Pacific Graphics 2014
J. Keyser, Y. J. Kim, and P. Wonka
(Guest Editors)

Volume 33 (2014), Number 7

Illuminant Aware Gamut-Based Color Transfer

R. M. H. Nguyen[1] , S. J. Kim[†2] and M. S. Brown[1]

[1]National University of Singapore
[2]Yonsei University

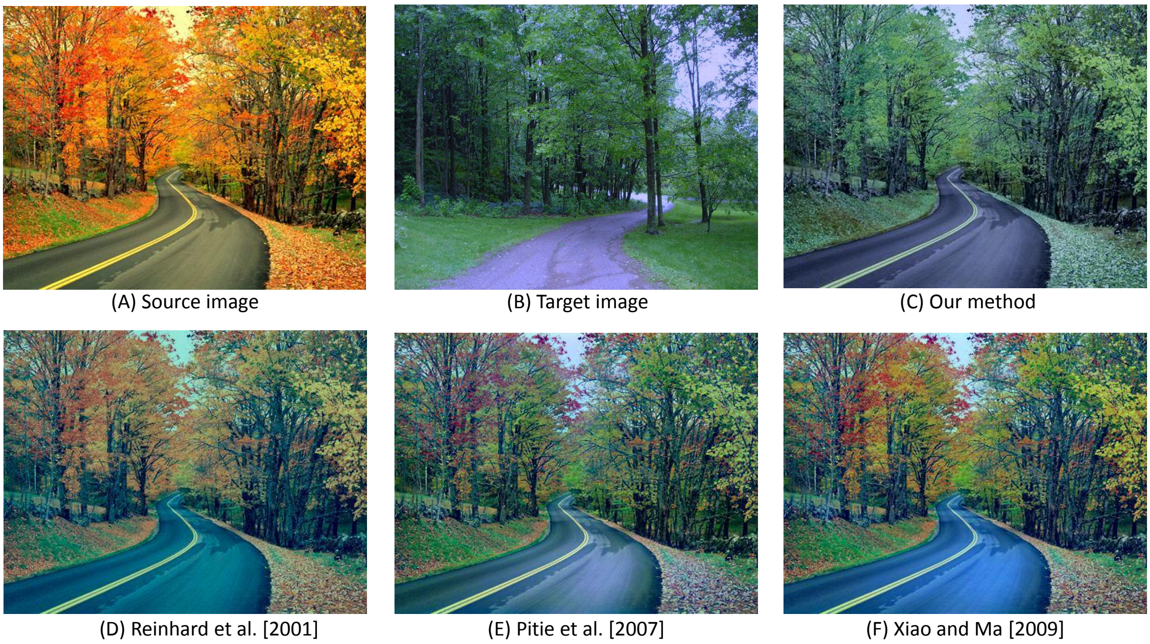

Figure 1: *This figure compares color transfer results of several methods. Our method incorporates information about the source and target scene illuminants and constrains the color transfer to lie within the color gamut of the target image. Our resulting image has a more natural look and feel than existing methods.*

Abstract

This paper proposes a new approach for color transfer between two images. Our method is unique in its consideration of the scene illumination and the constraint that the mapped image must be within the color gamut of the target image. Specifically, our approach first performs a white-balance step on both images to remove color casts caused by different illuminations in the source and target image. We then align each image to share the same 'white axis' and perform a gradient preserving histogram matching technique along this axis to match the tone distribution between the two images. We show that this illuminant-aware strategy gives a better result than directly working with the original source and target image's luminance channel as done by many previous methods. Afterwards, our method performs a full gamut-based mapping technique rather than processing each channel separately. This guarantees that the colors of our transferred image lie within the target gamut. Our experimental results show that this combined illuminant-aware and gamut-based strategy produces more compelling results than previous methods. We detail our approach and demonstrate its effectiveness on a number of examples.

Categories and Subject Descriptors (according to ACM CCS): I.3.m [Computer Graphics]: Computational Photography— Image Processing

† Corresponding author

1. Introduction

Color transfer is a process of manipulating the color values of a source image such that it shares the same "look and feel" of a specified target image. Fig. 1 shows a typical case of color transfer where the goal is to make the source image A appear more like the target image B. There has been a wide range of approaches targeting color transfer (e.g. [RAGS01, PKD07, XM09]). Fig. 1-(D-F) shows the results from the representative methods. While these approaches use various strategies for the color transfer process, they all share the common theme of manipulating the input image's color distribution in a way that better fits the target image's distribution. Interestingly, these techniques often perform the color manipulation in a manner that is agnostic to information specific to the scene content. In addition, many techniques do not prevent the color mapping from producing new colors in the transferred image that are not in the color gamut of the target image.

In this paper, we propose a method that considers the scene illumination in the color transfer process and constrains the result to fit within the target image's gamut. The consideration for the scene illumination is motivated by the observation that the color transfer problem shares similarities to the problem of color constancy and white-balancing. An image captured by a camera is an integrated signal resulting from the camera's sensitivity of the spectral scene content and scene illumination. Scene illumination can have a significant effect on the overall RGB values of an image, introducing noticeable color casts that can give an image a very different appearance. While color casts are commonly removed using white-balancing, they are sometimes allowed to remain in an image to give a particular look and feel, e.g. creating a warm or cool image. In fact, for many pairs of source/target images shown in prior color transfer literature, one of the major factors attributing to the source/target color differences is due to the color cast caused by the illumination. As such, we explicitly consider the scene illumination in our color mapping process.

Our motivation for constraining the result of the color transfer is even clearer. Out-of-gamut colors can give a strange appearance to the target image as shown by some of the results in Fig. 1. This can be avoided by explicitly enforcing the color transform to produce a result that lies within the target image's color gamut. As demonstrated by our results, this combined illuminant-aware and gamut-based method produces results that better preserve the look and feel of the target image. We explain each procedure in detail and show its effectiveness on a wide variety of input images.

The remainder of this paper is as follows: Section 2 describes related works; Section 3 overviews our approach and explains how we incorporate illumination estimation into the color mapping process and constrain the gamut; Section 4 shows experiments to evaluate our procedure and we conclude the paper with a discussion and summary in Section 5.

2. Related Work

While there are existing color transfer methods that focus on transferring color to match colors between multiple images of the same scene to enhance photo-consistency (e.g. [OSS11, HSGL13, HLKK14]), we focus on color transfer between two totally different images. One of the earliest known works in this area is by Reinhard et al. [RAGS01] who introduced a simple and efficient framework for color transfer. Their method matches the means and the standard deviations of the global color distributions of the two images in the Lab color space.

Pitie et al. [PKD05] presented a color transfer method based on the iterative use of a 1D histogram matching technique. This approach uses a 3D rotation matrix to rotate the source and the target images to a certain axis before applying the iterative 1D histogram matching for the color transfer. The rotation matrix can be chosen randomly over all possible angular combinations. In [PKD07], Pitie et al. improved their work by adding a gradient preservation constraint as a post-processing step to reduce artifacts.

Xiao and Ma [XM06] proposed to handle color transfer in the RGB space. Their method uses linear transforms (translation, scale, and rotation) to align the two gamuts of the source and target images. These linear transforms are computed according to the mean and the covariance matrix of the two images. In [XM09], Xiao and Ma extended their work to map the histograms of two images together and included a step to preserve the gradient of the source image.

Pouli and Reinhard [PR11] have recently introduced a color transfer method based on a histogram matching similar to [PKD05]. Their updated approach uses progressive histogram matching and operates in the Lab color space. Their method introduces a parameter to control the level of color transfer, which allows a partial color transfer between a pair of images. Additionally, the paper addresses the problem of different dynamic range between the source and the target image.

There are also several methods that apply local color transfers rather than using a global transfer. For example, Tai et al. [TJT05] segmented the image through an EM framework and applied local color transfer for each of the segments. An and Pellacini [AP10] present an interactive tool where a user selects local regions to perform the color transfer.

Another area that is related to this paper is computational color constancy. The human visual system has an innate ability, termed color constancy, to perceive colors under different illumination in a constant manner [KSK02]. For cameras, however, color changes due to illumination must be corrected through post-processing in a white-balancing step which attempts to estimate the illumination in the scene. Based on the estimated illumination, the color of the image is transformed such that the illumination direction lies along

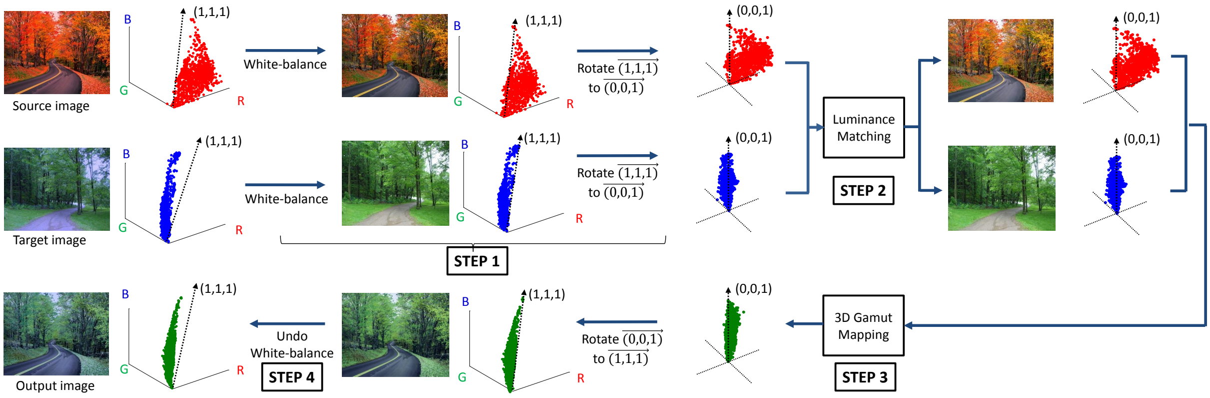

Figure 2: *This figure shows our color transfer framework. Step 1: the "white" points of the source and target images are matched together using white-balancing. These are then rotated along the $(0,0,1)$ axis. Step 2: a gradient preserving technique is applied on the luminance channel (white-axis) of the source image. Step 3: the 3D gamut of the source image is aligned to that of the target image. Step 4: the image's white point is transformed back to the target image white point (i.e the white-balancing is undone).*

the achromatic line in the color space (for excellent surveys on color constancy see [GGVDW11, BMCF02]).

3. Our Approach

Consider a pair of input images: a source image I_s and a target image I_t. The goal of color transfer is to obtain an output image I_o having the content from the source image I_s and the color palette from the target image I_t. Fig. 2 shows four steps of our color transfer approach. First, the "white" points of the source and target images are matched together using white-balancing. Then, a gradient preserving matching technique is applied on the luminance channel of the source image. Next, the 3D gamut of the source image is aligned to that of the target image. Finally, the color transferred output I_o is obtained after undoing the white-balance according to the computed white point of the target image. These procedures are explained in details in the following.

3.1. Matching White Points

To take the illumination into account, the first step in our color transfer approach is to run white-balancing on both the source and the target images. This step starts by computing the illumination color (white point) and dividing each pixel's color by the scene's illuminant color. After removing the illumination cast, scene content that is assumed to be achromatic should lie along the white line in the RGB space. Computing the white points (r_{ws}, g_{ws}, b_{ws}) and (r_{wt}, g_{wt}, b_{wt}) for the source and target images can be achieved by using any existing state of the art white-balancing techniques [CHZ12, CGZ07, GGVDW12]. In this paper, we used the weighted Grey-Edge algorithm proposed in [GGVDW12] for estimating the white point. Dividing each pixel's value by the computed white point color will map the "white" point of each image to the RGB value $(1,1,1)$. The white-balancing step serves as an illumination color normalization and the vector from $(0,0,0)$ to $(1,1,1)$ represents the shades of gray color, meaning that this vector can serve as a luminance channel.

Fig. 3 shows the importance of this step in removing the color cast and bias in an image. The figure shows a scene captured with a color chart inserted. The last row of the chart contains a series of achromatic patches ranging from white to black (pure diffuse white material). The scene has been rendered under several different color temperatures. Only the white-balanced image shows that the achromatic patches have no color bias. This can be seen by the convergence of the color histograms for the white-balanced image. Once white-balanced, the scene's white content is now aligned with the $(0,0,0)$ and $(1,1,1)$ vector.

It is worth noting that standard RGB (sRGB) images have a gamma correction applied to compensate for nonlinearities in display devices. White-balancing is technically applied in the linearized sRGB space before this gamma correction. We tested our approach using white-balance on both gamma-corrected sRGB and linear RGB (i.e. undo the gamma correction step). The results from both these approaches were quite similar. Results shown in this paper were obtained using the original sRGB images.

To facilitate the following processes, we rotate the color space of both the source and the target images such that the white axis is changed from $\overrightarrow{(1,1,1)}$ to $\overrightarrow{(0,0,1)}$. This is done by multiplying each color with the following rotational ma-

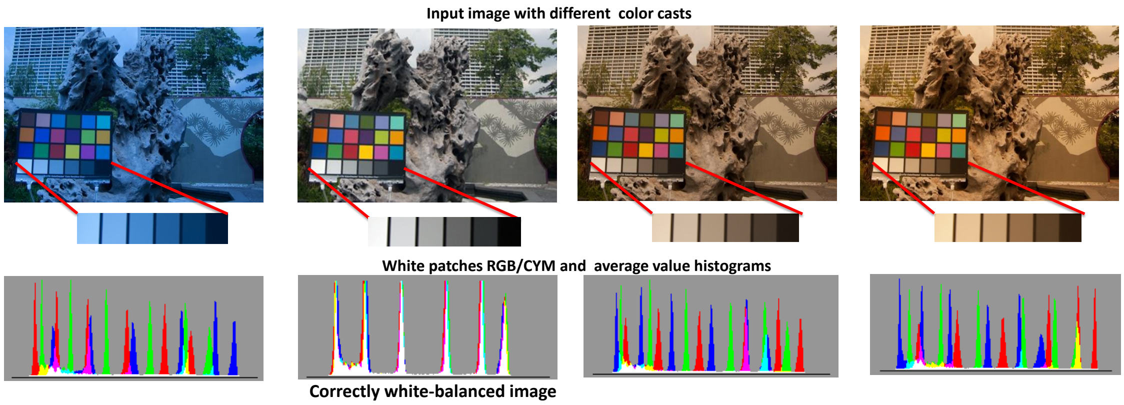

Figure 3: *This figure shows the importance of proper white-balance in determining the proper scene luminance. A scene was captured with a color chart and white balanced with different settings. The achromatic patches on the color chart are extracted and their color channel histograms as well as overall average is shown. We can see that for the correct white-balance setting, the white patches histograms converge for each patch given six coherent peaks.*

trix:

$$\mathbf{R} = \begin{vmatrix} \cos(\beta) & 0 & \sin(\beta) \\ 0 & 1 & 0 \\ -\sin(\beta) & 0 & \cos(\beta) \end{vmatrix} \begin{vmatrix} \cos(\alpha) & -\sin(\alpha) & 0 \\ \sin(\alpha) & \cos(\alpha) & 0 \\ 0 & 0 & 1 \end{vmatrix}, \tag{1}$$

where α is the angle of the rotation around $\overrightarrow{(0,0,1)}$ axis to align $\overrightarrow{(1,1,1)}$ on the plane created by $\overrightarrow{(1,0,0)}$ axis and $\overrightarrow{(0,1,0)}$ axis; and β is the angle of the rotation around $\overrightarrow{(0,1,0)}$ axis to align $\overrightarrow{(1,1,1)}$ to $\overrightarrow{(0,0,1)}$.

After the rotation, the $\overrightarrow{(0,0,1)}$ axis represents the luminance (achromatic) channel and the other two axes represent chromaticity.

3.2. Matching Luminance Channel

The next step is to match the overall brightness between the two images. We use the transformed luminance values for this step and adopt Xiao et al.'s gradient preserving matching technique [XM09]. After this step, the output image will have a similar brightness histogram with the target image while preserving the gradient of the source image.

In this procedure, histogram matching is used to convert the source luminance L_s into the intermediate luminance L_f, which has exactly the same histogram as the target luminance L_t (in Eq. 2).

$$L_f = \mathcal{C}_t^{-1}\left(\mathcal{C}_s(L_s)\right), \tag{2}$$

where $\mathcal{C}_s$ and $\mathcal{C}_t$ are the cumulative histogram of L_s and L_t respectively.

Next, the output luminance L_o is obtained by solving the following linear equation.

$$\left[\mathbf{I}+\lambda(\mathbf{D_x}^\top\mathbf{D_x}+\mathbf{D_y}^\top\mathbf{D_y})\right] L_o = L_f+\lambda(\mathbf{D_x}^\top\mathbf{D_x}+\mathbf{D_y}^\top\mathbf{D_y})L_s, \tag{3}$$

where $\mathbf{I}$ is the identity matrix; $\mathbf{D_x}$, $\mathbf{D_y}$ are two gradient matrices along to x, and y direction; λ is a regularization parameter. As shown by [XM09], the gradient regularization term helps to reduce halo artifacts that may arise in the histogram mapping. In our experiments, we set λ to be 1.

3.3. Aligning the Color Gamut

To align the source color gamuts to the target resulting from the previous step, the centers of the source and the target image gamuts are estimated based on the mean values μ_s and μ_t of the source and target images. Note that these mean values are from the images which are already white-balanced. The color gamuts are shifted so that the center of the gamuts are located at the origin as follows:

$$\begin{aligned} I_s &= I_s - \mu_s, \\ I_t &= I_t - \mu_t. \end{aligned} \tag{4}$$

The gamut mapping process is approximated by a linear transformation $\mathbf{T}$ that includes a scale and a rotation (defined in Eq. 5), which is used to align the gamut of the source image to that of the target image (Fig. 4).

$$\mathbf{T} = \begin{vmatrix} s_1\cos(\theta) & -s_1\sin(\theta) & 0 \\ s_2\sin(\theta) & s_2\cos(\theta) & 0 \\ 0 & 0 & 1 \end{vmatrix}, \tag{5}$$

where s_1, s_2 are two scale values for the two chromatic axes and θ is an angle for the rotation around the luminance axis.

To compute the parameters for the transformation matrix

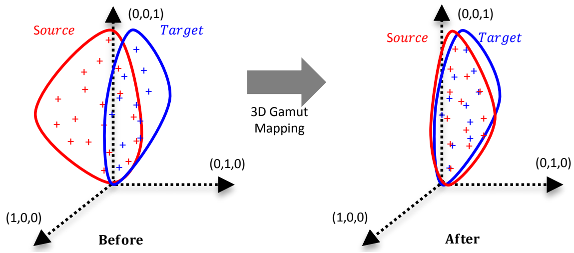

Figure 4: *Our gamut mapping step to align the color distributions between two images.*

T, we minimize for the following cost function:

$$f(\mathbf{T}) = 2\mathcal{V}((\mathbf{T} \times CH_s) \oplus CH_t) - \mathcal{V}(CH_t) - \mathcal{V}(\mathbf{T} \times CH_s), \quad (6)$$

where CH_s, and CH_t are the full 3D convex hulls of the source and target image respectively. The operator $\oplus$ is the point concatenation operation between two convex hulls and the operator $\mathcal{V}(\cdot)$ is the volume of the convex hull. A volume of a combination of two convex hulls is always larger or equal to that of individual convex hull. The idea behind this optimization is to use the volume of the convex hull for the optimization, that is, to make the gamut of the output image to be inside of the target image's gamut and enlarge the gamut of the output image as much as possible. Since Eq. 6 is a non-convex function, a brute-force search is required to find the global optimum. We obtain an approximate solution using the the quasi-Newton method found in Matlab's `fminunc` optimization toolbox.

After the transformation matrix **T** is computed, the output image is obtained by transforming the source image by **T** and shifting it back to the original center of the target gamut as follows:

$$I_o = \mathbf{T} I_s + \mu_t. \quad (7)$$

3.4. Undoing White-Balance

To compute the final color transferred image, the resulting image I_o from the previous step is rotated back so that the luminance axis $\overrightarrow{(0,0,1)}$ is mapped to the original white point vector $\overrightarrow{(1,1,1)}$. This is done by multiplying with the inverse of the rotation matrix **R** defined in Eq. 1. The final step is to undo the white balancing by multiplying the colors with the previously computed white point of the target image (r_{wt}, g_{wt}, b_{wt}).

4. Experiments

4.1. Evaluation Metric

Ideally, the goal of color transfer is to obtain an output image sharing the gamut with the target image. To this end, we propose to evaluate the transform by measuring the distance between the gamut of the output image and the gamut of the target image as follows:

$$D(I_t, I_o) = (\mathcal{V}(CH_c) - \mathcal{V}(CH_t)) + (\mathcal{V}(CH_c) - \mathcal{V}(CH_o)), \quad (8)$$

where CH_t, and CH_o are the convex hull of the target and the output images respectively. The term CH_c is the convex hull of the combination of the target and the output, the operator $\mathcal{V}(\cdot)$ is the volume of the convex hull. As mentioned above, the volume of a combination of two convex hulls is always larger or equal to that of individual convex hull. Therefore, the value of Eq. 8 is non-negative. We acknowledge that this metric does not provide a fair comparison with existing methods since our approach explicitly minimizes an objective function based on this metric. However, the metric does reveal to the extent that other methods produce results that are out of gamut with the target image.

4.2. Results

We first evaluate the contribution of the gamut mapping and the white-balancing steps in our framework. To do this, we test our method with and without the white-balancing step. We also test Pitie et al's [PKD07] method by adding white-balancing as the first step. We use the source and target images in Fig. 1 for comparison. The results of all these approach are shown in Fig. 5. It can be clearly seen that our method with the white-balancing step gives a better look and feel than the case without the white-balancing. It is worth noting that both Petie et al's methods (with and without white-balancing step) has a great deal of out-of-gamut colors.

Next we compare our method to other global color transfer methods: [RAGS01], [PKD07], and [XM09]. The qualitative comparisons are shown in Fig. 1, 6, 7, and 8 (where Fig. 1 shows Example 1; Figs. 6, 7, and 8 show Examples 2-10).

It can be observed that most of the source and the target images have notable illumination differences. This can be seen by examining regions in the image that represent white surfaces or the illumination source (e.g. the sky). Such image pairs occur frequently in the color transfer literature. The source and color images are also selected such that the gamuts are different, either larger or smaller. For most of images, estimating white points can be done using the algorithm proposed in [GGVDW12]. However, in a few cases like Examples 1 and 7, this algorithm did not perform satisfactorily. Therefore, these images required the user to manually select the white points in the images. This is done by selecting regions in the image that represent white surfaces or illumination sources.

The target and the source image pairs in Example 4 and Example 10 have been swapped to show the effect of reversing the color transfer direction. As can be seen in the examples, the results obtained using our method have less out-of-

Figure 5: *This figure shows the contribution of the gamut mapping and white-balancing (WB) in our framework. It is clear seen that the gamut mapping step help our method reduce out-of-gamut colors in comparison with the results from Petie et al. [PKD07]. While the white-balancing step make the color cast of the output image closer to that of the target image.*

gamut color than the other methods. Moreover, our approach arguably produces output images with a closer look and feel to target images than the other methods.

We also provide quantitative evaluation using the metric developed in Eq. 8. Table 1 shows the quantitative comparisons between all methods and our method also performs the best in the quantitative measure.

All methods are implemented in MATLAB v.8.0 on a dual core 3.10 GHz PC with 16.0 GB RAM. The comparison of execution time between all methods is shown in Table 2. Note that the timing performance of our method is taken as the baseline for the comparisons.

Our method, like other global color transfer methods, is not without limitation. It can fail when the source and target images have complicated color distributions like in Fig. 9. In this example, the goal is to make the foliage in the source image to become greener and remove the color cast caused by the sun. This can not be handled by a linear matrix. As a result, the color cast in the sky region can not be removed, therefore the output image still does not have the same look and feel as the target image (see Fig. 9). Performing color transfer in a local manner may solve this case.

5. Discussion and Summary

This paper has presented a new approach for color transfer. Our method first removes the color cast in the source

Example	Ours	[RAGS01]	[PKD07]	[XM09]
#1	**0.0424**	0.2533	0.2679	0.3363
#2	**0.1630**	0.2856	0.2114	0.3032
#3	**0.1510**	0.2080	0.2151	0.2656
#4	**0.0599**	0.2556	0.0646	0.1434
#5	0.1907	0.2485	0.1915	**0.1776**
#6	**0.0799**	0.1112	0.1458	0.1271
#7	**0.0385**	0.0745	0.0839	0.1229
#8	**0.0692**	0.1492	0.0751	0.0773
#9	**0.0111**	0.0165	0.0665	0.1315
#10	**0.0371**	0.1078	0.0863	0.1526

Table 1: *This table shows the comparisons between all methods in terms of the difference between target and output gamut from Eq. 8. The images for these examples are shown in Figs. 1, 6, 7, and 8.*

Method	Relative performance
Ours	1.000
[RAGS01]	0.047
[PKD07]	2.193
[XM09]	1.823

Table 2: *This table shows the comparisons between all methods in terms of timing performance. Timing performance of our method is taken as the baseline for comparing with other methods.*

Figure 6: *This figure shows Examples 2, 3, and 4 for comparisons between all methods.*

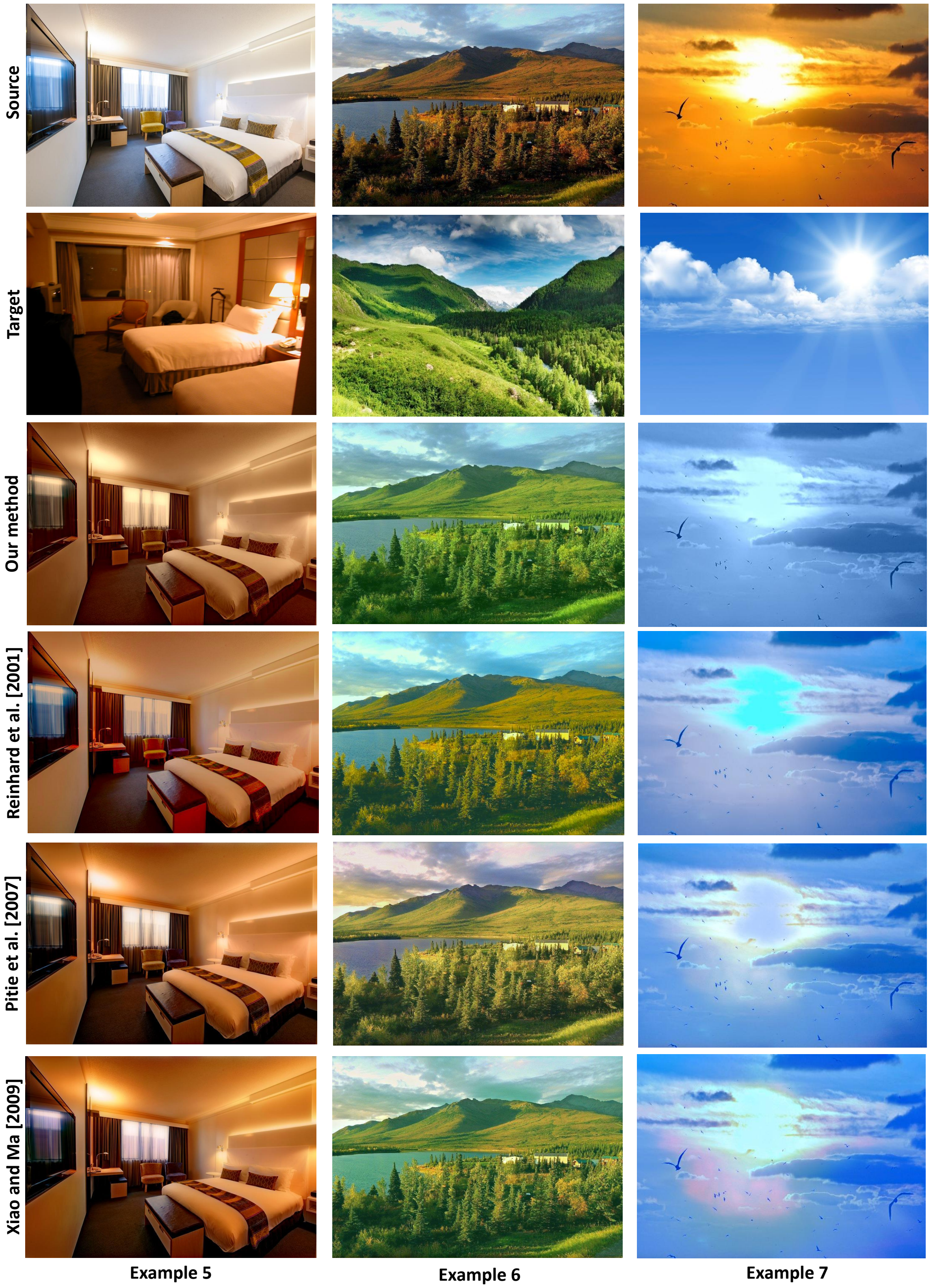

Figure 7: *This figure shows Examples 5, 6, and 7 for comparisons between all methods.*

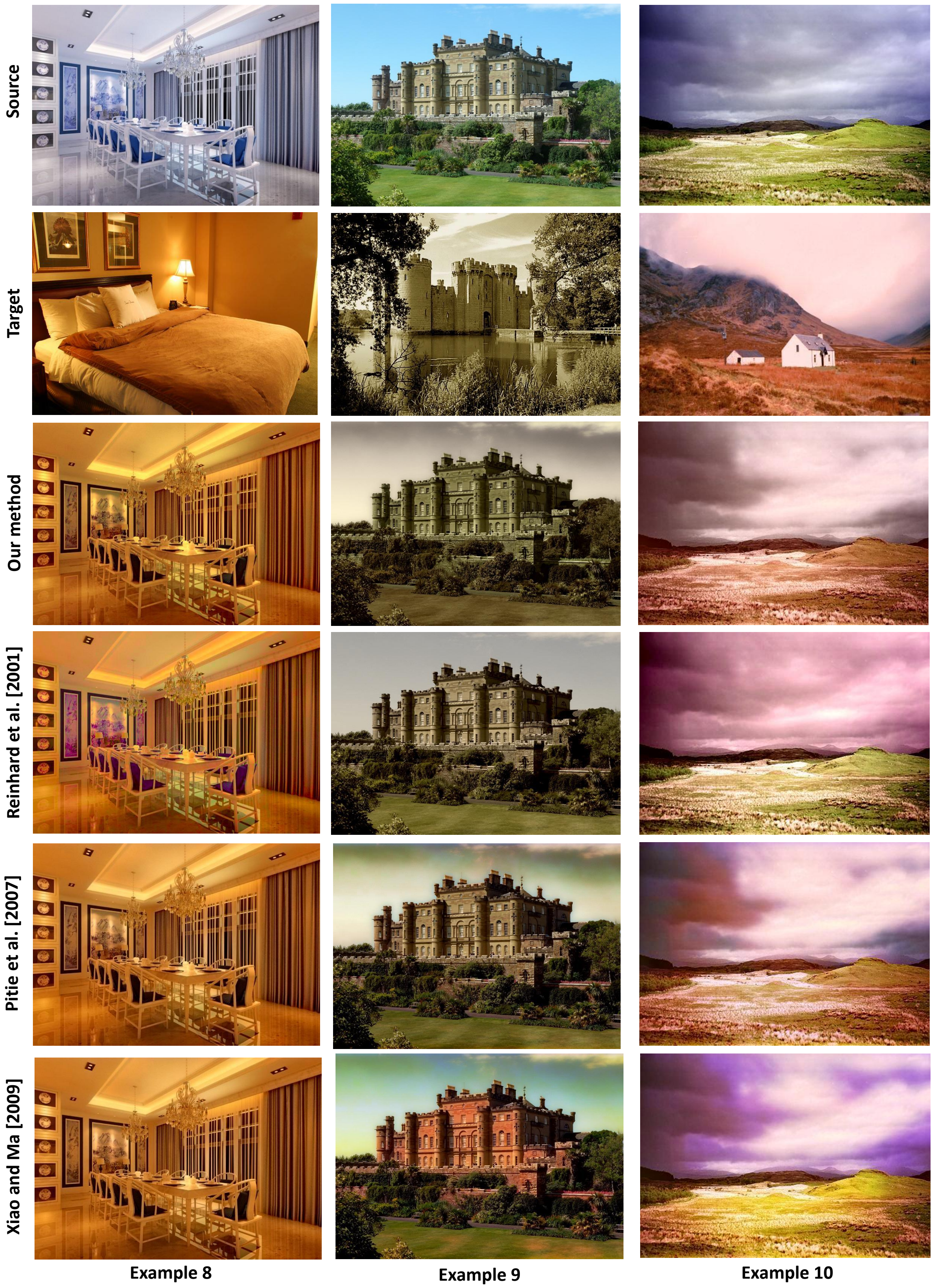

Figure 8: *This figure shows Examples 8, 9, and 10 for comparisons between all methods.*

Figure 9: *This figure shows an failed case of our method. In this example, the goal is to make the foliage in the source image become greener and remove the color cast caused by the sun. This can not be handled by a linear matrix. As a result, the color cast in the sky region can not be removed, and the output image still does not have the same look and feel as the target image.*

and target image through white-balancing. This step allows us to align the scene content along its white-axis to facilitate luminance processing. It also allows our gamut-mapping technique to only manipulate the chromatic axes. It is worth noting that this step does relies on the white-balancing algorithm's success in finding the correct white-point in the scene. In the event that this fails, the user can easily manually select a white-point in the scene. In some cases, the images have already been white-balanced. In such cases, the white-point has already lain along the (0,0,0) to (1,1,1) line in the sRGB color space and will not affect our subsequent steps. We have also presented a simple metric to determine how much overlap there is between the color transferred image and the target image. This gives us a way to quantify our results. Our experiments show that our illuminant aware and gamut constrained method produces images that are both subjectively and quantitatively better than many of the previous methods. In the future, we plan to extend our method for the local color transfer as well as to videos.

Acknowledgement

This work was funded by A*STAR Science and Engineering Research Council, Public Sector Research Funding Grant no. 1121202020 and ICT R&D program of MSIP/IITP 14-824-09-006.

References

[AP10] AN X., PELLACINI F.: User-controllable color transfer. *Computer Graphics Forum (Proceedings of Eurographics 2010) 29*, 2 (2010), 263–271. 2

[BMCF02] BARNARD K., MARTIN L., COATH A., FUNT B.: A comparison of computational color constancy algorithms. ii. experiments with image data. *TIP 11*, 9 (2002), 985–996. 3

[CGZ07] CHONG H. Y., GORTLER S. J., ZICKLER T.: The von kries hypothesis and a basis for color constancy. In *ICCV* (2007), pp. 1–8. 3

[CHZ12] CHAKRABARTI A., HIRAKAWA K., ZICKLER T.: Color constancy with spatio-spectral statistics. *TPAMI 34*, 8 (2012), 1509–1519. 3

[GGVDW11] GIJSENIJ A., GEVERS T., VAN DE WEIJER J.: Computational color constancy: Survey and experiments. *TIP 20*, 9 (2011), 2475–2489. 3

[GGVDW12] GIJSENIJ A., GEVERS T., VAN DE WEIJER J.: Improving color constancy by photometric edge weighting. *TPAMI 34*, 5 (2012), 918–929. 3, 5

[HLKK14] HWANG Y. B., LEE J. Y., KWON I. S., KIM S. J.: Color transfer using probabilistic moving least squares. In *CVPR* (2014), pp. 1–8. 2

[HSGL13] HACOHEN Y., SHECHTMAN E., GOLDMAN D. B., LISCHINSKI D.: Optimizing color consistency in photo collections. *ACM Transactions on Graphics (Proceedings of ACM SIGGRAPH 2013) 32*, 4 (2013), 85:1 – 85:9. 2

[KSK02] KRASILNIKOV N., SHELEPIN Y., KRASILNIKOVA O.: Mathematical model of the color constancy of the human visual system. *Journal of Optical Technology C/C of Opticheckii Zhurnal 69*, 5 (2002), 327–332. 2

[OSS11] OLIVEIRA M., SAPPA A. D., SANTOS V.: Unsupervised local color correction for coarsely registered images. In *CVPR* (2011), pp. 201–208. 2

[PKD05] PITIE F., KOKARAM A. C., DAHYOT R.: N-dimensional probability density function transfer and its application to color transfer. In *ICCV* (2005), pp. 1434–1439. 2

[PKD07] PITIÉ F., KOKARAM A. C., DAHYOT R.: Automated colour grading using colour distribution transfer. *Computer Vision and Image Understanding 107*, 1 (2007), 123–137. 2, 5, 6

[PR11] POULI T., REINHARD E.: Progressive color transfer for images of arbitrary dynamic range. *Computers & Graphics 35*, 1 (2011), 67–80. 2

[RAGS01] REINHARD E., ADHIKHMIN M., GOOCH B., SHIRLEY P.: Color transfer between images. *Computer Graphics and Applications 21*, 5 (Sep 2001), 34–41. 2, 5, 6

[TJT05] TAI Y.-W., JIA J., TANG C.-K.: Local color transfer via probabilistic segmentation by expectation-maximization. In *CVPR* (2005), pp. 747–754. 2

[XM06] XIAO X., MA L.: Color transfer in correlated color space. In *Proceedings of the 2006 ACM international conference on Virtual reality continuum and its applications* (2006), pp. 305–309. 2

[XM09] XIAO X., MA L.: Gradient-preserving color transfer. In *Computer Graphics Forum* (2009), Wiley Online Library, pp. 1879–1886. 2, 4, 5, 6