D-H	< 0.001
Poly [13]	2D-H [11]	0.95
Poly [13]	3D-H	< 0.001
2D-H [11]	3D-H	< 0.001
SSIM overall p-value < 0.001		
MK [18]	Poly [13]	< 0.001
MK [18]	2D-H [11]	< 0.001
MK [18]	3D-H	< 0.001
Poly [13]	2D-H [11]	0.48
Poly [13]	3D-H	< 0.001
2D-H [11]	3D-H	< 0.001

close to the original higher-resolution color transfer result without acceleration.

In our experiment, we choose two computationally expensive methods [16,20] as the inputs and we compare our performance (MATLAB implementation) with a state-of-the-art method TR [10] (Python implementation). Figure 5 shows the output comparisons between the original color transfer results and the acceleration results. The results indicate that our re-coding method can significantly reduce the computational time ($25\times$ to $30\times$ faster depending on the speed of original color transfer algorithm and the input image resolution) for these complicated color transfer methods while preserving color transfer fidelity. Compared with TR [10], our method produces similar quality of output for global color transfer approximation, however, at a much reduced cost of computation (about $10\times$ faster). Although TR [10] is less efficient, it is also worth noting that TR supports a wider range of image manipulation accelerations which include non-global color change.

5.2 Color transfer artifact reduction

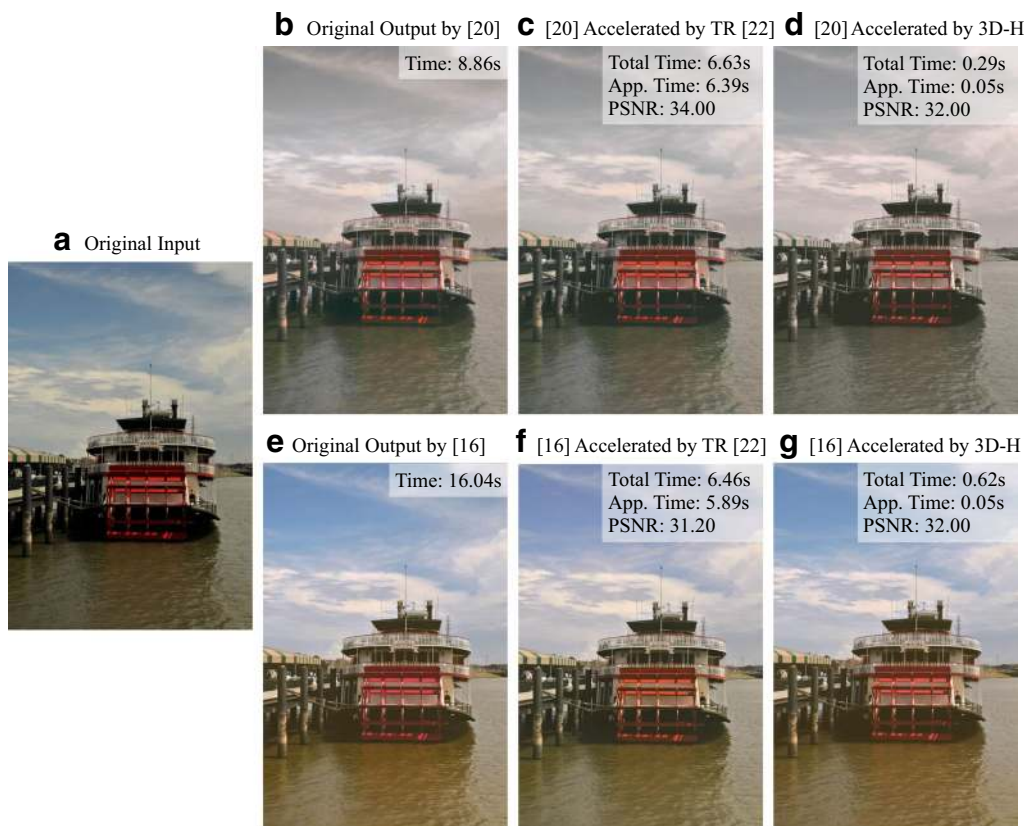Color transfer methods often produce artifacts during the color matching process. Here we show that our color trans-

Fig. 5 Color transfer acceleration. The color transfers [16,20] of an original image (**a**) are accelerated by our 3D-H color transfer re-coding method and a state-of-the-art method transform recipe (TR) [10]. The top right labels in (**b**) and (**e**) show the original color transfer time. The top right labels in (**c**, **d**) and (**f**, **g**) show the improved processing time and the measurements of similarity (PSNR) to the original color transfer output. "App. Time" indicates time for color transfer approximation only
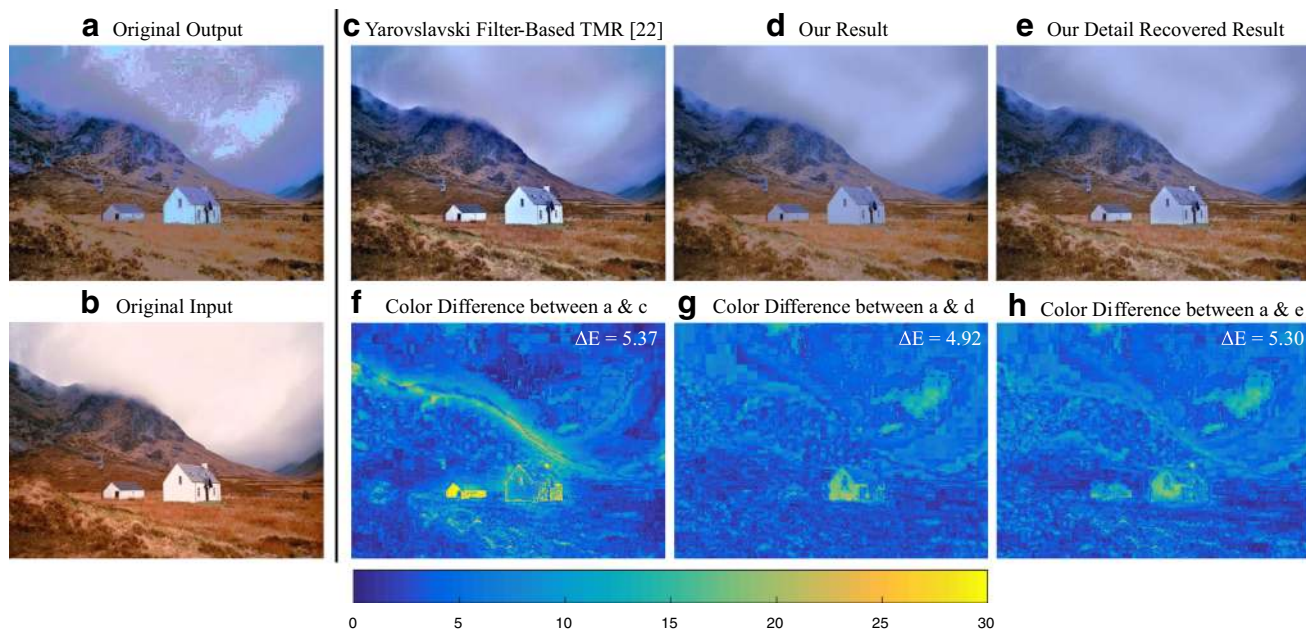
Fig. 6 Artifact reduction comparison based on an original image (**b**) and its defective color transfer output (**a**). **c** is an artifact reduction result by a state-of-the-art method— transportation map regularization (TMR) [22]. **d** is our re-coding result where the artifacts are indirectly smoothed at the layer of shading scales. **e** is our alternative enhancement result which makes its higher-frequency detail similar to the original input image. **f**, **g** CIE 1976 Color Difference ΔE [24] visualization of **c**–**e** where the average color difference is labeled at the top right corner. A lower ΔE indicates a closer approximation to (**a**)

fer re-coding method can be applied as an alternative to reduce some undesirable artifacts (*e.g.*, JPEG compression block edges). As discussed in Sect. 4.3, the color transfer artifacts are empirically "absorbed" in the shading scale component. Therefore, we can simply filter the shading scale layer by using the de-noising step described in Sect. 4.3. Figure 6 shows our artifact reduction result where we also compare with a state-of-the-art color transfer artifact reduction method—transportation map regularization (TMR) [22]. Compared with the original color transfer output, our processed result better preserves its contrast and color similarity (*e.g.*, the grass). Meanwhile, it also removes the undesirable artifacts. And, depending on an individual user's personal preference, the result of TMR could also be preferred since TMR makes the contrast of its result more similar to the original input. While one of our goals is to make the processed colors close to the original *output* image's colors, it is also possible to transfer the details of the original *input* image to our result using the detail transfer method suggested in [5]. The result of this input detail recovery is shown in Figure 6e.

5.3 Color-robust image stitching

The input images for image stitching are not always taken by the same camera or under the same illumination con-

ditions. The camera's automatic image processing pipeline also modifies the colors. Direct image stitching without color correction may therefore leave imperfections in the final blending result. Since the color change between images of different views is unknown but photo-realistic, our color transfer approximation model can be applied to address this color inconsistency problem. Figure 7 shows an example of a color-robust image stitching using our color transfer re-coding method where 2 input images taken by 2 different cameras and in different illuminations are supplied for image stitching. In our color-robust procedure, we first register these two images and find the overlapping pixels. With the per-pixel correspondences, we estimate a 3D color homography color transfer model that transfers the colors of the first image to the second image's. We then apply the estimated color transfer model to correct the first image. Finally, the corrected first input image and the original second image are supplied to the image stitching software AutoStitch [2]. Although the multi-band blending proposed in [2] provides a smooth transition at the stitching boundary, the color difference between the two halves is still noticeable (especially for the sky and tree colors) in the original stitching result. After our color alignment process, the colors of the two halves look more homogeneous. We also compare our method with a local color correction method—gain compensation [2].
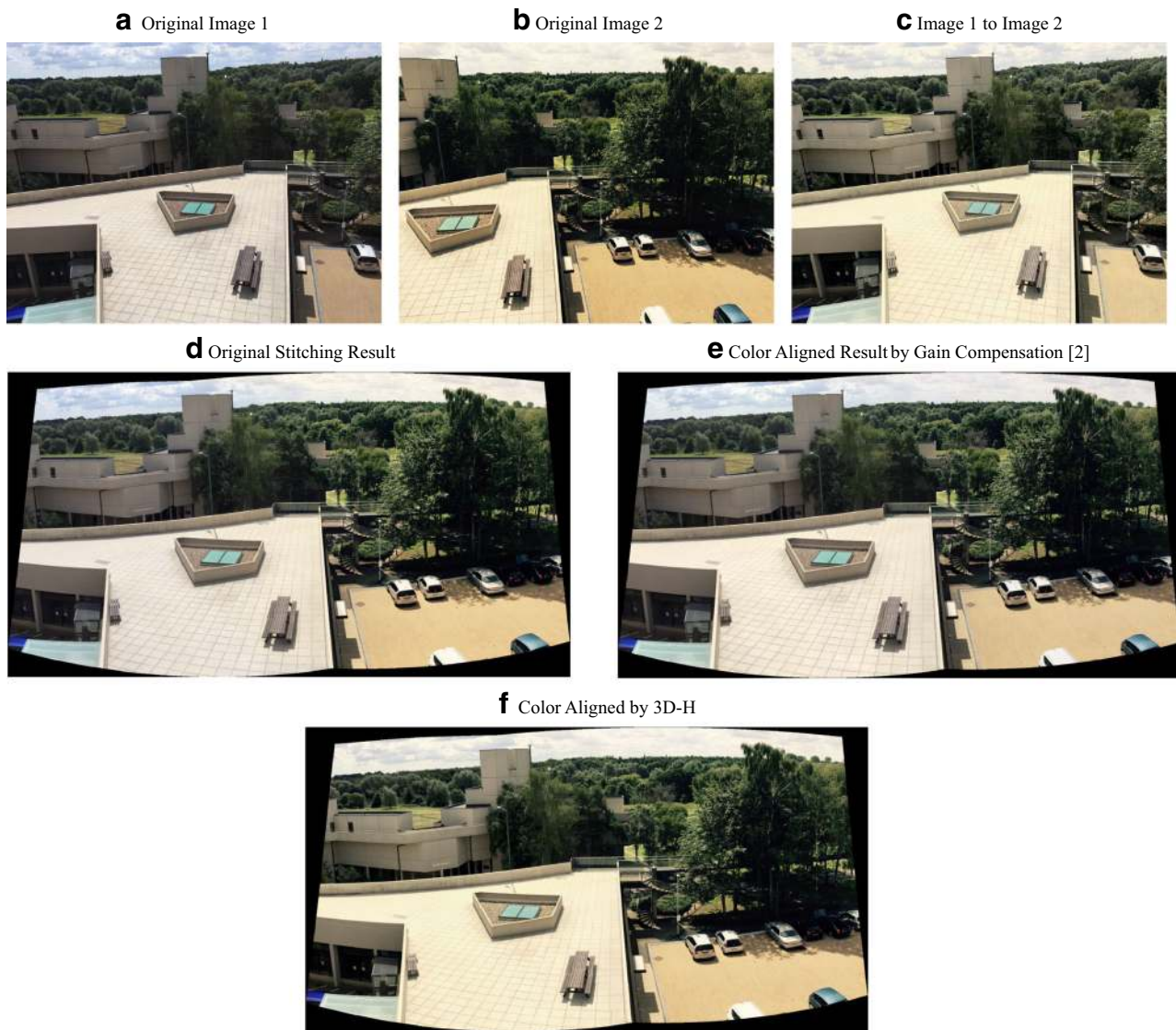
a Original Image 1   b Original Image 2   c Image 1 to Image 2

d Original Stitching Result   e Color Aligned Result by Gain Compensation [2]

f Color Aligned by 3D-H

Fig. 7 Color-robust image stitching. **a**, **b** are the original images taken by 2 different cameras. **c** is the color-transferred results from Image 2 to Image 1. **d** shows the original image stitching result without color transfer. **e**, **f** are the image stitching results matched to the colors of Image 2 by using different color alignment algorithms (The color transition in **f**) is better, which is especially visible in the sky.)

6 Conclusion

In this paper, we have shown that a global color transfer can be approximated by a combination of 3D color space mapping and mean intensity mapping. Our experiments show that the proposed color transfer model approximates well for many photo-realistic color transfer methods as well as unknown global color change in images. We have demonstrated three applications for color transfer acceleration, color transfer artifact reduction, and color-robust image stitching.

Appendix

Quadratic programming solution for function $g()$

In this subsection, we explain the details of solving Eq. 6 using quadratic programming. In our solution, we try to build

a 1000-entry lookup table (LUT) that approximates the mean intensity mapping function $g()$. We first quantize input mean intensity values $\underline{x}$ into 1000 uniform levels distributed within $[0, 1]$. To construct a deterministic one-to-one mean intensity mapping, we have two issues to address before the actual optimization: (1) Among the training mean intensity pairs, it is likely that each unique input mean intensity level x_i (where i is an index) is mapped to more than one reference output mean intensity values. To address this, we define the mean of all reference output mean intensity values of x_i as a unique reference output mean intensity value y_i for x_i. (2) Each unique input level x_i may have a different number of (or none) input training data pairs. To balance data fitness in the later weighted least-squares optimization, we construct a weight vector $\underline{w}$ whose element w_i is the count of training data associated with x_i which can be 0 if there are no data associated with the quantization level. (See Fig. 1 for an example.)

Suppose that the elements of the 1000-vectors $\underline{x}$ and $\underline{y}$ are simultaneously sorted in an ascending order based on the elements of $\underline{x}$ (*i.e.*, $x_i < x_{i+1}$), we can approximate the minimization by finite differentiation and rewrite Eq. 6 as:

$$\operatorname*{argmin}_{g()} \left(\sum_{i=1}^{n} w_i \|y_i - g(x_i)\|^2 \right) +$$
$$p \left(\sum_{i=3}^{n} \left\| \frac{g'(x_i) - g'(x_{i-1})}{(n-1)^{-1}} \right\|^2 \right)$$
$$\text{s.t.} \quad \forall i \in \{2, 3, \ldots n\} \quad g(x_i) - g(x_{i-1}) \geq 0 \quad (8)$$
$$0 \leq g(x_{i-1}), g(x_i) \leq 1$$
$$\text{where} \quad g'(x_i) \approx \frac{g(x_i) - g(x_{i-1})}{x_i - x_{i-1}} = \frac{g(x_i) - g(x_{i-1})}{(n-1)^{-1}}$$
$$n = 10^3.$$

where n is the number of unique input mean intensity values $\underline{x}$ (for training). Further, since the mean intensity mapping is a positive-to-positive number mapping, we can define the mean intensity mapping function as $g(x_i) = \alpha_i x_i$ where α_i is a positive scalar of a scalar vector $\underline{\alpha}$. We can then rewrite Eq. 8 in matrix multiplication form as:

$$\operatorname*{argmin}_{\underline{\alpha}} \left(\underline{y}^\mathsf{T}\mathbf{diag}(\underline{w})\underline{y} - 2\underline{y}^\mathsf{T}\mathbf{diag}(\underline{w})\mathbf{diag}(\underline{x})\underline{\alpha} + \right.$$
$$+ (\mathbf{diag}(\underline{x})\underline{\alpha})^\mathsf{T}\mathbf{diag}(\underline{w})(\mathbf{diag}(\underline{x})\underline{\alpha})$$
$$+ p(D_{g^2}\underline{\alpha})^\mathsf{T}(D_{g^2}\underline{\alpha})$$
$$\text{s.t.} \quad D_n(\underline{x})\underline{\alpha} \geq \underline{\mathbf{0}}$$
$$\underline{\mathbf{0}} \leq \underline{x} \leq \underline{\mathbf{1}} \quad (9)$$
$$\text{where} \quad D_n = \begin{bmatrix} -1 & 1 & & & \\ & -1 & 1 & & \\ & & \ddots & \ddots & \\ & & & -1 & 1 \end{bmatrix}_{(n-1)\times n}$$
$$\underline{d}_{g^1} = (n-1)^{-1}D_n\mathbf{diag}(\underline{x})\underline{\alpha}$$
$$\underline{d}_{g^2} = (n-1)^{-1}D_{n-1}\underline{d}_{g^1}$$

In Eq. 9, D_n is defined as a $(n-1) \times (n)$ sparse forward difference matrix (where n is a variable size), $\underline{\mathbf{0}}$ and $\underline{\mathbf{1}}$ are, respectively, a $(n-1)$ zero vector and a $(n-1)$ one vector, $\underline{d}_{g^1}$ and $\underline{d}_{g^2}$ are, respectively, the first-order and second-order finite difference vectors of $g()$, with respect to $\underline{x}$.

Finally, we convert the minimization function of Eq. 9 into a quadratic form:

$$\operatorname*{argmin}_{\underline{\alpha}} \tfrac{1}{2}\underline{\alpha}^\mathsf{T} M \underline{\alpha} + f^\mathsf{T}\underline{\alpha}$$
$$\text{where} \quad M = 2(pT^\mathsf{T}T + \mathbf{diag}(\underline{w})\mathbf{diag}^2(\underline{x})) \quad (10)$$
$$T = (n-1)^{-2}D_{n-1}D_n\mathbf{diag}(\underline{x})$$
$$f = -2\underline{y}^\mathsf{T}\mathbf{diag}(\underline{w})\mathbf{diag}(\underline{x})$$

where M is a sparse Hessian matrix and f is a linear term. With the pre-computed M, f, and the inequality conditions, we can apply quadratic programming to solve for $\underline{\alpha}$. The mapped one-to-one deterministic output mean intensity value is computed as $\mathbf{diag}(\underline{x})\underline{\alpha}$.

Implementation details

We implemented the proposed algorithm in MATLAB. In ALS (Algorithm 1), we exclude the undersaturated pixels with zero RGBs as they contain no useful color information. Practically, we also limit the maximum iteration number of ALS to 20, beyond which we do not find any significant fidelity improvement.

References

1. An, X., Pellacini, F.: User-controllable color transfer. Comput. Graph. Forum **29**(2), 263–271 (2010)
2. Brown, M., Lowe, D.G.: Automatic panoramic image stitching using invariant features. Int. J. Comput. Vis. **74**(1), 59–73 (2007)
3. Chen, Q., Xu, J., Koltun, V.: Fast image processing with fully-convolutional networks. In: IEEE International Conference on Computer Vision (ICCV). (2017)
4. Coleman, T.F., Li, Y.: A reflective newton method for minimizing a quadratic function subject to bounds on some of the variables. SIAM J. Optim. **6**(4), 1040–1058 (1996)
5. Durand, F., Dorsey, J.: Fast bilateral filtering for the display of high-dynamic-range images. ACM Trans. Gr. **21**, 257–266 (2002)
6. Fattal, R., Lischinski, D., Werman, M.: Gradient domain high dynamic range compression. ACM Trans. Gr. (TOG) **21**, 249–256 (2002)
7. Finalyson, G.D., Gong, H., Fisher, R.B.: Color homography color correction. In: Color and Imaging Conference. Society for Imaging Science and Technology (2016)
8. Finlayson, G.D., Gong, H., Fisher, R.B.: Color homography: theory and applications. IEEE Transactions on Pattern Analysis and Machine Intelligence (2017). To appear
9. Finlayson, G.D., Mohammadzadeh Darrodi, M., Mackiewicz, M.: The alternating least squares technique for non-uniform intensity color correction. Color Res. Appl. **40**(3), 232–242 (2015)

10. Gharbi, M., Shih, Y., Chaurasia, G., Ragan-Kelley, J., Paris, S., Durand, F.: Transform recipes for efficient cloud photo enhancement. Trans. Gr. **34**(6), 228 (2015)
11. Gong, H., Finlayson, G.D., Fisher, R.B.: Recoding color transfer as a color homography. In: British Machine Vision Conference. BMVA (2016)
12. Hwang, Y., Lee, J.Y., Kweon, I.S., Kim, S.J.: Color transfer using probabilistic moving least squares. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 3342–3349. (2014)
13. Ilie, A., Welch, G.: Ensuring color consistency across multiple cameras. In: IEEE International Conference on Computer Vision, vol. 2, pp. 1268–1275. (2005)
14. Maloney, L.: Evaluation of linear models of surface spectral reflectance with small numbers of parameters. J. Opt. Soc. Am. A **3**, 1673–1683 (1986)
15. Marimont, D., Wandell, B.: Linear models of surface and illuminant spectra. J. Opt. Soc. Am. A **9**(11), 1905–1913 (1992)
16. Nguyen, R.M.H., Kim, S.J., Brown, M.S.: Illuminant aware gamut-based color transfer. Comput. Gr. Forum **33**(7), 319–328 (2014)
17. Paris, S., Durand, F.: A fast approximation of the bilateral filter using a signal processing approach. In: European Conference on Computer Vision. pp. 568–580 (2006)
18. Pitié, F., Kokaram, A.: The linear monge-kantovitch linear colour mapping for example-based colour transfer. In: European Conference on Visual Media Production (IET). pp. 1–9. (2007)
19. Pitie, F., Kokaram, A., Dahyot, R.: N-dimensional probability density function transfer and its application to color transfer. Int. Conf. Comput. Vis. **2**, 1434–1439 (2005)
20. Pitié, F., Kokaram, A.C., Dahyot, R.: Automated colour grading using colour distribution transfer. Comput. Vis. Image Underst. **107**(1–2), 123–137 (2007)
21. Pouli, T., Reinhard, E.: Progressive histogram reshaping for creative color transfer and tone reproduction. In: ACM International Symposium on Non-Photorealistic Animation and Rendering, pp. 81–90. New York, USA (2010)
22. Rabin, J., Delon, J., Gousseau, Y.: Removing artefacts from color and contrast modifications. IEEE Trans. Image Process. **20**(11), 3073–3085 (2011)
23. Reinhard, E., Ashikhmin, M., Gooch, B., Shirley, P.: Color transfer between images. IEEE Comput. Gr. Appl. **21**(5), 34–41 (2001)
24. Robertson, A.R.: The cie 1976 color-difference formulae. Color Res. Appl. **2**(1), 7–11 (1977)
25. Tai, Y.W., Jia, J., Tang, C.K.: Local color transfer via probabilistic segmentation by expectation-maximization. In: IEEE Conference on Computer Vision and Pattern Recognition. pp. 747–754. (2005)
26. Takane, Y., Young, F.W., De Leeuw, J.: Nonmetric individual differences multidimensional scaling: an alternating least squares method with optimal scaling features. Psychometrika **42**(1), 7–67 (1977)
27. Wang, Z., Bovik, A.C., Sheikh, H.R., Simoncelli, E.P.: Image quality assessment: from error visibility to structural similarity. Trans. Image Process. **13**(4), 600–612 (2004)
28. Wu, F., Dong, W., Kong, Y., Mei, X., Paul, J.C., Zhang, X.: Content-based colour transfer. Comput. Graph. Forum **32**(1), 190–203 (2013)

Han Gong He received his Ph.D. from the University of Bath. He is currently a Senior Research Associate at the University of East Anglia. His research interests include computer vision and computational photography.

Graham D. Finlayson He is currently a Professor of Computing Sciences with the University of East Anglia. His research interests span 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.

Robert B. Fisher He has been an academic in the School of Informatics at University of Edinburgh since 1984 and a full Professor since 2003. He received his Ph.D. from University of Edinburgh (1987). His previous degrees are a B.S. with honors (Mathematics) from California Institute of Technology (1974) and a MS (Computer Science) from Stanford University (1978).

Fufu Fang He is a Ph.D. student in School of Computing Science in University of East Anglia, UK. His research interests include computer vision, especially in color correction.