

**COLOR MAPPING FOR CAMERA-BASED COLOR
CALIBRATION AND COLOR TRANSFER**

NGUYEN HO MAN RANG

NATIONAL UNIVERSITY OF SINGAPORE

2016

**COLOR MAPPING FOR CAMERA-BASED COLOR
CALIBRATION AND COLOR TRANSFER**

NGUYEN HO MAN RANG

(B.E., Ho Chi Minh City University of Technology)

A THESIS SUBMITTED

**FOR THE DEGREE OF DOCTOR OF PHILOSOPHY
SCHOOL OF COMPUTING
NATIONAL UNIVERSITY OF SINGAPORE**


2016

© 2016, NGUYEN Ho Man Rang

Declaration

I hereby declare that this thesis is my original work and it has been written by me in its entirety. I have duly acknowledged all the sources of information which have been used in the thesis.

This thesis has also not been submitted for any degree in any university previously.

A handwritten signature in blue ink, consisting of a series of loops and curves, positioned above a horizontal line.

Nguyen Ho Man Rang
August 17, 2016

To my parents and wife

Acknowledgements

I would like to thank all the people who contributed in some way to the work presented in this thesis. First and foremost, I would like to express my sincere appreciation and gratitude to my advisor Prof. Michael S. Brown for his enthusiastic and patient guidance and extremely encouraging advice during my research. His guidance helped me in all the time of research and writing of this thesis.

Additionally, I would like to thank my committee Prof. Mohan Kankanhalli, Prof. Ng Teck Khim, and Prof. Yasuyuki Matsushita for their insightful suggestions and feedbacks from my thesis proposal. Their comments and advices were critical in making my thesis more accurate, solid and widen my research from various perspectives.

I would also like to thank my co-authors Dr. Dilip Prasad, Dr. Seon Joo Kim for their great contribution to my research work. Besides, I would like to thank Dr. Lin Haiting, Dr. Den Fanbo, Dr. Gao Junhong, Dr. Li Yu, Dr. Cheng Dongliang, Russell Looi, Mahsa Paknezhad, Abdelrahman Kamel, Hakki Can Karaimer, Hu Sixing and their members in vision lab, who as both labmates and friends, were always willing to help and gave their best suggestions. Our friendship has made my life as a graduate student very colorful and enjoyable.

Lastly, I would like to express my great gratitude to my family for their unflagging love and unconditional support throughout my life and my studies. I would like to thank my parents for their constant love and support. I would also like to thank my wife who is always by my side. This thesis would not have been possible if without her love, understanding and support.

Contents

Abstract	iv
List of Figures	vi
List of Tables	xi
1 Introduction	1
1.1 Motivation	2
1.2 Selective Literature Review	5
1.3 Objective	6
1.4 Contributions	7
1.5 Road Map	9
2 Background and Related Work	10
2.1 Background	10
2.1.1 Color perception	10
2.1.2 Color representation	13
2.1.3 Color spaces	14
2.1.4 Color image formation and camera pipeline	18
2.2 Related work	21
2.2.1 Color calibration between camera devices	21
2.2.2 RAW reconstruction from its corresponding sRGB image	23
2.2.3 Color transfer between a pair of images	25
2.2.4 Color constancy	27
2.3 Summary	28
3 RAW-to-RAW: Mapping between Image Sensor Color Responses	29
3.1 Introduction	30

3.2	Preliminaries	32
3.3	Evaluating mapping approaches	35
3.3.1	Mapping methods	35
3.3.2	Global versus illumination-specific	36
3.3.3	Discussion	39
3.4	Proposed illumination-independent method	40
3.5	Experiments and results	42
3.5.1	Controlled image set	44
3.5.2	Outdoor image set	49
3.6	Example application	52
3.7	Discussion and Summary	52
4	Raw to Photo-finished sRGB Output Mapping	54
4.1	Introduction	55
4.2	Proposed Approach	57
4.2.1	In-Camera Imaging Model Estimation	58
4.2.2	Modified Octree Partitioning	62
4.2.3	Metadata Embedding	63
4.2.4	RAW Reconstruction	66
4.3	Experiments	66
4.4	Applications	70
4.4.1	White-Balance Correction	70
4.4.2	Image Deblurring	71
4.5	Discussion and Summary	71
5	Color Transfer between a Pair of Images	75
5.1	Introduction	77
5.2	Our approach	78
5.2.1	Matching white points	79
5.2.2	Matching brightness channel	82
5.2.3	Aligning the color gamut	83
5.2.4	Undoing white-balance	85
5.3	Experiments	85

5.3.1	Evaluation metric	85
5.3.2	Results	86
5.4	Discussion and Summary	92
6	Conclusion and Future Directions	94
6.1	Overall Summary	94
6.2	Future directions	96
6.2.1	Harmonizing a group of images	96
6.2.2	Two-way reconstruction between RAW and sRGB	96
	Bibliography	98

Abstract

This thesis examines color mapping methods that aim to reduce color difference between images in three contexts. The first context is at the camera sensor level, where differences in spectral sensitivity functions of the sensors result in different RGB responses to the incoming light. This work attempts to produce an accurate color mapping between camera sensor-specific color spaces such that the imaged scenes appear the same. The second context targets the camera processing pipeline where in-camera photo-finishing operations have heavily processed the original RAW image to produce the final sRGB output. This work aims to find a mapping to undo the in-camera processing to obtain the original sensor-specific colors. The third context targets color mapping between images from unknown sources (e.g. from the internet, photo-sharing sites, etc). For these type of images, our work focuses on color transfer methods that attempts to manipulate a source image such that it shares a more similar “look and feel” of a specified target image.

This thesis begins by motivating the need for color calibration and color transfer between images. This is followed by a brief introduction on how color is represented and related work in the literature focused on both color calibration and color transfer. Afterwards, we describe three contributions made as part of this thesis work. In particular, we present a novel approach to estimate a mapping to an image of an arbitrary scene and illumination from one camera’s raw color space to another camera color space. This is achieved using an illumination-independent mapping approach that uses white-balancing to assist in reducing the number of required transformations. Our second contribution is a new method to encode the necessary metadata with a photo-finished sRGB

image for reconstructing its corresponding unprocessed RAW image. Our third contribution is a new approach for color transfer between two given images that is unique in its consideration of the scene illumination and the target image's color gamut. The thesis is concluded with summary of the existing contribution and potential future works.

List of Figures

1.1	This figure shows an example of RAW images of the same scene and illumination from different cameras. (a) and (b) show RAW images taken from Canon 1D, Nikon D40 respectively. (c) shows the numerical difference as root mean square error (RMSE) between (a) and (b). The color map shown on the right explains how much error each color denotes for (e.g. blue color denotes 0% error, while red color denotes up to 20% error). Note that RAW images shown in (a) and (b) are applied a gamma of 1/2.2 for better visualization purpose.	2
1.2	This figure shows an example of RAW and sRGB images. The bottom show their corresponding sizes.	3
1.3	This figure shows an example of color transfer problem.	4
2.1	Normalized spectral sensitivities of short (S), medium (M), and long (L) wavelength cones. <i>The image is reproduced from [Fairchild 2013].</i> . .	12
2.2	The figure shows the electromagnetic spectrum for different ranges and the close-up of the visible spectrum. Note that the visible spectrum is a rather narrow portion of the electromagnetic spectrum. <i>The image is reproduced from [Fairchild 2013].</i>	13
2.3	The diagram shows how scene spectral reflectances are converted to the CIE XYZ color space. CIE XYZ proposed three spectral response functions that map real world spectral power distributions (SDPs) to the X/Y/Z basis. The Y value in the CIE XYZ standard is mapped to the standard observer's luminosity function and is taken to represent the perceived brightness of the scene.	15

2.4	The sRGB and NTSC color spaces primaries and white-points as defined in the CIE XYZ color space. These establish the mapping between CIE XYZ and sRGB/NTSC and vice-versa.	16
2.5	The figures from left to right show the color sensitivity functions of three different cameras: Canon 1D Mark III, Nikon D40, and Sony Nex5N respectively.	19
2.6	This figure shows the pipeline to obtain sRGB image in consumer cameras. Note that the red circles denote for 'white' point while the coordinate systems represent the corresponding color space.	20
3.1	Top row shows three RAW images taken from Canon 1D, Nikon D40, and Sony α 57. Bottom row shows the numerical difference as root mean square error (RMSE) between the RAW images. The color map shown on the right explains how much error each color denotes for (e.g. blue color denotes 0% error, while red color denotes up to 20% error).	31
3.2	This figure shows the RAW-to-RAW calibration setup. Images of color calibration charts are taken under several different lighting conditions by the source and target cameras. Mapping between these two cameras' RAW-RGB colorspace can be estimated using a global mapping (all illuminations combined) or via multiple illuminant-specific mappings.	34
3.3	This figure shows the overview of our RAW-to-RAW calibration and conversion approach. (A) shows the steps of our calibration procedure. A global mapping f^G is computed using all of the color chart points. White-balancing is then applied to the color charts values from both cameras. Next a mapping on the canonical-illumination, f^{L_c} is computed. (B) illustrates the conversion procedure.	42
3.4	Example images of the controlled image set of paint chips and paper samples. The cyan rectangles are regions used for extracting the RAW values.	44

3.5	Comparison between all approaches. This figure shows the results on a Canon 1D and Nikon D40. Two lighting conditions: fluorescent and tungsten are shown with the camera setting given to the DNG software. Results show the mean RAW pixel errors (normalized) and the errors at the 25%, 50% (median) and 75% quartiles (Q1, Q2, Q3).	45
3.6	Comparison between all approaches. This figure shows the results on a Nikon D40 and a Sony α 57. Two lighting conditions: fluorescent and tungsten are shown with the camera setting given to the DNG software. Results show the mean RAW pixel errors (normalized) and the errors at the 25%, 50% (median) and 75% quartiles (Q1, Q2, Q3).	46
3.7	Comparison between all approaches. This figure shows the results on a Olympus E-PL6 and a Panasonic GX1. Two lighting conditions: fluorescent and tungsten are shown with the camera setting given to the DNG software. Results show the mean raw pixel errors (normalized) and the errors at the 25%, 50% (median) and 75% quartiles (Q1, Q2, Q3).	47
3.8	Comparison between all approaches. This figure shows the results on a Canon 600D and a Nikon D5200. Two lighting conditions: fluorescent and tungsten are shown with the camera setting given to the DNG software. Results show the mean RAW pixel errors (normalized) and the errors at the 25%, 50% (median) and 75% quartiles (Q1, Q2, Q3).	48
3.9	The figure shows example images of outdoor image set.	50
3.10	This figure shows an example of image mosaicing application. Three different cameras Nikon D40, Sony α 57, and Canon 1D are used. This figure shows the comparison before and after conversion. All the images are converted to the RAW-RGB space of the Sony α 57. These mosaics have been adjusted by a gamma for better visualization. . . .	53

4.1	(a) A 5616×3744 resolution high-quality sRGB-JPEG with our meta-data embedded Original JPEG size (9,788 KB); new size (9,852 KB). (b) Original RAW image is 25,947 KB. (c) Our reconstructed RAW image using the data in the self-contained JPEG. (d) Error map between (b) and (c). Overall reconstruction error is 0.2%.	55
4.2	This figure shows an overview of our approach. The section of the detailing each component is shown.	58
4.3	This figure shows an example of with/without using saturation threshold for estimating an inverse tone-curve f^{-1}	60
4.4	This figure shows an example of partition color space using uniform and octree approaches. The same number of bins $64 = 4^3$ is used for both two approaches.	61
4.5	This figure shows an example of our encoding method which avoids null characters.	66
4.6	This figure shows comparisons between our approach and our implementation of the upsampling approach proposed by Yuan and Sun for various scenes and cameras (a Canon 1Ds Mark III, a Canon 600D, a Nikon D5200, and a Sony $\alpha 57$). The white points on the difference maps indicate overexposed pixels with a value of 255 in any of the channels. The RMSEs for the each method are shown in the bottom right of each error map.	68
4.7	This figure shows an example of using different qualities of sRGB-JPEG images for reconstructing the RAW-RGB image. Here, three different qualities: fine, normal, and basic (which supports in Nikon cameras) are examined. The RMSEs for the each quality are shown in the bottom right of each error map.	69

4.8	This figure shows examples on correcting white-balance for different cameras: a Canon 1Ds Mark III, a Canon 600D, a Nikon D5200, a Nikon D7000, a Sony α 200 and a Sony α 57. The first column is the input images captured under the wrong white-balance settings; the second column shows the ground truth images captured under the proper settings. The third column displays the results applied the white-balance correction on our reconstructed RAW images. The final column shows the results applied the white-balance correction directly on the sRGB-JPEG images.	72
4.9	This figure shows examples for image deblurring for different cameras: a Canon 1Ds Mark III, a Nikon D7000, and a Sony α 200. A motion blur on the non-blurred ground truth RAW images is performed. The blurred sRGB image is synthesized using the parameterized color pipeline model. We applied our method to reconstruct the blurred RAW image, then deblurred it, and converted it back to the sRGB image. The first, third and fifth rows show the results, while the second, fourth and sixth rows show close-ups of particular regions. The signal-to-noise ratios (SNRs) were reported at the bottom right of each image.	73
5.1	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.	76
5.2	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).	79

5.3	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.	81
5.4	Our gamut mapping step to align the color distributions between two images.	84
5.5	This figure shows the contribution of the gamut mapping and white-balancing 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.'s method. While the white-balancing step make the color cast of the output image close to that of the target image.	86
5.6	This figure shows Examples 2, 3, and 4 for comparisons between all methods.	89
5.7	This figure shows Examples 5, 6, and 7 for comparisons between all methods.	90
5.8	This figure shows Examples 8, 9, and 10 for comparisons between all methods.	91
5.9	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.	92
6.1	The figure shows an example of a group of input images for designing a brochure.	97

List of Tables

3.1	The table shows the comparisons of error in terms of RMSE between all linear and non-linear models in three categories: global, specific and white-balancing. We used color calibration charts taken under four lighting conditions: Fluorescent, Incandescent, Halogen, and LED. Average means the average error for all the lightings. The source and target cameras shown here are for a Canon 1D and Nikon D40.	37
3.2	The table shows the comparisons of error in terms of RMSE between all linear and non-linear models in three categories: global, specific and white-balancing. We used color calibration charts taken under four lighting conditions: Fluorescent, Incandescent, Halogen, and LED. Average means the average error for all the lightings. The source and target cameras shown here are for an Olympus E-PL6 and a Panasonic GX1.	38
3.3	The table shows the comparisons of percentage error (in %) between white points (W) and color points (C) by the global transform.	40
3.4	The table shows the comparisons of histogram distance computed by the equation 3.5 between all the approaches from three cameras: Canon 1D, Nikon D40, and Sony α 57. For each pair of the cameras, four results are reported namely Before (B), Adobe (A), Hong et al. (H), and Ours (O).	50

3.5	The table shows the comparisons of histogram distance computed by the equation 3.5 between all the approaches from three cameras: Olympus E-PL6, Panasonic GX1, and Samsung NX2000. For each pair of the cameras, four results are reported namely Before (B), Adobe (A), Hong et al. (H), and Ours (O).	50
3.6	The table shows the comparisons of histogram distance between all the approaches from three cameras: Canon 600D, Nikon D5200, and Olympus E-PL6. For each pair of the cameras, four results are reported namely Before (B), Adobe (A), Hong et al. (H), and Ours (O).	51
3.7	The table shows the comparisons of histogram distance between all the approaches from three cameras: Canon 600D, Sony α 57, and Panasonic GX1. For each pair of the cameras, four results are reported namely Before (B), Adobe (A), Hong et al. (H), and Ours (O).	51
4.1	This table shows the amount of data allocated to model the camera-pipeline parameters into the metadata of a JPEG image. The g^{-1} allows up to 4728 control points pairs consisting of an sRGB and RAW-rgb color point (i.e. 6 values in total).	59
4.2	This table shows the three different strategies to select the scattered points for modeling the gamut mapping. These are uniform partition, k-means clustering, and our octree partitioning.	65
4.3	This table shows the comparison between our method and up-sampling method proposed by Yuan and Sun in terms of RMSE. For up-sampling method proposed by Yuan and Sun, RAW images at resolutions of 1/2 of the original size and 100×90 are used for upsampling.	69
5.1	The table shows the comparisons between all methods in terms of the difference between target and output gamut. The images for these examples are shown in Figs. 5.1, 5.6, 5.7, and 5.8.	88
5.2	The 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.	92

Chapter 1

Introduction

This thesis addresses the problem of color mapping for camera color space calibration and color transfer between images. These two terms are distinguished from one another based on the type of inputs given to the two respective algorithms and the assumptions pertaining to the inputs. In the case of color calibration we assume that there is a priori knowledge regarding the image formation, often specific to a particular camera. The goal is to archive an accurate mapping between the two specific color spaces. Once estimated, the color mapping can be applied to any subsequent images under one color space to transform to the other color space. This type of color mapping is similar in nature to colorimetric calibration of imaging devices. On the other hand, the term “color transfer” is used to distinguish algorithms that have no prior knowledge of the underlying image formation model. In these cases, the input is general a pair of images, a source and target image, where we desire to make the source image have a similar “look and feel” to the target image. This is a much more general problem than that of camera color calibration, and is intended more for visual compatibility versus accuracy.



Figure 1.1: This figure shows an example of RAW images of the same scene and illumination from different cameras. (a) and (b) show RAW images taken from Canon 1D, Nikon D40 respectively. (c) shows the numerical difference as root mean square error (**RMSE**) between (a) and (b). The color map shown on the right explains how much error each color denotes for (e.g. blue color denotes 0% error, while red color denotes up to 20% error). Note that RAW images shown in (a) and (b) are applied a gamma of $1/2.2$ for better visualization purpose.

1.1 Motivation

The color of an image is often attributed to the reflectance properties of the objects within the image, however, there are a number of additional factors often overlooked that also contribute to the image color. These include scene illumination, the camera sensor's sensitivity to the incoming light, and photo-finishing operations performed onboard a camera. These factors often cause problems when designing a robust computer vision algorithm intended to work effectively on a variety of camera models as well as illuminations. Therefore, for some computer vision tasks, such in-camera processing operations must be undone to map processed RGB values back to physically meaningful values (e.g. see [Chakrabarti et al. 2009; Debevec and Malik 1997; Diaz and Sturm 2011; Kim et al. 2012; Xiong et al. 2012]). Fortunately, most consumer cameras now allow images to be saved in *RAW* format that represents a minimally processed image obtained from the camera's sensor. This format is desirable for computer vision tasks as the RAW-RGB values

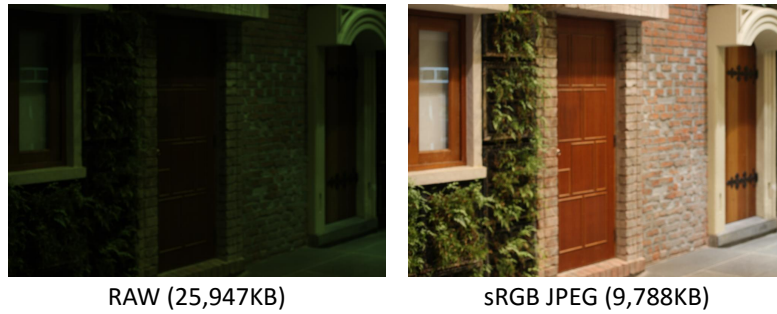


Figure 1.2: This figure shows an example of RAW and sRGB images. The bottom show their corresponding sizes.

are known to be linearly related to scene radiance [Mitsunaga and Nayar 1999; Lin et al. 2004; Pal et al. 2004; Lin and Zhang 2005; Chakrabarti et al. 2009; Kim et al. 2012], thereby avoiding the need to undo photo-finishing. One drawback, however, is that manufacturers have yet to agree on a standard RAW format. As a result, the RAW-RGB values are device specific and RAW images of the same scene and illumination from different cameras can differ significantly (as shown in Figure 1.1). Therefore, calibrating cameras' RAW-RGB color spaces to a standard color space still plays an important part for many computer vision tasks.

The second problem is that although RAW has many advantages over sRGB, including linear response to scene radiance, wider color gamut, and higher dynamic range (generally 12 – 14 bits), RAW images need significantly more storage space than their corresponding sRGB images. In addition, the vast majority of existing image-based applications are designed to work with 8-bit sRGB images, typically saved in JPEG format. Images saved in RAW must undergo some intermediate processes to convert them into sRGB. Figure 1.2 shows an example of RAW and sRGB images. Therefore, providing a fully self-contained JPEG image that allows RAW image reconstruction when needed is useful for many existing computer



Figure 1.3: This figure shows an example of color transfer problem.

graphics and vision tasks.

The third problem arises when we have a given collection of images that have already been processed from both in-camera processing and potentially color manipulation by image editing software. In these cases, the colors between these images can have significantly different “look and feel”, with different color casts and scene contrasts. It is often desirable to alter these images such that they share similar colors and contrast properties. One common way to do this is to choose an image as a reference (target) and alter another image’s (source) colors according to the color characteristics from the reference image. This procedure has been termed “color transfer” [Reinhard et al. 2001]. 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 reference image (as shown in Figure 1.3).

In the rest of this chapter a brief literature review on selective related work are discussed. This is followed by a discussion on the scope of the work in this thesis targeting color mapping and color transfer. The chapter concludes with the road map of the remainder of the thesis.

1.2 Selective Literature Review

For the past decades, many researchers have been working on color camera calibration and color transfer. Most color calibration works were to focus on transforming camera RGB output image to a standard color space [Kanamori et al. 1990; Hung 1993; Finlayson and Drew 1997; Hong et al. 2001; Martinez-Verdu et al. 2003; Funt and Bastani 2014]. These works focused on the related problem of making cameras colorimetric by finding a mapping between a camera’s RAW-RGB values and a color chart with known device-independent CIE XYZ values. They were mainly done by a simple 3×3 linear transform and are agnostic to information specific to the scene content (e.g. scene’s illumination). There were few prior works that address the mapping between camera-specific RAW-RGB spaces.

In case of reconstructing a camera-specific RAW image from the photo-finished sRGB output image, there have been a number of works on this topic [Chakrabarti et al. 2009; Kim et al. 2012; Chakrabarti et al. 2014]. However, these existing methods have two limitations. The first limitation is the need to calibrate the color processing models for a given camera. As discussed by [Kim et al. 2012], this involves computing multiple parameterized models for different camera settings (e.g. different picture styles). As a result, a single camera would have several different color mappings. Such calibration can be burdensome in practice. Second, the parameterized models are still saved as offline data and the appropriate model based on the camera settings needs to be determined when one desires to reverse an sRGB image.

Color transfer, on the other hand, is a well-studied topic in computer graphics with a number of existing methods (e.g., [Reinhard et al. 2001; Tai et al. 2005;

Pitié et al. 2007; Xiao and Ma 2009; Oliveira et al. 2011; HaCohen et al. 2011; Pouli and Reinhard 2011; Hwang et al. 2014]). These methods aim to modify an input image’s colors such that they are closer to a reference image colors. These methods work in either a global or local manner with some additional constraints (e.g., color distribution [Reinhard et al. 2001; Tai et al. 2005; Pitié et al. 2007], color gradient [Pitié et al. 2007; Xiao and Ma 2009], tone mapping [HaCohen et al. 2011]). However, these techniques do not prevent the color transformations from producing new colors in the transferred image that are not in the color gamut of the target image. The out-of-gamut colors can give a strange appearance to the target image which results in less color consistent between the images. Therefore, the main objectives of this thesis are to address these gaps on both color calibration and color transfer.

1.3 Objective

Our first target is to estimate a mapping that can convert a RAW image of an arbitrary scene and illumination from one camera’s RAW color space to another camera’s RAW color space. The goal here is to standardize the camera’s RAW-RGB spaces that is useful for a variety of reason, from comparing scene objects between different cameras to mosaicing RAW images from multiple cameras. This approach exploits the knowledge of how the image was formed in the camera-specific RAW-RGB color space. Like many other color calibration methods requiring the pixel correspondence, our approach also uses a standard color chart for calibration procedure.

Our second target is to compute a mapping between an sRGB and RAW image

pair and embed this information to the sRGB-JPEG image. The goal is to be able to reconstruct RAW image when needed using a self-contained sRGB-JPEG image. Unlike other radiometric calibration methods that require many pair of RAW and sRGB images under different settings, our approach requires only a pair of RAW and sRGB image for the calibration procedure.

Our third target is to investigate the problem of transferring colors of one image to the colors of another image. The goal is to make the colors consistent between images which is especially useful for creating an album or a video. Unlike color calibration requiring pixel correspondence, color transfer is more flexible and makes no assumptions about the image formation process. As a result, our approach can handle the case when the source and target image having significantly different scene content.

1.4 Contributions

This thesis makes three contributions to advance the state-of-the-art for color calibration and color transfer. These contributions are as follows:

- First, we present a novel approach for estimating a mapping that can convert a RAW image of an arbitrary scene and illumination from one camera’s RAW space to another. To this end, we examine various mapping strategies including linear and non-linear transformations applied both in a global and illumination-specific manner. We show that illumination-specific mappings give the best result, however, at the expense of requiring a large number of transformations. To address this issue, we introduce an illumination-independent mapping approach that uses white-balancing to assist in reduc-

ing the number of required transformations. We show that this approach achieves state-of-the-art results on a range of consumer cameras and images of arbitrary scenes and illuminations. This work has been published in the IEEE Conference on *Computer Vision and Pattern Recognition (CVPR)*, 2014 [Nguyen et al. 2014a].

- Second, we describe a state-of-the-art method to encode the necessary meta-data with the sRGB image for reconstructing a high-quality RAW image. As part of this procedure, we describe a fast breadth-first-search octree algorithm for finding the necessary control points to provide a mapping between the sRGB and RAW sensor color spaces that allows the number of octree cells to be controlled. In addition, we also describe a method to encode our data efficiently within the allowed 64 KB text comment field that is supported by the JPEG standard. This allows our method to be fully compatible with existing JPEG libraries and workflows. We compare our approach with existing methods and demonstrate the usefulness of the reconstructed RAW on two applications: white-balance correction and image-deblurring. This work has been published in the IEEE Conference on *Computer Vision and Pattern Recognition (CVPR)*, 2016 [Nguyen and Brown 2016].
- Thirdly, we propose a new approach for color transfer between two images. Our method is unique in its consideration of the scene illumination and the constraint of the color gamut of the output 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. Finally, 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. This work has been published in the *Journal of Computer Graphics Forum (CGF)*, 2014 [Nguyen et al. 2014c].

1.5 Road Map

The rest of this thesis is organized as follows. Chapter 2 provides a background on the fundamentals needed to understand color and various color mappings as well as work related to the topics in this thesis. Our approach for camera’s RAW-RGB spaces calibration is discussed in Chapter 3. Chapter 4 describes a novel method to embed the necessary information with sRGB-JPEG image for reconstructing its corresponding RAW image. Chapter 5 presents a new illuminant-aware approach for color transfer. Finally, Chapter 6 concludes the thesis with a short discussion on possible future research directions.

Chapter 2

Background and Related Work

This chapter provides a background in the fundamentals for color mapping targeting camera color calibration and color transfer. Section 2.1 begins with a brief overview of human color perception and descriptions of color representation and standard color spaces. This is followed by the discussion on how color is captured and processed on consumer digital cameras. Section 2.2 discusses related work targeting existing color mapping methods are provided.

2.1 Background

2.1.1 Color perception

Color perception depends on many different factors such as material properties of an object, the environment and the characteristics of the observer. In particular, color derives from scene's spectral power distribution interacting with spectral sensitivities in the retina of the human eyes. The color perception in human eyes and brain relates to the complicated physical and neural processes, some

of these haven't been fully understood. Therefore, this section will present a basic understanding of human visual perception with respects to color and color representation.

The human retina is organized as a grid of cells that are sensitive to light. These light-sensitive cells (photoreceptors) are divided into two classes: rods and cones. The ability of distinguishing colors of human eyes is due to the cone cells that are sometimes referred to as color receptors. There are three types of color receptors which are sensitive to different wavelengths of light. One type, reasonably separate from the other two, has the peak of wavelengths around 450 nm which is most sensitive to light perceived as blue; cones of this type are sometimes called short-wavelength cones, S cones, or blue cones. The other two types are closely related to each other, namely middle-wavelength and long-wavelength cones. The first are sometimes called M cones, or green cones with the peak of wavelengths around 540 nm which are most responsive to light perceived as green. While the second, L cones, or red cones with the peak of wavelengths around 570 nm, are most responsive to light perceived as greenish yellow. Figure 2.1 shows the normalized spectral sensitivities of these three type of cones.

The other type of light-sensitive cell in the eye, the rod, is more sensitive to the low level of illumination. In normal situations when light is bright enough to strongly stimulate the cones, the rods almost do not contribute to human vision. However, in dim light condition, there is not enough light energy to activate the cones, only the signal from the rods is perceived resulting in a colorless response. This also explains why objects that appear as colorless forms in moonlight although it is brightly colored in daylight.

Light or electromagnetic radiation is characterized by its wavelength and its

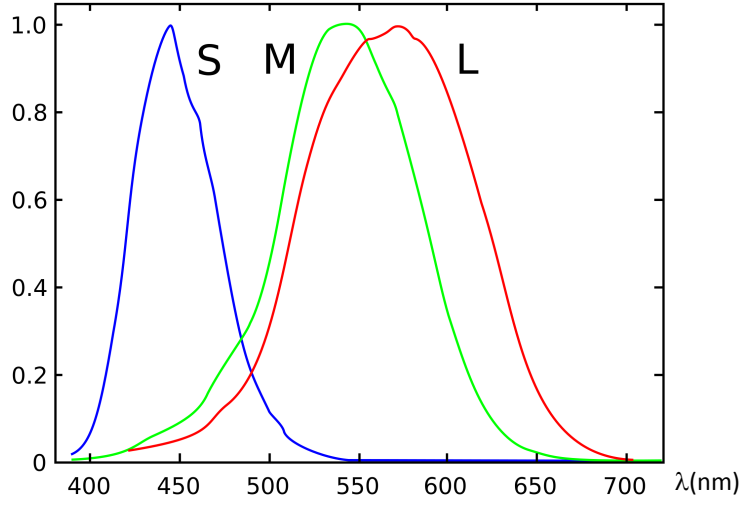


Figure 2.1: Normalized spectral sensitivities of short (S), medium (M), and long (L) wavelength cones. *The image is reproduced from [Fairchild 2013].*

intensity. When the wavelength is within the visible spectrum approximately from 400 nm to 700 nm (the range of wavelengths humans can perceive), it is known as “visible light”. Figure 2.2 shows the electromagnetic spectrum of the different ranges and the close-up of the visible spectrum. Note that the visible spectrum is a rather narrow portion of the electromagnetic spectrum. Visible light, no matter how many wavelengths it has, is reduced to three color components by the three types of cones when it comes into human eyes. In the retina, three types of cone cells response to incoming light corresponding to each location in the visual scene and result in three signals. These amounts of stimulation are sometimes called tristimulus values and can be formulated as follows:

$$C_i(x) = \int_{\omega} L(x, \lambda) S_i(\lambda) d\lambda, \quad (2.1)$$

where λ represents the wavelength, ω is the visible spectrum 400 – 700nm, $S_i(\lambda)$ is

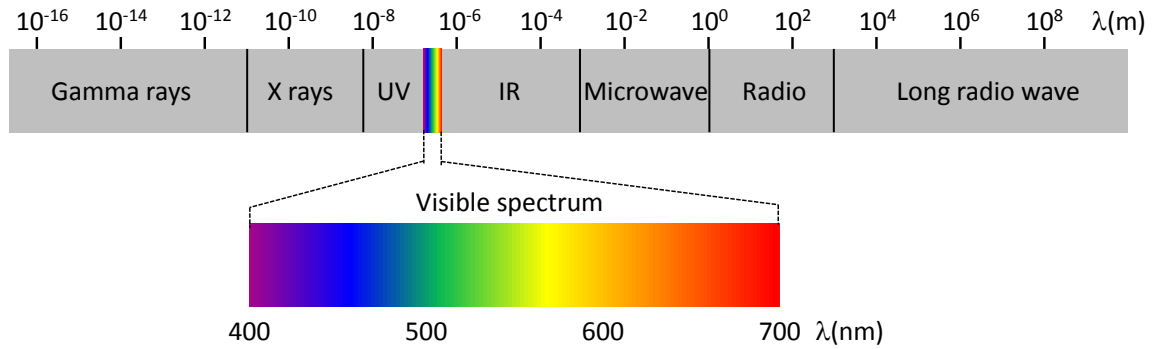


Figure 2.2: The figure shows the electromagnetic spectrum for different ranges and the close-up of the visible spectrum. Note that the visible spectrum is a rather narrow portion of the electromagnetic spectrum. *The image is reproduced from [Fairchild 2013].*

the sensitivity of the cone of the i -th type at wavelength λ ($i \in \{S, M, L\}$), and $L(x, \lambda)$ represents the light spectral power distribution of the location x on the scene.

2.1.2 Color representation

Humans commonly use color names to describe and distinguish colors from each other such as red, green, blue, orange, yellow, violet, and others. However, for science and industrial applications that work directly with color, a quantitative way is needed to quantify colors based on their relationship to human perception.

Thomas Young (1803) and Hermann von Helmholtz (1852) proposed a hypothesis about color vision. They suggested that color vision is based on three different photoreceptor types which are sensitive to a particular range of visible light. Their hypothesis was proved later when the human retina was discovered (as mentioned in Section 2.1.1). This hypothesis is also called the three-color or trichromatic theory. Based on the trichromatic theory, each color C can be synthesized from the

additive color mixture of three appropriate colors C_1 , C_2 , and C_3 as follows:

$$C \cong \alpha_1 C_1 + \alpha_2 C_2 + \alpha_3 C_3, \quad (2.2)$$

where the symbol \cong denotes visual equivalent, and α_1 , α_2 , and α_3 are corresponding coefficients. If the three colors C_1 , C_2 , and C_3 are chosen as the primaries, they will form a color space. There are many color spaces which serve for different purposes. However, they are all derived from CIE XYZ color space. More details about these color spaces are presented in the next section.

2.1.3 Color spaces

Virtually all modern color spaces used in image processing and computer vision trace their definition to the work by Guild and Wright [Guild 1932; Wright 1929] who performed experiments on human subjects to establish a standard RGB color space. Their findings were adopted in 1931 by the International Commission on Illumination (commonly referred to as the CIE from the French name *Commission Internationale de L'Éclairage*) to establish the CIE 1931 XYZ color space. Even though other color spaces were introduced later (and shown to be superior), the CIE 1931 XYZ remains the defacto color space for camera and video images.

CIE XYZ (dropping 1931 for brevity) established three hypothetical color primaries, X , Y , and Z . These primaries provides a mean to describe a spectral power distribution (SPD) by parameterizing it in terms of the X , Y , and Z . This means a three channel image I under the CIE XYZ color space can be described as:

$$I_c(\mathbf{x}) = \int_{\omega} C_c(\lambda) R(\mathbf{x}, \lambda) L(\lambda) d\lambda, \quad (2.3)$$

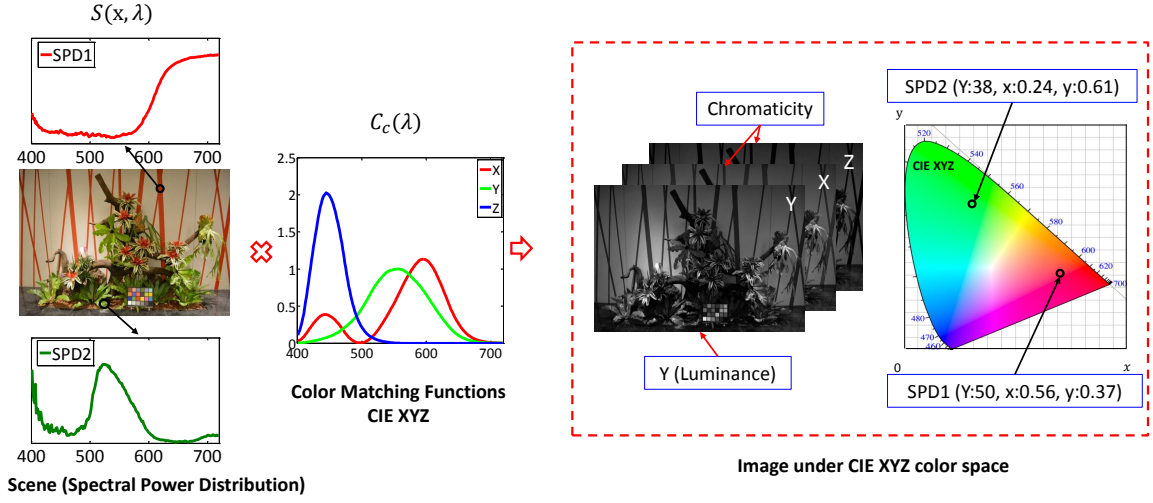


Figure 2.3: The diagram shows how scene spectral reflectances are converted to the CIE XYZ color space. CIE XYZ proposed three spectral response functions that map real world spectral power distributions (SDPs) to the X/Y/Z basis. The Y value in the CIE XYZ standard is mapped to the standard observer's luminosity function and is taken to represent the perceived brightness of the scene.

where λ represents the wavelength, ω is the visible spectrum 400 – 700nm, C_c is the CIE XYZ color matching function, and $c = X, Y, Z$ are the primaries. The term $R(x, \lambda)$ represents the scene's spectral reflectance at pixel x and $L(\lambda)$ is the spectral illumination in the scene. In many cases, the spectral reflectance and illumination at each pixel are combined together into the spectral power distribution $S(x, \lambda)$ (see in Figure 2.3). Therefore, Equation 2.3 can be rewritten as:

$$I_c(\mathbf{x}) = \int_{\omega} C_c(\lambda) S(\mathbf{x}, \lambda) d\lambda. \quad (2.4)$$

In this case, any $S(\mathbf{x})$ that maps to the same X/Y/Z values is considered to be perceived as the *same color* to an observer. The color space was defined such that the matching function associated with the Y primary has the same response as

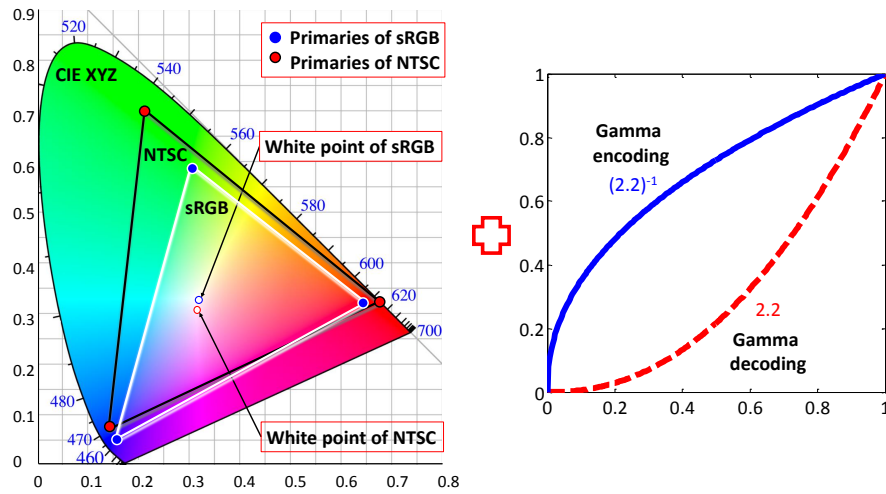


Figure 2.4: The sRGB and NTSC color spaces primaries and white-points as defined in the CIE XYZ color space. These establish the mapping between CIE XYZ and sRGB/NTSC and vice-versa.

the luminosity function of a standard human observer [Fairman et al. 1997]. This means that the Y value for a given spectral power distribution indicates how bright it is perceived with respect to other scene points. As such, Y is referred to as the “luminance of a scene” and is a desirable attribute of an imaged scene.

While CIE XYZ is useful for colorimetry to describe the relationships between SPDs, a color space based on RGB primaries related to real imaging and display hardware is desirable. To establish a new color space, two things are needed, namely the location of the three primaries (R, G, B) and the white-point in CIE XYZ. The white-point is used to determine what CIE XYZ color will represent white (or achromatic colors) in the color space. In particular, it is selected to match the viewing conditions of color space. For example, if it is assumed that a person will be observing a display in daylight, then the CIE XYZ value corresponding to daylight should be mapped to the new color space’s white value.

Figure 2.4 shows examples for the 1996 sRGB and 1987 National Television System Committee (NTSC) color spaces. Here, NTSC is used as an example. There are many other spaces as noted in [Süsstrunk et al. 1999], e.g. Adobe RGB, PAL, Apple RGB, and variations over the years, such as NTSC 1953, NTSC 1987, etc. Each color space has its own 3×3 linear transform based on its respective RGB primaries and white-point location within CIE XYZ.

For the sRGB primaries, the matrix to convert from sRGB to CIE XYZ is:

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} 0.4124 & 0.3576 & 0.1805 \\ 0.2126 & 0.7152 & 0.0722 \\ 0.0193 & 0.1192 & 0.9505 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix}. \quad (2.5)$$

The transform for NTSC (1987) back to CIE XYZ is:

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} 0.6071 & 0.1736 & 0.1995 \\ 0.2990 & 0.5870 & 0.1140 \\ 0.0000 & 0.0661 & 1.1115 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix}. \quad (2.6)$$

In both Eqs. 2.5 and 2.6, it is important to note that the R, G, B values need to be from images encoded in these respective color spaces. Such R, G, B values are often termed the “linear RGB” values, since both sRGB and NTSC use a final nonlinear gamma function as described in the following.

Gamma sRGB/NTSC were designed for display on CRT monitors and televisions. These devices did not have a linear response to voltage and an *encoding gamma* was applied to the three R/G/B channels as compensation as shown in Figure 2.4. For example, a red pixel would take the form $R' = R^{1/\gamma}$, where R is the linear RGB value and R' is the resulting gamma encoded value. This nonlinear gamma was embedded as the final step in the sRGB/NTSC definition. The gamma for NTSC

was set to $\gamma = 2.2$, the one for sRGB can be approximated by $\gamma = 2.2$ but is in fact slightly more complicated [Anderson et al. 1996]. Before sRGB or NTSC color spaces can be converted back to CIE XYZ, values must first be linearized using the inverse gamma.

2.1.4 Color image formation and camera pipeline

A digital camera is also a tristimulus system that simulates human visual system. A camera receives the visible light from the scene and reduces it into three response values: red, green, and blue. In specific, scene radiance (light spectra) first goes through the camera lens and then is filtered by the color filter array. Next it hits the cameras photosensors (CCD or CMOS), causing RAW sensor responses. The color filters filter light spectrum by wavelength range based on their spectral sensitivity functions (as shown in Figure 2.5). Color filters are necessary since the typical photosensors detect light intensity with little or no wavelength specificity, and they therefore cannot separate color information. Generally, these color filters, placed right above the photosensors, are composed from several different types of color filters (at least three different types) and arranged according to a particular pattern such as RGGB (Bayer) pattern, RGBE pattern, CYYM pattern, and others. Take the Bayer pattern on an imaging sensor for an example, each two-by-two submosaic on the pattern contains two green, one blue and one red filters, and each of them covers one pixel sensor. Therefore, Bayer filter pattern results in RGB tristimulus camera RAW responses. The physical formulation of RAW responses are similar to the tristimulus from human retina:

$$I(\mathbf{x}) = \int_{\omega} R_c(\lambda) S(\mathbf{x}, \lambda) L(\lambda) d\lambda, \quad (2.7)$$

where λ represents the wavelength, ω is the visible spectrum 400 – 700nm, R_c is the camera's spectral response, and c is the color channel $c = r, g, b$. The term $S(\mathbf{x}, \lambda)$ represents the scene's spectral response at pixel \mathbf{x} and $L(\lambda)$ is the lighting in the scene, assumed to be spatially uniform.

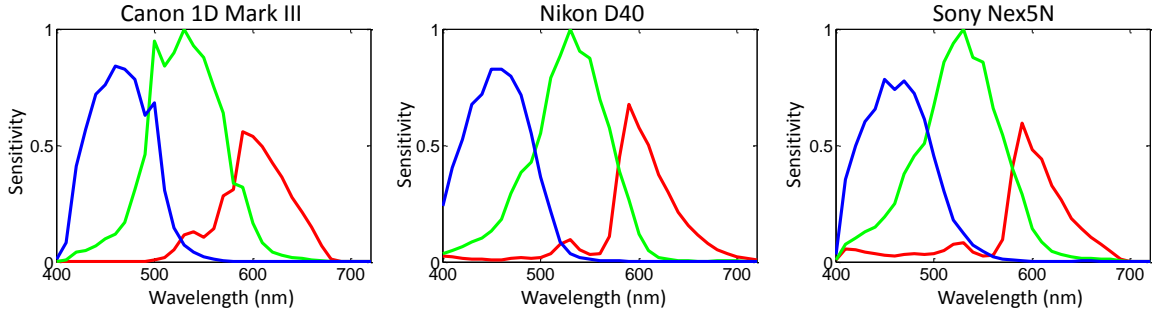


Figure 2.5: The figures from left to right show the color sensitivity functions of three different cameras: Canon 1D Mark III, Nikon D40, and Sony Nex5N respectively.

Figure 2.6 shows an overview of the common steps in a digital camera image pipeline. First, the RAW image is formed by response of scene's spectral on the camera sensitivities of a camera sensor. However, these values are not the same as CIE XYZ. This means that camera images are in their own camera-specific RAW-RGB color space which must be converted to sRGB. Before this happens, the image is generally white-balanced using a diagonal 3×3 matrix to remove illumination color casts and properly map the scene's white colors to lie along the achromatic line. After white-balancing, the image's RAW-RGB values are converted to CIE XYZ using a 3×3 color correction matrix (CCM). Once in the CIE XYZ color space, the image can be mapped to sRGB and the sRGB gamma is applied. However,

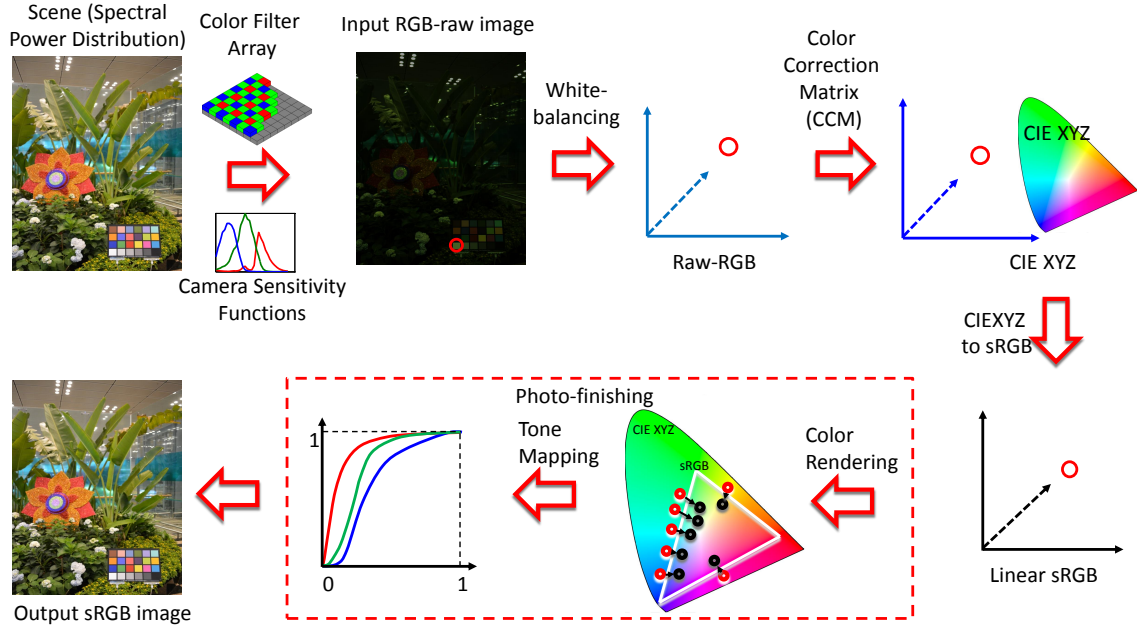


Figure 2.6: This figure shows the pipeline to obtain sRGB image in consumer cameras. Note that the red circles denote for ‘white’ point while the coordinate systems represent the corresponding color space.

most cameras apply their own tone-curve [Grossberg and Nayar 2003a; Kim and Pollefeys 2008; Lin et al. 2004; Lin and Zhang 2005] and/or additional selective color rendering [Chakrabarti et al. 2014; Kim et al. 2012; Lin et al. 2011; Xiong et al. 2012] as part of their proprietary photo-finishing.

Examining the pipeline, it can be clearly seen that there are four different factors that can affect the output colors: the scene’s content, illumination spectra, the spectral sensitivities of camera sensor, and the photo-finishing in camera. Among these factors, applying photo-finishing operations (e.g. tone-mapping, white-balancing, etc.) can change the image colors dramatically. As previously mentioned, for many computer vision tasks this in-camera processing must be undone to map sRGB values back to physically meaningful values (e.g. see [Debevec and Malik 1997; Diaz

and Sturm 2011; Kim et al. 2012; Xiong et al. 2012]).

2.2 Related work

This section will review existing color mapping methods. Based on their goals, we classify these methods into three main groups: color calibration between camera devices, RAW reconstruction from its corresponding sRGB image, and color transfer between a pair of images.

2.2.1 Color calibration between camera devices

Many applications in computer vision often require a system of several cameras. These cameras can be in different brands and brand models. Therefore, it is necessary to calibrate them to a standard color space. Color calibration process often relies on the pixel correspondences between each pair of images to compute mapping between camera devices. As such, a standard scene such as a color checker can easily extract the corresponding values from two devices is often used. Most of the existing methods choose CIE XYZ or its derivations (e.g. sRGB, CIE LAB and others) as a standard color space. These methods map the camera RGB outputs to this color space using different types of mappings: linear transform, polynomial transform or 3D look-up table.

Linear transform was used in works [Finlayson and Drew 1997; Martinez-Verdu et al. 2003; Funt et al. 2004; Funt and Bastani 2014]. Most of these method used some variation of least squares fitting to compute a simple 3×3 matrix transform. For example, [Finlayson and Drew 1997] proposed the white-point preserving least-squares (WPPLS) for determining the best least-squares transform that takes RGBs

to XYZs subject to the constraint that the RGB response induced by a white reflectance is mapped without error. On the other hand, [Funt et al. 2004] extended the standard least squares solution to a weighted version in which the weights are defined (in an intuitive sense) to be inversely proportional to the approximate size of the MacAdam color discrimination ellipse that would surround each XYZ. In a different manner, [Porikli 2003] built the joint-histogram (see also co-occurrence matrix) of the two images and computed the mapping by using dynamic programming based on the joint-histogram values.

Polynomial transform was proposed in [Hong et al. 2001] which extended the linear transformation by adding high-order terms (e.g., rg , rb , gb , r^2 , g^2 , b^2). The parameters can be obtained by solving a 3×11 matrix in a least-squares manner.

3D look-up table was suggested by [Kanamori et al. 1990; Hung 1993]. These methods often combines a table look up method, and a 3D color space interpolation method.

However, there is not a great deal of prior works addressing the mapping between RAW color spaces. This is mainly done by a simple 3×3 linear transform (e.g. [Martinez-Verdu et al. 2003]). Adobe has arguably been the most serious in addressing this issue. Adobe proposed a standard RAW space termed Digital Negative (DNG) in 2004, however, few manufacturers have adopted it. Adobe provides an SDK ¹ that can convert a wide range of camera models into Adobe's DNG format. Adobe has also provided a report describing the conversion [Adobe], which is explained in more detail in Chapter 3.

Other related works, which attempt to recover the full spectral response of the cameras (e.g. [Jiang et al. 2013; Urban et al. 2010; Prasad et al. 2013]) and spectral

¹<http://www.adobe.com/support/downloads/product.jsp?product=106&platform=Windows>

information of the scene (e.g. [Agahian et al. 2008; Abed et al. 2009; Nguyen et al. 2014b]), often contain complex procedures that may not be practical for mainstream use. Moreover, knowing the spectral responses does not directly reveal the best approach to use to map between different cameras.

2.2.2 RAW reconstruction from its corresponding sRGB image

Work related to RAW image reconstruction can be categorized into two areas: radiometric/camera color calibration and methods for image-upsampling.

Radiometric/Color Calibration are methods that aims to compute the necessary mappings to invert the non-linear transformations applied onboard cameras in order to have pixel values that are linear with respect to scene radiance. Conventional radiometric calibration algorithms used multiple images taken with controlled exposures in order to compute inverse response functions of the camera output intensity values to the incoming light. These methods targeted greyscale images [Lin and Zhang 2005], or computed a individual response functions per color channel [Debevec and Malik 2008; Grossberg and Nayar 2003b; Mann et al. 1995; Mitsunaga and Nayar 1999]. The main difference among these methods are the models used to represent the response function, e.g. exponentiation [Mann et al. 1995], polynomial [Mitsunaga and Nayar 1999], non-parametric [Debevec and Malik 2008], and PCA-based model [Grossberg and Nayar 2003b].

These early methods discarded RGB values that were too saturated, treating them as outliers to the radiometric model. Work in [Chakrabarti et al. 2009] and [Kim et al. 2012] found that these outliers were due to limitations in the radiometric models being used. To overcome this, Chakrabarti et al. [Chakrabarti et al. 2009]

proposed a method that used combinations of cross-channel linear transforms with per-channel multi-variate polynomials to model the camera color mapping process. Kim et al. [Kim et al. 2012] proposed a new in-camera imaging model that introduced an additional gamut mapping step for handling the out-of-gamut (i.e. saturated) colors. Later, Chakrabarti et al. [Chakrabarti et al. 2014] extended this idea and suggested using uncertainty modelling for handling the quantization of the sRGB colors. These methods significantly improved the ability to reverse sRGB images back to their RAW values, however, they do have two limitations with respect to the problem addressed in Chapter 4. The first limitation is the need to calibrate the color models for a given camera. As discussed by [Kim et al. 2012], this involves computing multiple parameterized models for different camera settings (e.g. different picture styles). As a result, a single camera would have several different color mappings. Such calibration can be burdensome in practice. Second, the parameterized models are still saved as offline data and the appropriate model based on the camera settings needs to be determined when one desires to reverse an sRGB image.

Image Upsampling methods that attempt to increase the resolution, or quality, of an image. Representative work include interpolation-based methods [Hou and Andrews 1978; Thévenaz et al. 2000], edge-based methods [Dai et al. 2007; Fattal 2007; Sun et al. 2008], and example-based methods [Freeman et al. 2000; Glasner et al. 2009]. These methods leverage a dictionary of image patches from high-quality images that are used to guide the upsampling process. The most similar to the problem addressed in Chapter 4 is the work by Yuan and Sun [Yuan and Sun 2011] who demonstrated a hybrid-image method that stored a small resolution version of the RAW image. In this work, the RAW values were upsampled to have

the same size as the sRGB image by using the sRGB image to guide the upsampling process. The RAW images used in this work were one half or one quarter size of the original RAW image. While these small-RAW are smaller than the original RAW image, they are still range approximately 1.5 – 6 MB in size. Also, like the work of [Kim et al. 2012] and [Chakrabarti et al. 2014], this approach requires additional data to be stored separately from the JPEG image in order to perform upsampling.

2.2.3 Color transfer between a pair of images

In this section, we will provide an overview and categorization of color transfer methods. One of the earliest known works in this area were proposed by Reinhard et al. [Reinhard et al. 2001]. They introduced a simple and efficient framework for color transfer color style between a pair of images. This problem still draws research attention (see the survey in [Faridul et al. 2014] for more detail). This problem can be approached in several different ways: feature correspondence, statistical properties, or user assistance. These methods are discussed below in more detail.

Feature Correspondence is often used for cases where source and target images shares a significantly large part of the same scene (e.g. frames in a video). In this method, a set of corresponding features are first extracted from the two input images and automatically matched together by applying one of the feature detection methods such as Scale Invariant Feature Transform (SIFT) [Lowe 2004] or Speeded Up Robust Features (SURF) [Bay et al. 2006]. This corresponding set can be then refined to reject outliers by applying the Random Sample Consensus (RANSAC) algorithm [Fischler and Bolles 1981]. Once the set of corresponding

features between the two images is determined, a color mapping between them can be computed from this set. For example, [Yamamoto et al. 2007] used SIFT to determine the set of corresponding features and subsequently applied a Gaussian convolution kernel to these feature locations to detect corresponding colors. The use of blur kernels (or other forms of windowing) improves robustness for instance against noise pollution. Corresponding colors are then used to build look-up tables that are subsequently applied to the target frames.

To obtain more improvement on the robustness, this approach can be extended to the set of corresponding regions in the images [Shao et al. 2007; Wang et al. 2010] rather than rely on only the piecewise feature correspondences. To this end, the image is first segmented, for instance with a mean-shift based image segmentation technique [Comaniciu and Meer 2002], followed by feature detection using an optical-flow based algorithm [Wang et al. 2010]. While [Tai et al. 2005] segmented the image through an EM framework and applied local color transfer to each of the segments.

Statistical Properties When feature correspondences between the input images are not available because the two images are captured under different scenes, statistical approaches are often more suitable to construct a mapping between the two input images. In this method, color pixels in an image are considered as a set of 3-dimensional points in 3D color space and its statistical properties such as the mean, standard deviation, histogram, or gamut is considered.

In most of the cases, images are stored using RGB color space where each channel is highly correlated to each other. As a result, changes in one channel can cause unpredictable effects. Therefore, many statistical-based methods, which perform the color transfer for each channel separately [Reinhard et al. 2001; Xiao

and Ma 2009; Pouli and Reinhard 2011], need to be first converted into a less correlated color space. In these methods, the selection of color space plays a crucial role [Reinhard and Pouli 2011]. In contrast, there are also several color transfer works which can work in 3D color space [Pitié et al. 2005; Xiao and Ma 2006; Pitié et al. 2007]. These methods are, therefore, less sensitive to the working color space.

User assistance In cases that the source image is grayscale or contains some objects having a restricted range of plausible colors (e.g. human faces), most of the automatic methods mentioned above will fail to find successful mapping. In this case, it requires user input to insert some constraints on some special region of the source image or guide the source and target correspondences. For example, [Cohen-Or et al. 2006; Wen et al. 2008; Pouli and Reinhard 2011] required the user to manually define layer mask or strokes for preserving regions from any color mapping. [An and Pellacini 2010] present an interactive tool where a user selects local regions to run the color transfer. On the other hand, manual user interaction may be used to define region correspondences between images [Welsh et al. 2002] or serving as initial colorized guidance through colored scribbles [Levin et al. 2004] and colored regions [Sauvaget et al. 2010].

2.2.4 Color constancy

Another area that is related to color mapping is the computational color constancy. The human visual system has an innate ability, termed color constancy, to perceive colors under different illumination in a constant manner. This ability of humans was discovered by [Daw 1967]. In 1971, it was experimented and presented by [Land and McCann 1971], who formulated retinex theory to explain

it. 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 the achromatic line in the color space.

In computer vision, color constancy is a crucial feature and many methods have been developed for this purpose. Most of these follow the retinex model which is based on a 3×3 diagonal matrix to correct the illumination. Although the real human color perception has been verified to be more complicated [Hurlbert and Wolf 2002], this model is still widely accepted in computer vision. It is currently a well-known topic and there are hundreds of researches published in this area [Buchsbaum 1980; Brainard and Wandell 1986; Zaidi et al. 1997; Finlayson and Trezzi 2004; Van De Weijer et al. 2007; Shi and Funt 2012]. A complete literature review on color constancy falls outside the scope of this thesis. Reader can refer to excellent surveys [Gijssen et al. 2011; Barnard et al. 2002] for more details.

2.3 Summary

The color mapping addressed in this thesis targets three different purposes. The first is to calibrate the colors of two cameras for further processing using two or more sample images. The second is to reconstruct RAW image from its sRGB. The final is to adjust the colors of two images for perceptual visual compatibility.

Chapter 3

RAW-to-RAW: Mapping between Image Sensor Color Responses

In this chapter, we will investigate color mapping in the low-level which is used for calibrating the camera's RAW-RGB spaces. Camera images saved in RAW format are being adopted in computer vision tasks since RAW pixel values represent minimally processed sensor responses. Camera manufacturers, however, have yet to adopt a standard for RAW images and current RAW-RGB values are device-dependent color spaces due to different sensors spectral sensitivities. This results in significantly different RAW images for the same scene captured with different cameras. This chapter focuses on estimating a mapping that can convert a RAW image of an arbitrary scene and illumination from one camera's raw space to another. To this end, we examine various mapping strategies including linear and non-linear transformations applied both in a global and illumination-specific manner. We show that illumination-specific mappings give the best result, however, at the expense of requiring a large number of transformations. To address

this issue, we introduce an illumination-independent mapping approach that uses white-balancing to assist in reducing the number of required transformations. We show that this approach achieves state-of-the-art results on a range of consumer cameras and images of arbitrary scenes and illuminations.

3.1 Introduction

Camera manufactures apply photo-finishing operations (e.g. tone-mapping, white-balancing, etc.) before saving images in the standard rgb format (i.e. JPEG). For many computer vision tasks, such in-camera processing must be undone to map sRGB values back to physically meaningful values (e.g. see [Chakrabarti et al. 2009; Debevec and Malik 1997; Diaz and Sturm 2011; Kim et al. 2012; Xiong et al. 2012]). Most consumer cameras now allow images to be saved in *RAW* format that represents a minimally processed image obtained from the camera’s sensor. This format is desirable for computer vision tasks as the RAW-RGB values are known to be linearly related to scene radiance [Chakrabarti et al. 2009; Kim et al. 2012], thereby avoiding the need to undo photo-finishing. One drawback, however, is that manufacturers have yet to agree on a standard raw format. As a result, the RAW-RGB values are device specific and RAW images of the same scene and illumination from different cameras can differ significantly (see Figure 3.1).

The goal of this chapter is to find a mapping between different cameras’ RAW-RGB colorspace. This is useful for a variety of reasons from comparing scene objects between different cameras to mosaicing RAW images from multiple cameras. This problem is inherently challenging since the RAW-RGB values represent a projection of scene radiance onto only three color channels, where the projection

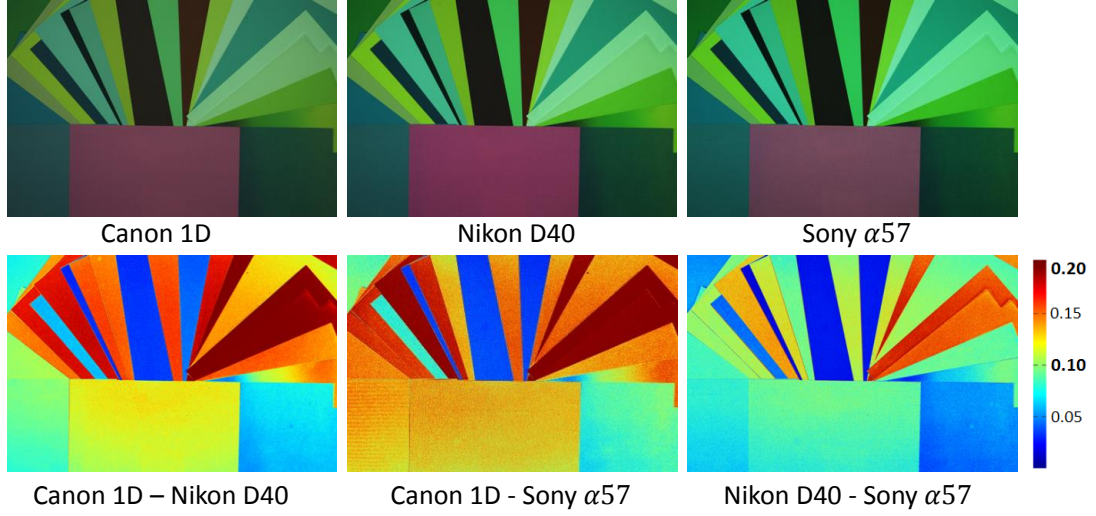


Figure 3.1: Top row shows three RAW images taken from Canon 1D, Nikon D40, and Sony α 57. Bottom row shows the numerical difference as root mean square error (RMSE) between the RAW images. The color map shown on the right explains how much error each color denotes for (e.g. blue color denotes 0% error, while red color denotes up to 20% error).

differs due to the unique spectral responses of sensors found in different camera makes and models.

Contribution The contribution of our work is to conduct an analysis of a range of strategies to perform the RAW-to-RAW mapping between cameras. In particular, linear transformations, radial basis functions, gaussian process regression, and polynomial fitting are evaluated in a global mapping manner (all illuminations combined) and an illumination-specific manner. From this analysis, a new calibration and mapping method is proposed that uses two linear transformations together with a white-balancing step to provide a compact solution that offers state-of-the-art results. The effectiveness of this approach is demonstrated on a wide range of cameras and images. In addition, a data set for studying the RAW-

to-RAW mapping problem has been assembled with over 250 RAW images from eight different cameras of color charts and various scenes under different illuminations for calibration and testing.

3.2 Preliminaries

We begin our preliminaries by discussing the problem of color constancy/white-balancing given its relevance to the RAW-to-RAW mapping problem and its use in our proposed method described in Section 3.4.

Preliminaries We start with the image formation model in the spectral domain, where an rgb image I can be written as:

$$I(\mathbf{x}) = \int_{\omega} R_c(\lambda) S(\mathbf{x}, \lambda) L(\lambda) d\lambda, \quad (3.1)$$

where λ represents the wavelength, ω is the visible spectrum 400 – 700nm, R_c is the camera’s spectral response, and c is the color channel $c = r, g, b$. The term $S(\mathbf{x}, \lambda)$ represents the scene’s spectral response at pixel \mathbf{x} and $L(\lambda)$ is the lighting in the scene, assumed to be spatially uniform.

The color constancy problem can be expressed as follows (dropping the spatial location \mathbf{x} for simplicity):

$$\begin{aligned} I^{L_1} &= \int_{\omega} R_c(\lambda) S(\lambda) L_1(\lambda) d\lambda \\ I^{L_2} &= \int_{\omega} R_c(\lambda) S(\lambda) L_2(\lambda) d\lambda, \end{aligned} \quad (3.2)$$

where R_c and S are assumed to be fixed (i.e. images I^{L_1} and I^{L_2} are taken by the same camera). The difference between the images is due to the different lightings L_1 and

L_2 . The goal in color constancy is to compute a transformation to make the image values of these two images as similar as possible. It is generally accepted that a diagonal 3×3 matrix T is sufficient to map between these images [Chakrabarti et al. 2012; Chong et al. 2007; Gijsenij et al. 2011], where the diagonal matrix T maps the rgb values of I^{L_1} to I^{L_2} . When the target illumination is not specified (as is often the case), the problem becomes one of mapping the source illumination, L_1 , to a canonical illumination. This is typically referred to as white-balancing where an observed or estimated white-point (denoted as r_w, g_w, b_w) in the image I^{L_1} maps to the rgb values (1,1,1). This means that the diagonal entries of T are $1/r_w, 1/g_w, 1/b_w$. White-balancing is a thoroughly studied topic and a variety of techniques to estimate T exist (e.g. [Buchsbaum 1980; Finlayson and Schaefer 2001; Forsyth 1990; Cheng et al. 2014], for an excellent survey see [Gijsenij et al. 2011]).

The RAW-to-RAW mapping problem can be similarly expressed as:

$$\begin{aligned} I_1 &= \int_{\omega} R_{1c}(\lambda) S(\lambda) L(\lambda) d\lambda \\ I_2 &= \int_{\omega} R_{2c}(\lambda) S(\lambda) L(\lambda) d\lambda, \end{aligned} \tag{3.3}$$

where, in this case, the changing variables are the camera response functions, R_{1c} and R_{2c} , i.e. images I_1 and I_2 are from different cameras. Similar to color constancy, the goal is to compute a mapping, denoted as f , such that we can map image I_1 to appear as image I_2 , i.e.:

$$I_2 = f(I_1). \tag{3.4}$$

Ideally, this mapping should work for any scene and lighting condition. Since f does not depend on the image's spatial content, it can be considered a colorspace conversion from I_1 's RAW-RGB to I_2 's RAW-RGB. While similar to color constancy,

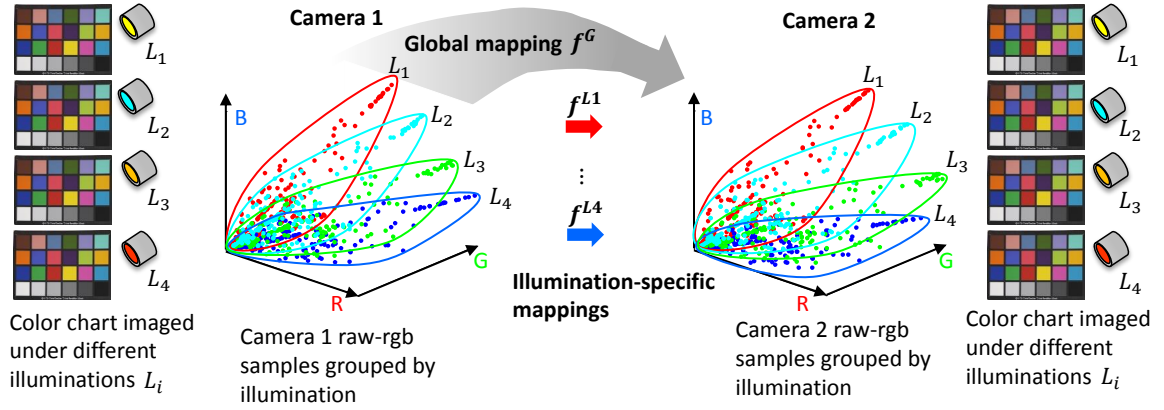


Figure 3.2: This figure shows the RAW-to-RAW calibration setup. Images of color calibration charts are taken under several different lighting conditions by the source and target cameras. Mapping between these two cameras’ RAW-RGB colorspace can be estimated using a global mapping (all illuminations combined) or via multiple illuminant-specific mappings.

the RAW-to-RAW mapping has to compensate for three different color response functions versus one change in spectral lighting. In addition, this mapping needs to be computed in the face of camera-based metamerism.

To estimate f , a number of corresponding RAW-RGB samples between the two cameras colorspace is needed. The most direct way to obtain this is by having both cameras image a calibration color chart under several different illuminations as shown in Figure 3.2. The use of the chart establishes corresponding RAW-RGB between cameras (i.e. patches are unique colors); the multiple illuminations help to span the camera’s gamut and serve to reveal how scene colors S shift due to illumination change L . The question now is how to best estimate f ? We explore this in the next section examining five mapping approaches applied in both a global manner for any type of scene and illumination (denoted as f^G), as well as the more restrictive case of an illumination-specific transformation, denoted as f^L , where L

is a known illumination.

3.3 Evaluating mapping approaches

3.3.1 Mapping methods

We examine five different transformations for approximately estimating f . These are as follows:

Linear transform ($T_{3 \times 3}$) This uses a simple 3×3 matrix to perform the mapping. The values can be solved using any standard least-squares method. Unlike color constancy, the transform is a full matrix, not a diagonal.

Linear transform with offset ($T_{3 \times 4}$) This method extends the 3×3 matrix by adding an 3×1 offset vector to make an affine transform. The matrix and offsets can be solved using a standard least-squares method.

Polynomial model (Poly) This method was proposed in [Hong et al. 2001] and extends the linear transformation by adding high-order terms (e.g., $rg, rb, gb, r^2, g^2, b^2$). The parameters can be obtained by approximately solving for a 3×11 matrix via least-squares.

Radial basis functions (RBF) RBF is a well known scatter point interpolation method which is described as: $f(x) = \sum_{i=1}^N w_i \phi(\|x - c_i\|)$, where the approximating function $f(x)$ is a linear combination of N radial basis functions $\phi(r)$. Each basic function is controlled by a different center c_i estimated from a set of corresponding points given between the source and target space and weighted by an appropriate coefficient w_i that can be computed by linear least-squares manner. For more information see [Buhmann 2003].

Gaussian process regression (GPR) GPR uses a joint Gaussian distribution for estimation: $f(x) \sim N(m, K(\theta, x, x'))$, where m is the mean function value, $K(\theta, x, x')$ is the covariance matrix between all possible pairs (x, x') for a given set of hyperparameters θ . For more information see [Rasmussen 2006].

3.3.2 Global versus illumination-specific

As mentioned in Section 3.2, we evaluate the different mappings in a global and illumination-specific manner. For the global approach, we estimate the mapping using the five models previously described using all of the colorchart samples under different illuminations. For the illumination-specific method, we use only the color samples for a particular illumination.

The results of these five strategies applied as a global mapping and illumination-specific mapping are shown in Table 3.1 in columns one and two. To compute the mapping, two Macbeth color charts (24 patches and 140 patches) is imaged under four different lighting conditions: Fluorescent, Incandescent, Halogen, and LED. The source and target cameras shown here are for a Canon 1D and Nikon D40. A similar trend is shown in Table 3.2 for an Olympus E-PL6 and a Panasonic GX1. For the **RBF** model, we used one more dataset (paper and paint chips described in Section 3.5.1) for cross-validation to control the number of central points to avoid overfitting. The residual errors shown in Tables 3.1 and 3.2 were computed on the color calibration charts and are reported as root mean square error (**RMSE**).

Method	Mapping	Light				
		Fluorescent	Incandescent	Halogen	LED	Average
Global	$T_{3\times 3}$	0.0165	0.0123	0.0077	0.0063	0.0107
	$T_{3\times 4}$	0.0161	0.0105	0.0064	0.0066	0.0099
	Poly	0.0156	0.0084	0.0038	0.0036	0.0079
	RBF	0.0157	0.0084	0.0038	0.0037	0.0079
	GPR	0.0153	0.0067	0.0035	0.0047	0.0076
Illuminant-specific	$T_{3\times 3}$	0.0129	0.0095	0.0052	0.0032	0.0077
	$T_{3\times 4}$	0.0125	0.0068	0.0032	0.0057	0.0063
	Poly	0.0118	0.0062	0.0029	0.0026	0.0059
	RBF	0.0118	0.0062	0.0029	0.0026	0.0059
	GPR	0.0124	0.0063	0.0030	0.0037	0.0061
Ours	$T_{3\times 3}$	0.0135	0.0099	0.0055	0.0036	0.0081
	$T_{3\times 4}$	0.0128	0.0075	0.0037	0.0033	0.0068
	Poly	0.0126	0.0073	0.0037	0.0033	0.0067
	RBF	0.0127	0.0073	0.0037	0.0033	0.0068
	GPR	0.0128	0.0066	0.0032	0.0032	0.0065

Table 3.1: The table shows the comparisons of error in terms of **RMSE** between all linear and non-linear models in three categories: global, specific and white-balancing. We used color calibration charts taken under four lighting conditions: Fluorescent, Incandescent, Halogen, and LED. Average means the average error for all the lightings. The source and target cameras shown here are for a Canon 1D and Nikon D40.

Method	Mapping	Light				
		Fluorescent	Incandescent	Halogen	LED	Average
Global	$T_{3 \times 3}$	0.0085	0.0084	0.0067	0.0072	0.0077
	$T_{3 \times 4}$	0.0070	0.0094	0.0058	0.0082	0.0076
	Poly	0.0060	0.0093	0.0059	0.0077	0.0072
	RBF	0.0060	0.0093	0.0042	0.0077	0.0072
	GPR	0.0053	0.0118	0.0059	0.0077	0.0074
Illuminant-specific	$T_{3 \times 3}$	0.0033	0.0029	0.0034	0.0030	0.0032
	$T_{3 \times 4}$	0.0026	0.0024	0.0028	0.0030	0.0027
	Poly	0.0024	0.0023	0.0027	0.0030	0.0026
	RBF	0.0024	0.0023	0.0028	0.0030	0.0026
	GPR	0.0025	0.0024	0.0028	0.0030	0.0027
Ours	$T_{3 \times 3}$	0.0034	0.0030	0.0035	0.0032	0.0033
	$T_{3 \times 4}$	0.0027	0.0025	0.0030	0.0034	0.0029
	Poly	0.0026	0.0024	0.0029	0.0033	0.0028
	RBF	0.0026	0.0024	0.0029	0.0033	0.0028
	GPR	0.0025	0.0023	0.0028	0.0034	0.0027

Table 3.2: The table shows the comparisons of error in terms of **RMSE** between all linear and non-linear models in three categories: global, specific and white-balancing. We used color calibration charts taken under four lighting conditions: Fluorescent, Incandescent, Halogen, and LED. Average means the average error for all the lightings. The source and target cameras shown here are for an Olympus E-PL6 and a Panasonic GX1.

3.3.3 Discussion

It is clear from Tables 3.1 and 3.2 that among all the approaches for modeling f , the illumination-specific transformations give the lowest errors. Not surprisingly, the more complex transformation methods such as **RBF** and **GPR** provide better fitting in both the global and illumination-specific approaches. However, this finding comes at a cost in terms of evaluation time, where **RBF** and **GPR** can take several minutes to convert an image, while the other transformations can be done in a matter of seconds.

The obvious downside of the illumination-specific strategy is the need to store several mappings for the most common illuminations. This also requires that when an image is converted its illumination must be known to select the correct transformation. This makes illumination-specific mappings less attractive for practical applications for RAW-to-RAW mapping.

It is important to note that the fitting errors from the global methods were not distributed equally for all color values. For example, RAW-RGB values that represent illumination (i.e. the gray/white patches on the colorchart) report a lower error than the average error of the other colors. Table 3.3 shows these errors using the $T_3 \times 4$ transformation in terms of percentage error to remove the effect of the magnitude of the RAW-RGB values. This finding led us to develop an alternative to the illumination-specific method that provided similar results but in a much more compact manner. This approach is described in the following section.

	Canon1D		Nikon D40		Sony $\alpha 57$	
	W	C	W	C	W	C
Canon 1D	-	-	3.76	5.37	4.71	5.61
Nikon D40	3.65	5.27	-	-	5.34	6.31
Sony $\alpha 57$	4.95	5.83	5.46	6.60	-	-

Table 3.3: The table shows the comparisons of percentage error (in %) between white points (W) and color points (C) by the global transform.

3.4 Proposed illumination-independent method

Figure 3.3-(A) overviews our proposed calibration procedure, and Figure 3.3-(B) describes how to perform the RAW-to-RAW conversion. Both are described in detail below.

Calibration The approach starts by computing a global transformation f^G between the two cameras RAW-RGB colorspaces. Next, color values for each illumination are transformed by a white-balance transformation which uses the white-patches on the color chart to estimate the white points. This results in both cameras illuminations being mapped to a canonical illumination. Finally, a mapping, f^{L_c} , is computed between these transformed colorspaces, where the superscript L_c denotes the canonical illumination. This means that our method needs to compute only two mappings, a global mapping, f^G , and a canonicalized mapping f^{L_c} .

RAW-to-RAW mapping Consider an input source image I_s^L taken under an unknown illuminant L . Our goal is to transform this to another camera’s colorspace as if it was taken of the same scene and under the same illumination, i.e. to target image I_t^L . First, the white-point of I_s^L is estimated, i.e. (r_{ws}, g_{ws}, b_{ws}) . This can be done

with any robust white-balancing algorithm (e.g. [Chakrabarti et al. 2012; Chong et al. 2007; Gijsenij et al. 2011]). Next the source image is white-balanced using the estimated T_{ws} . This results in a new image, denoted as $I_s^{L_c}$, that has a canonical illumination. From this, the image can be mapped to the canonical lighting in target RAW-RGB space via $I_t^{L_c} = f^{L_c}(I_s^{L_c})$.

After conversion using f^{L_c} it is still necessary to map the target image $I_t^{L_c}$ back to the correct input illumination. We cannot use the estimated white-point (r_{ws}, g_{ws}, b_{ws}) , since this was estimated in the source colorspace. Instead, we find the target white-point by using the global transform, such that $(r_{wt}, g_{wt}, b_{wt}) = f^G(r_{ws}, g_{ws}, b_{ws})$. As previously discussed, while the global transformation f^G is not ideal for all colors, it has acceptable performance on color values associated with illuminations. Once we have this target white point, we can obtain the desired I_t^L by applying the inverse white-balance transform, T_{wt}^{-1} .

Transformation used Table 3.1 (last column) shows our proposed method implemented using the five different transformation methods discussed in Section 3.3.1. For the mapping models, while the non-linear mapping models perform better, the linear offset model follows them closely. Further, the inherent simplicity and lower dimensionality of this model scores over the non-linear models. Thus, we choose to implement our canonical illumination method using linear offset model (i.e. $T_{3 \times 4}$) for estimating f^G and f^{L_c} used in our method.

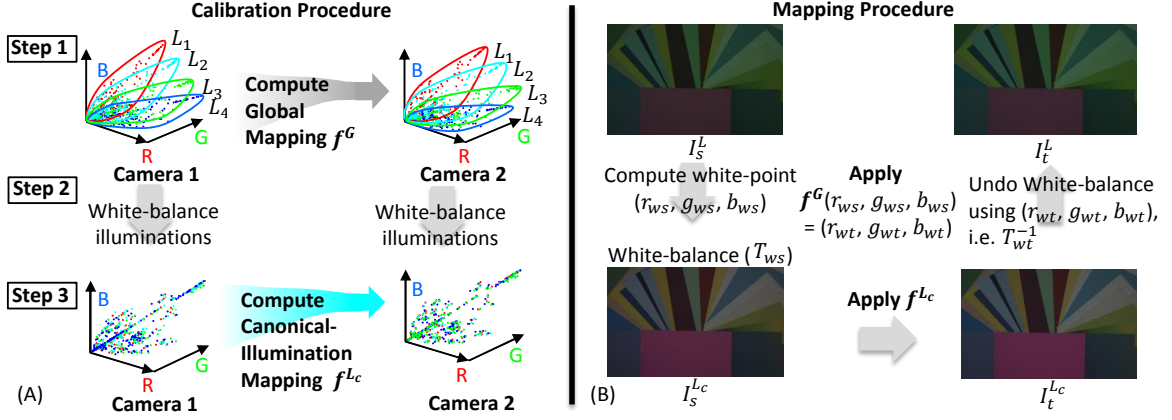


Figure 3.3: This figure shows the overview of our RAW-to-RAW calibration and conversion approach. (A) shows the steps of our calibration procedure. A global mapping f^G is computed using all of the color chart points. White-balancing is then applied to the color charts values from both cameras. Next a mapping on the canonical-illumination, f^{L_c} is computed. (B) illustrates the conversion procedure.

3.5 Experiments and results

We compare our method with the method presented in [Hong et al. 2001] and the Adobe’s Digital Negative (DNG) software. The experiments are tested on two different data sets. The first is an image set of paint and paper samples for which explicit point correspondences can be extracted easily (denoted as the *controlled image set*). The second data set is a series of outdoor scenes (termed *outdoor image set*) taken by multiple cameras from the same location. The global and canonical illuminations mapping are computed only once as described in Section 3.4 from RAW-RGB samples obtained from two Macbeth color charts imaged under six illuminations with four lighting conditions (two types of fluorescent, two types of incandescent, halogen, and LED light). Calibration and conversion code is written in Matlab and takes less than a second for RAW-to-RAW conversion.

Adobe outlines their procedure for DNG conversion in [Adobe] and recom-

mends estimating a $T_{3 \times 3}$ mapping for only two different lighting conditions, L_1 and L_2 . It is recommended that these lighting conditions be “far apart” from one another in terms of color temperature. From this, two transformations can be computed, T^{L_1} and T^{L_2} . Given a RAW input image the illumination is obtained from the RAW file’s metadata (i.e. the white-balance setting), and the RAW-to-RAW transformation, f , is estimated as: $f = w_1 T^{L_1} + w_2 T^{L_2}$ where the weights w_1 and w_2 are based on the similarity of the input image’s illuminant, L to the two illuminates L_1 and L_2 , s.t. $w_1 + w_2 = 1$. Details to how these weights are computed is not explicitly given.

The Adobe SDK does not explicitly map between two different cameras, but instead maps images to the DNG format which acts as canonical color space. Note that this is different from our proposed canonical illumination space. To make sure that the reported errors are in the same scale space, we use DNG space to compute a direct camera to camera mapping. This is done by converting the two images from the two cameras, denoted as I_s and I_t , to their DNG representation I_s^D and I_t^D using the DNG software. We can now compute a 3×3 linear transformation T_s^D that maps I_s to I_s^D . The transform T_t^D can be computed similarly for I_t . Since I_s^D and I_t^D are in canonical space, we have $I_t = (T_t^D)^{-1} \times T_s^D \times I_s$. We found that the residual errors in computing these T_s^D and T_t^D were very low, indicating that this was a reasonable strategy for comparison.

While the images in our data set have been taken with the same exposure settings, small differences attributed to lens optics and sensor gain require the use of a scale factor k to adjust the two RAW images to have similar exposure before evaluating the residual error. Here $k = \sum_{\mathbf{x}} I_{t,\mathbf{x}}^l / \sum_{\mathbf{x}} \hat{I}_{t,\mathbf{x}}^l$, where \mathbf{x} is the pixel location, I_t is the ground-truth target image, and \hat{I}_t is a converted RAW image.



Figure 3.4: Example images of the controlled image set of paint chips and paper samples. The cyan rectangles are regions used for extracting the RAW values.

3.5.1 Controlled image set

This data set consists of images of colored paint chips and paper samples. Examples are shown in Figure 3.4. These are imaged under indoor fluorescent, tungsten and outdoor daylight illuminations with the following cameras: Canon 1D Mark III, Canon EOS 600D, Nikon D40, Nikon D5200, Olympus E-PL6, Panasonic DMC-GX1X, Samsung NX2000, and Sony α 57. Images are aligned using a homography and further cropped to ensure that only the colors are being compared. A jet map is used to show the pixel error between the transformed RAW-RGB images and the ground truth RAW-RGB images.

We also show the following pixel error statistics: mean (M) error, 25% quartile (Q1) error, 50% quartile (Q2) error (i.e. median error) and 75% quartile (Q3) error. Examples of RAW-to-RGB conversions are shown in Figures 3.5, 3.6, 3.7, and 3.8. The error map as well as the mean and quartile errors show that the proposed method results in lower conversion errors

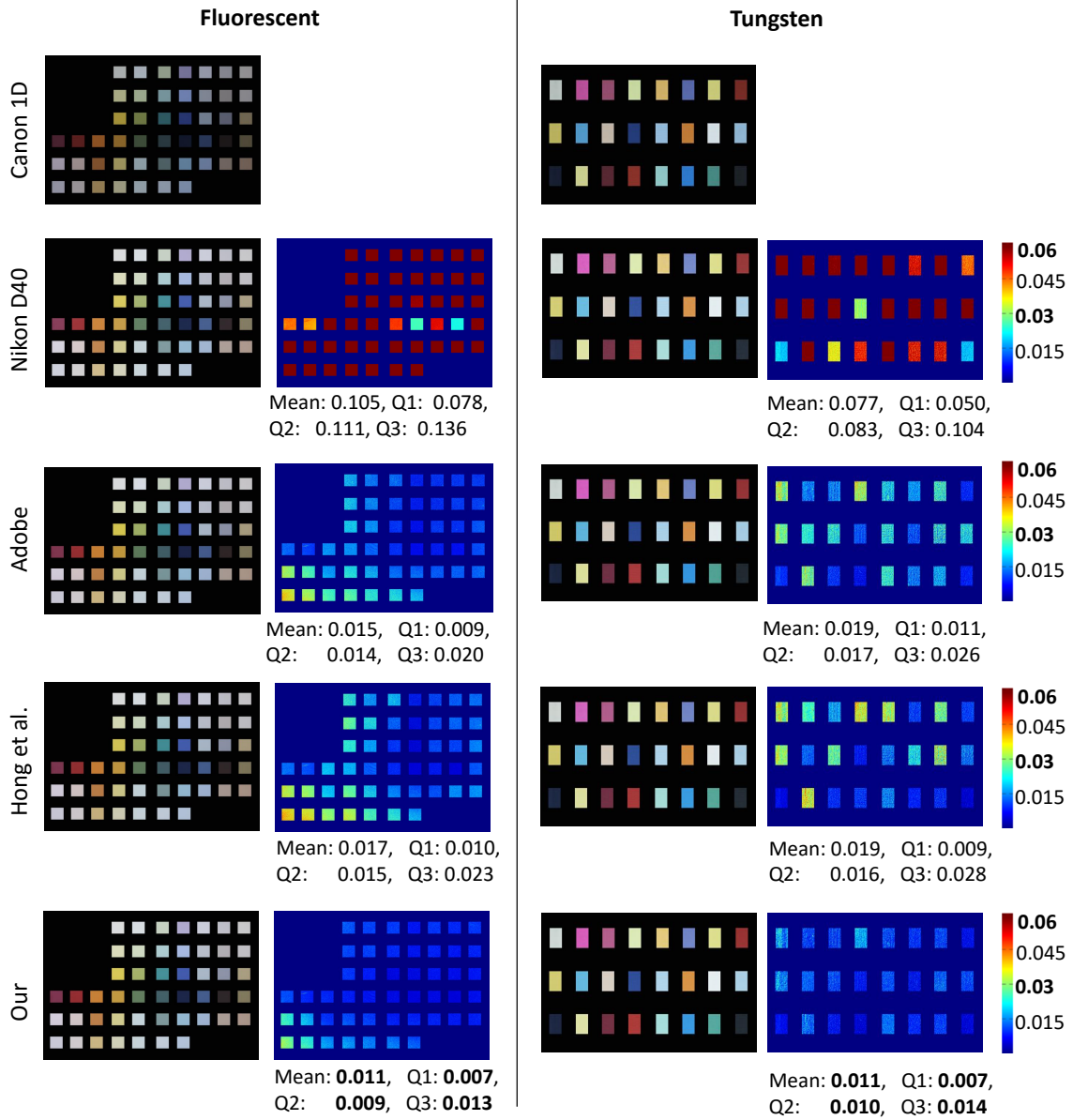


Figure 3.5: Comparison between all approaches. This figure shows the results on a Canon 1D and Nikon D40. Two lighting conditions: fluorescent and tungsten are shown with the camera setting given to the DNG software. Results show the mean RAW pixel errors (normalized) and the errors at the 25%, 50% (median) and 75% quartiles (Q1, Q2, Q3).

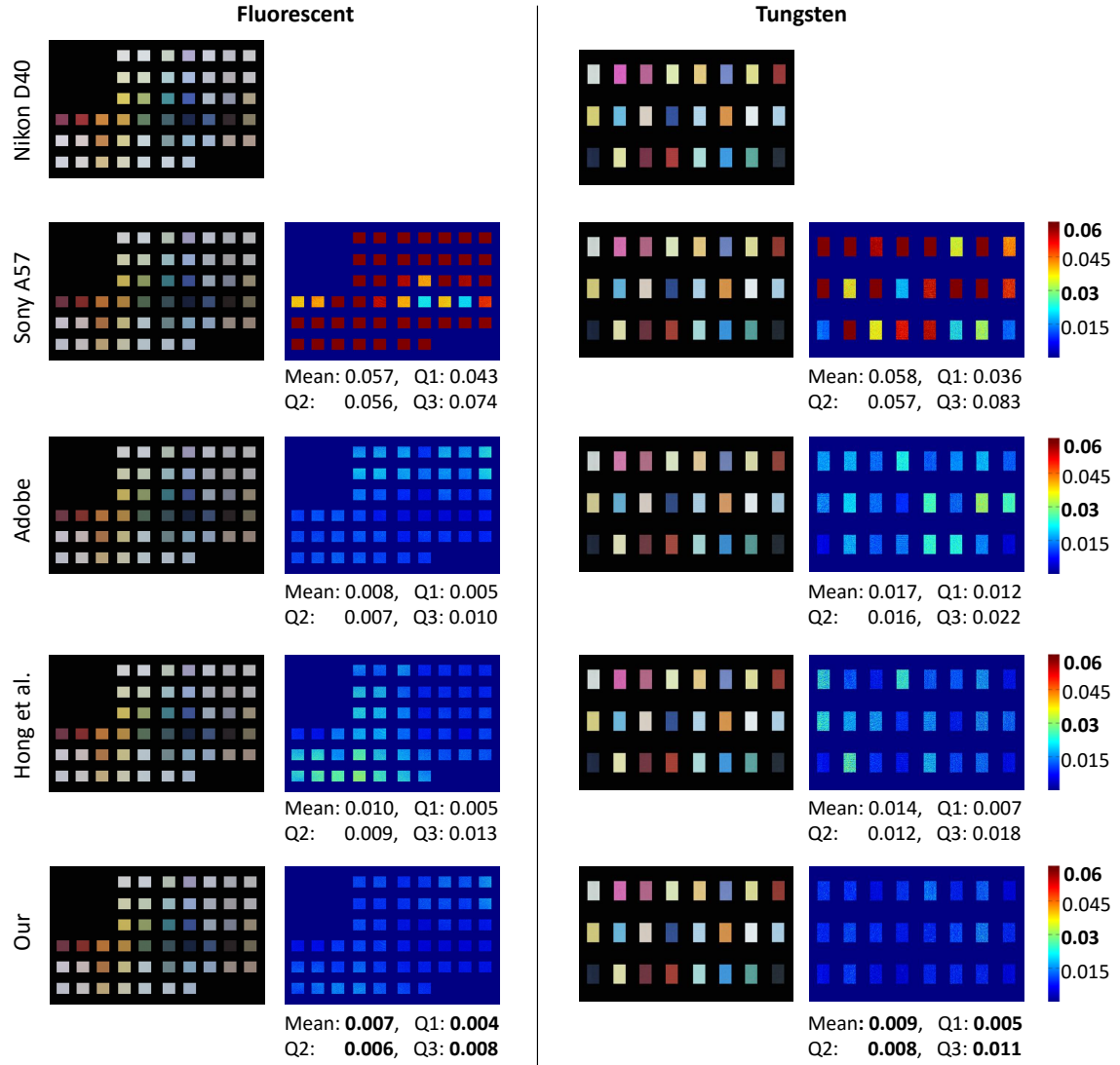


Figure 3.6: Comparison between all approaches. This figure shows the results on a Nikon D40 and a Sony α 57. Two lighting conditions: fluorescent and tungsten are shown with the camera setting given to the DNG software. Results show the mean RAW pixel errors (normalized) and the errors at the 25%, 50% (median) and 75% quartiles (Q1, Q2, Q3).

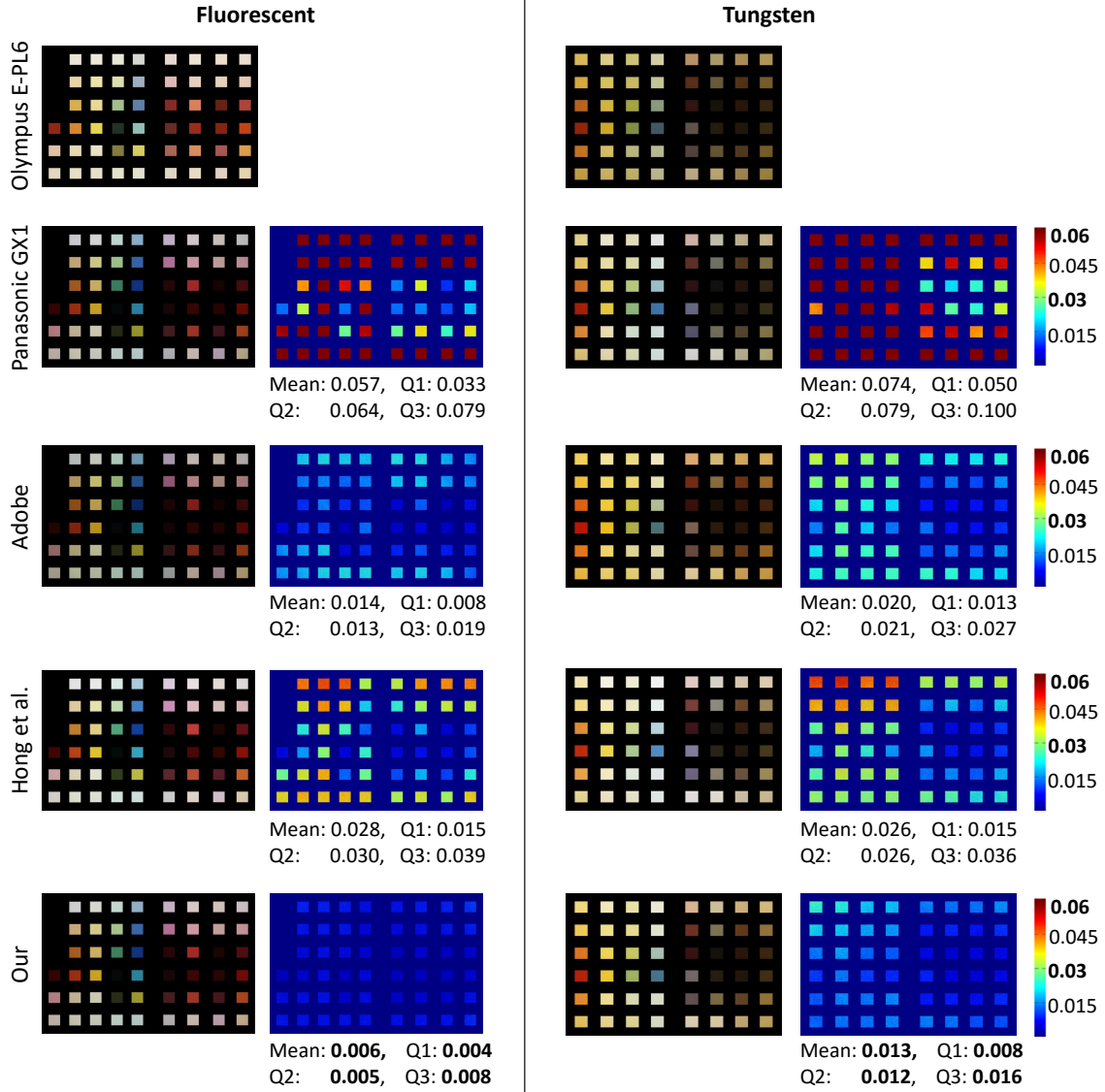


Figure 3.7: Comparison between all approaches. This figure shows the results on a Olympus E-PL6 and a Panasonic GX1. Two lighting conditions: fluorescent and tungsten are shown with the camera setting given to the DNG software. Results show the mean raw pixel errors (normalized) and the errors at the 25%, 50% (median) and 75% quartiles (Q1, Q2, Q3).

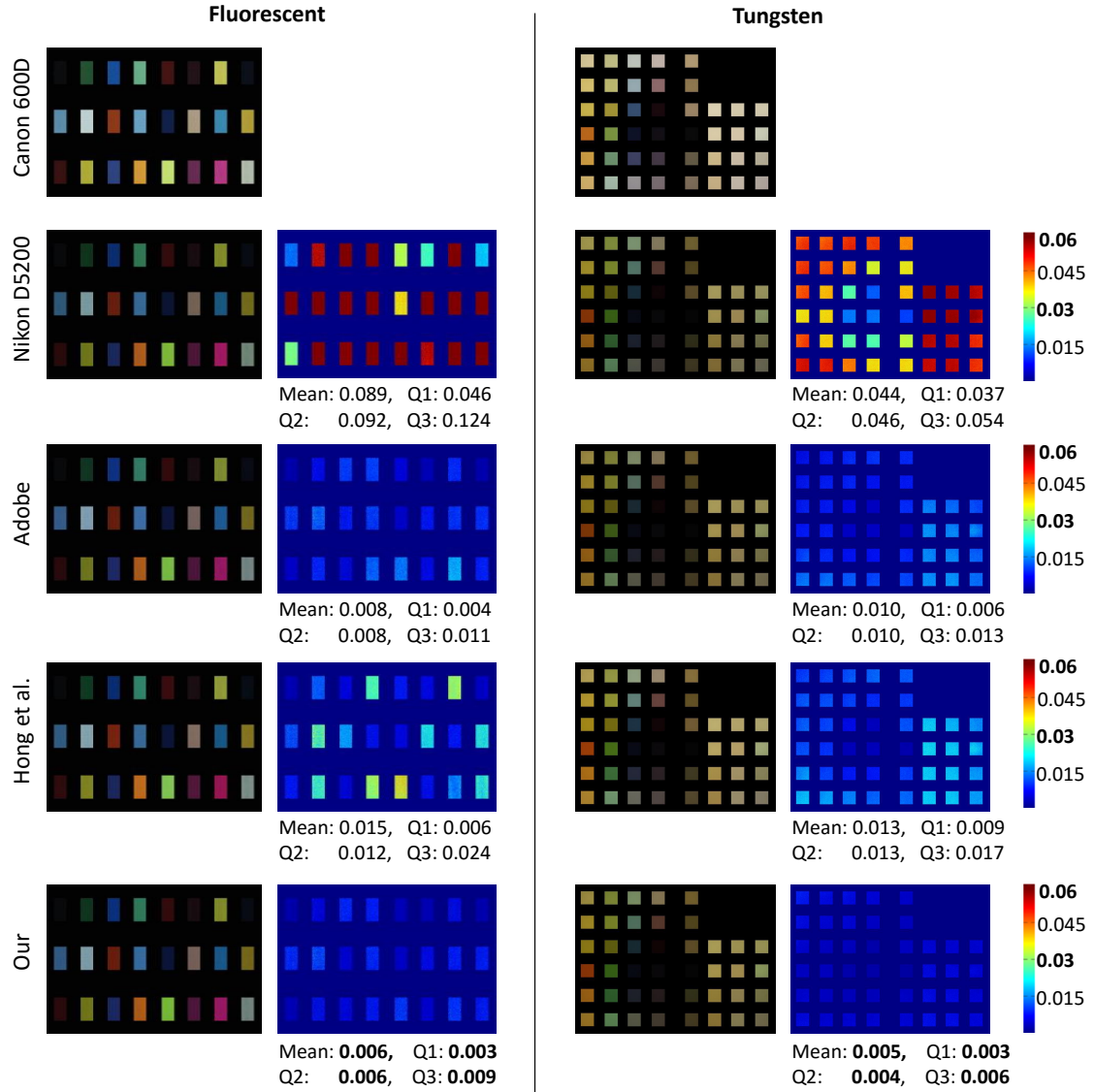


Figure 3.8: Comparison between all approaches. This figure shows the results on a Canon 600D and a Nikon D5200. Two lighting conditions: fluorescent and tungsten are shown with the camera setting given to the DNG software. Results show the mean RAW pixel errors (normalized) and the errors at the 25%, 50% (median) and 75% quartiles (Q1, Q2, Q3).

3.5.2 Outdoor image set

Figure 3.9 shows examples from the *outdoor image* data set. Care is taken to align and crop the images to be similar, however, there are still some slight misalignment. This makes it hard to evaluate the error by using point wise subtraction. Instead, we examine how well aligned the RAW-RGB color histograms of the ground truth and converted image are. The symmetric Kullback-Leibler (KL) divergence [Kullback and Leibler 1951] is used to measure the histogram distance:

$$\begin{aligned} D &= D_{KL}(H_t || \hat{H}_t) + D_{KL}(\hat{H}_t || H_t) \\ &= \sum_i \log\left(\frac{H_t(i)}{\hat{H}_t(i)}\right) H_t(i) + \sum_i \log\left(\frac{\hat{H}_t(i)}{H_t(i)}\right) \hat{H}_t(i) \end{aligned} \quad (3.5)$$

where H_t is the histogram of the target image I_t , \hat{H}_t is the histogram of the transformed image \hat{I}_t , and i is the index of each histogram bins up to 2^{12} (equal to the maximal level of the RAW image). All these histogram are normalized by their number of pixels. The distance is computed per color channel with the mean reported as the KL divergence distance.

The comparison between Adobe DNG, [Hong et al. 2001], and our approach is given in Tables 3.4, 3.5, 3.6, and 3.7. We tested on six different cameras: Canon 1D Mark III, Nikon D40, Sony α 57, Olympus E-PL6, Panasonic GX1, and Samsung NX2000. For each pair of cameras, eight pairs of RAW images are examined and the mean values of the KL divergence distances are shown in Tables 3.4 3.5, 3.6, and 3.7. Our proposed method performs better than the other two methods.



Figure 3.9: The figure shows example images of outdoor image set.

	Canon1D				Nikon D40				Sony α 57			
	B	A	H	O	B	A	H	O	B	A	H	O
Canon 1D	-	-	-	-	0.87	0.23	0.09	0.05	0.44	0.28	0.15	0.14
Nikon D40	0.87	0.19	0.11	0.06	-	-	-	-	0.69	0.19	0.20	0.16
Sony α 57	0.44	0.14	0.05	0.04	0.69	0.08	0.08	0.05	-	-	-	-

Table 3.4: The table shows the comparisons of histogram distance computed by the equation 3.5 between all the approaches from three cameras: Canon 1D, Nikon D40, and Sony α 57. For each pair of the cameras, four results are reported namely Before (B), Adobe (A), Hong et al. (H), and Ours (O).

	Olympus E-PL6				Panasonic GX1				Samsung NX2000			
	B	A	H	O	B	A	H	O	B	A	H	O
Olympus	-	-	-	-	0.95	0.18	0.08	0.04	0.67	0.26	0.26	0.04
Panasonic	0.95	0.45	0.28	0.02	-	-	-	-	0.32	0.10	0.29	0.05
Samsung	0.67	0.23	0.20	0.02	0.32	0.07	0.07	0.06	-	-	-	-

Table 3.5: The table shows the comparisons of histogram distance computed by the equation 3.5 between all the approaches from three cameras: Olympus E-PL6, Panasonic GX1, and Samsung NX2000. For each pair of the cameras, four results are reported namely Before (B), Adobe (A), Hong et al. (H), and Ours (O).

	Canon 600D				Nikon D5200				Olympus E-PL6			
	B	A	H	O	B	A	H	O	B	A	H	O
Canon	-	-	-	-	0.19	0.05	0.03	0.01	0.22	0.02	0.02	0.01
Nikon	0.19	0.07	0.06	0.01	-	-	-	-	0.18	0.12	0.07	0.04
Olympus	0.22	0.02	0.03	0.01	0.18	0.10	0.04	0.02	-	-	-	-

Table 3.6: The table shows the comparisons of histogram distance between all the approaches from three cameras: Canon 600D, Nikon D5200, and Olympus E-PL6. For each pair of the cameras, four results are reported namely Before (**B**), Adobe (**A**), Hong et al. (**H**), and Ours (**O**).

	Canon 600D				Sony $\alpha 57$				Panasonic GX1			
	B	A	H	O	B	A	H	O	B	A	H	O
Canon	-	-	-	-	0.12	0.07	0.08	0.05	0.16	0.07	0.04	0.03
Sony	0.12	0.02	0.09	0.01	-	-	-	-	0.16	0.04	0.10	0.03
Panasonic	0.16	0.05	0.04	0.02	0.16	0.04	0.05	0.03	-	-	-	-

Table 3.7: The table shows the comparisons of histogram distance between all the approaches from three cameras: Canon 600D, Sony $\alpha 57$, and Panasonic GX1. For each pair of the cameras, four results are reported namely Before (**B**), Adobe (**A**), Hong et al. (**H**), and Ours (**O**).

3.6 Example application

Here we show an application of the RAW-to-RAW conversion using our approach to mosaic images captured from different cameras. Three different cameras Nikon D40, Sony α 57, and Canon 1D are used. All three images are taken under the same exposure settings. Mosaicing without RAW-to-RAW conversion is shown in the top row of Figure 3.10. The Canon and Nikon images are converted to be in the Sony RAW-RGB colorspace based on our pre-computed transformation using the color charts described in Section 3.4 (no other blending or color conversion is applied). Figure 3.10 shows that after RAW-to-RAW conversion the mosaic has better color continuity and lesser apparent seams.

3.7 Discussion and Summary

This chapter has examined the problem of converting between camera’s RAW-RGB color spaces. We examined five different mapping models applied in a global and illumination-specific manner. Our results show that illumination-specific approach gives the best results, but at the disadvantage of maintaining multiple transformations and categorizing input images to scene illumination. To overcome this problem, we proposed an illumination-independent method that uses white-balancing to canonicalize the illumination. This method allows us to perform the RAW-to-RAW mapping using only two linear transformations. We described the calibration and our mapping procedure and showed its effectiveness on a range of inputs and under different illuminations. We have also prepared a data set useful in further exploration of the RAW-to-RAW conversion problem.

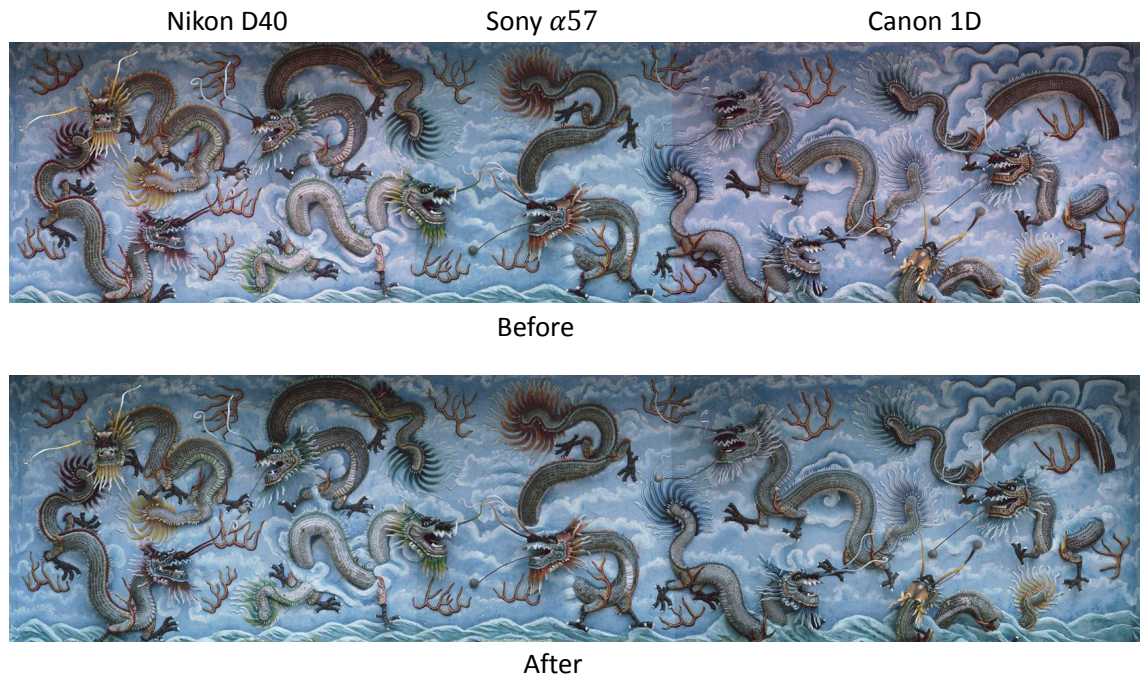


Figure 3.10: This figure shows an example of image mosaicing application. Three different cameras Nikon D40, Sony $\alpha 57$, and Canon 1D are used. This figure shows the comparison before and after conversion. All the images are converted to the RAW-RGB space of the Sony $\alpha 57$. These mosaics have been adjusted by a gamma for better visualization.

Like other related work, our work also suffers from the metamerism. Metamers arise when two different spectral power distributions map to the same raw-rgb value in a camera because of reducing from very large dimensions (up to hundreds) to only three dimensions. According to our experiment, around 10% of colors can cause metamers. This can lead to a many-to-one or one-to-many mapping between the cameras. A solution of metamerism is an interesting future work.

Chapter 4

Raw to Photo-finished sRGB Output Mapping

In this chapter, we investigate the color mapping between the camera sensor RAW and its final photo-finished sRGB image. In particular, we present a method to encode the necessary metadata within an sRGB image to reconstruct a high-quality version of its original corresponding RAW image. Our approach requires no calibration of the camera and can reconstruct the original RAW to within 0.3% error with only a 64 KB overhead for the additional data. More importantly, our output is a fully self-contained 100% compatible sRGB-JPEG file that can be used as-is, not affecting any existing image workflow. The RAW image can be extracted when needed, or ignored otherwise. We categorize this work as part of the color calibration methods and show its effectiveness against competing strategies.

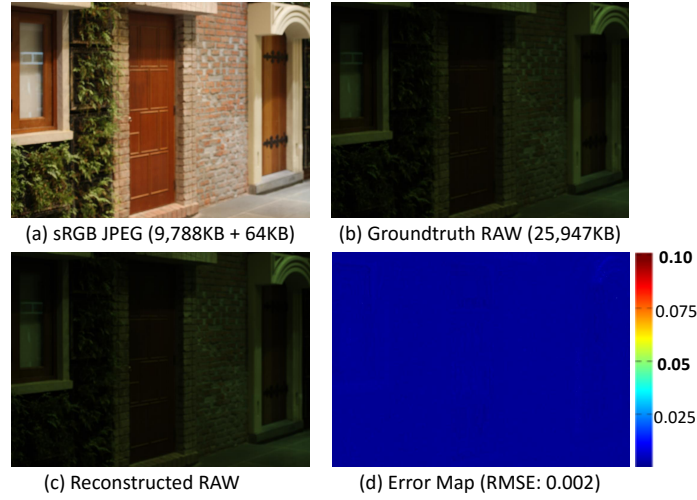


Figure 4.1: (a) A 5616×3744 resolution high-quality sRGB-JPEG with our metadata embedded Original JPEG size (9,788 KB); new size (9,852 KB). (b) Original RAW image is 25,947 KB. (c) Our reconstructed RAW image using the data in the self-contained JPEG. (d) Error map between (b) and (c). Overall reconstruction error is 0.2%.

4.1 Introduction

Although the previous chapter has described the benefits of RAW images, using RAW images as input is not popular in many computer vision and image processing applications. The main reason is that RAW files takes up significantly more space than their sRGB counterpart. In addition, RAW images are not ready for visualization and must undergo some intermediate processes to convert them into sRGB to be useful for many existing tasks. Therefore, the vast majority of images used in these applications are still standard RGB (sRGB) images, typically saved using the JPEG compression standard. There are several drawbacks, however, when working with sRGB images, e.g. it is well known that sRGB images have a number of non-linear operations applied that makes it difficult to related the sRGB values back to scene radiance [Debevec and Malik 2008; Grossberg and Nayar

2003b; Mann et al. 1995; Mitsunaga and Nayar 1999; Chakrabarti et al. 2014; Kim et al. 2012]. Therefore, it will be better if both advantages of RAW and sRGB can be utilized.

Given the utility of RAW image data, there has been a number of approaches to map sRGB images back to their RAW values. Work by Yuan and Sun [Yuan and Sun 2011] demonstrated an effective hybrid-image method that stored a smaller resolution version of the original RAW image (e.g. $\frac{1}{2}$ or $\frac{1}{4}$ resolution) and applied smart upsampling that leveraged the sRGB image. One impetus for [Yuan and Sun 2011] is that many cameras now support a small-RAW format that save the RAW image in either half and quarter-size resolutions. However, it is important to note that these smaller RAW images still require roughly 1.5 – 6 MB to store. Other work [Xiong et al. 2012; Chakrabarti et al. 2014; Kim et al. 2012; Lin et al. 2011; Lin et al. 2012] used several images to compute parameters to model the various steps in the on-board camera processing pipeline. These approaches are intended to restore arbitrary images captured from a particular camera. While this metadata is generally smaller than the 1.5 – 6 MB needed using the small-RAW strategy, these methods still have a drawback that additional metadata needs to be stored separately for reconstructing the RAW image. The goal of this work is to provide a fully self-contained JPEG image that allows RAW image reconstruction. In addition, we want to do this with a small memory overhead and in a manner that is 100% compatible with existing JPEG standards. Figure 4.1 shows an example of our proposed method’s ability. An sRGB-JPEG image is emdedded with an additional 64 KB metadata can be used to reconstruct the RAW image with an overall reconstruction error of less than 0.2% (in RAW pixels values).

Contribution We provide a straight-forward and effective procedure to extract the

necessary data for reconstructing a RAW image given the corresponding sRGB-JPEG image. As part of this procedure, we describe a fast breadth-first-search octree algorithm for finding the necessary control points to provide a mapping between the sRGB and RAW sensor color spaces that allows the number of octree cells to be controlled. In addition, we also describe a method to encode our data efficiently within the allowed 64 KB text comment field that is supported by the JPEG standard. This allows our method to be fully compatible with existing JPEG libraries and workflows. We compare our approach with existing methods and demonstrate the usefulness of the reconstructed RAW on two applications: white-balance correction and image-deblurring.

4.2 Proposed Approach

The work by Kim et al. [Kim et al. 2012] and Chakrabarti et al. [Chakrabarti et al. 2014] have shown the processing from RAW to sRGB can be described by a number of parameters that model various steps in the onboard camera color processing pipeline. In particular, there is a white-balance matrix T_w , and a color correction matrix T_s , that is first applied. The color correction matrix is used to convert the camera-specific RAW color space into a device-independent color space. This is followed by a tone map operator f applied to all color channels and a gamut mapping g that maps an 3-D color value to a new 3-D color value. The work in [Kim et al. 2012; Chakrabarti et al. 2014] parameterized these models by using many sRGB-JPEG pairs of color charts under different illuminations captured by the camera with specific settings. This type of parameterization would be applicable for generic input images. We follow a similar procedure, but our problem is slightly

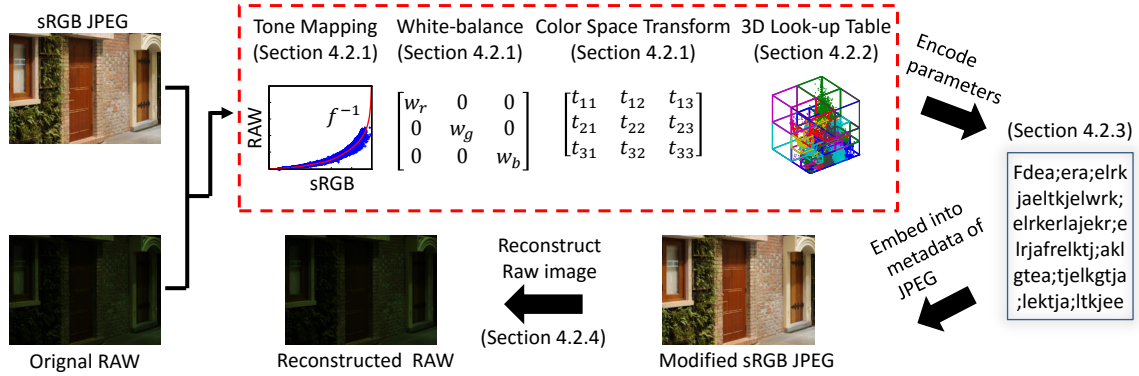


Figure 4.2: This figure shows an overview of our approach. The section of the detailing each component is shown.

simplified as we only need to compute these parameters for a single pair of RAW image E and sRGB-JPEG image I . In addition, our approach needs to keep the results to within 64 KB overhead and embed this as a text field in the JPEG image.

Figure 4.2 provides a diagram that illustrates our steps. Our input is the RAW image captured from the camera and the corresponding sRGB-JPEG. We assume the RAW image has been demosaiced and use the DCRAW utility [Coffin 1997] to perform this task. The data storage budget for each part of the camera model is pre-allocated as shown in Table 4.1. The total budget is less than 64 KB because it will later be converted to a text format that avoids a 0x00 bit sequence (described in Section 4.2.3). The following sections describe how the metadata is computed and embedded in the sRGB-JPEG file.

4.2.1 In-Camera Imaging Model Estimation

Our first step is to compute an inverse tone-curve, f^{-1} , from a pair of sRGB-JPEG and RAW images. This is used to make the sRGB values more linear with respect to the RAW value. We assume that the color correction matrix did not change

	Original Size	Type	Storage Size (byte)
T_w^{-1}	3×1	double	24
T_s^{-1}	3×3	double	72
f^{-1}	256×1	int16	512
g^{-1}	4728×6	int16	56,736

Table 4.1: This table shows the amount of data allocated to model the camera-pipeline parameters into the metadata of a JPEG image. The g^{-1} allows up to 4728 control points pairs consisting of an sRGB and RAW-rgb color point (i.e. 6 values in total).

the brightness of the RAW colors. Additionally, it is assumed that the gamut-map targets chromatic colors, e.g. $g(\mathbf{p}) = \mathbf{p}'$ where \mathbf{p} is an achromatic color (i.e. sRGB saturation value less than 0.2). Based on these two assumptions, the images are converted into the HSV color space, and the V channels are selected to estimate the inverse tone-curve f^{-1} . This curve can be estimated using the spline fitting technique [Reinsch 1967] as follows:

$$\begin{aligned}
f^{-1} &= \arg \min_{f^{-1}} \frac{1}{N} \sum_{i=1}^N \|f^{-1}(I_i) - E_i\|^2 + \lambda \|\nabla^2 f^{-1}\|^2, \\
&\text{s.t. } \nabla^1 f^{-1} \geq 0
\end{aligned} \tag{4.1}$$

where i is the index to color pixels, N is the number of color points after applying a saturation threshold, ∇^1 denotes the first derivative, and ∇^2 denotes the second derivative. The first term measures the agreement of f^{-1} with the observations while the second term constraints the smoothness of the curve. The weight λ controls the relative contribution of the two ($\lambda = 0.1$ for all our examples). There is also a hard constraint that the function is monotonically increasing. Note that color values with any channel set to 255 are not used as they represent fully saturated pixels are not reliable. Figure 4.3 shows an example of with/without using the

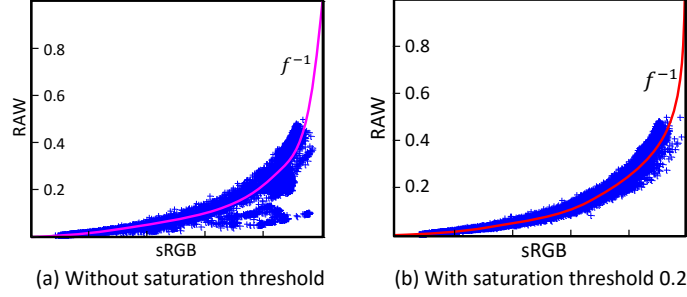


Figure 4.3: This figure shows an example of with/without using saturation threshold for estimating an inverse tone-curve f^{-1} .

saturation threshold for estimating an inverse tone-curve f^{-1} .

After the tone mapping f is estimated, sRGB values are converted to linearized sRGB values. As with the tone-curve, the linear color correction matrix T_c is computed by using the color values with low color saturation that are not affected by the gamut mapping. Here, the color correction matrix T_c is the combination of the white balance matrix T_w and the color space transformation matrix T_s . We estimate the matrix T_c that minimize the following error function:

$$T_c = \arg \min_{T_c} \sum_{i=1}^N \|f^{-1}(I_i) - T_c E_i\|^2. \quad (4.2)$$

Note that most of consumer cameras that are supported RAW format often embeds the white-balance matrix T_w with the RAW files. With T_w , we can obtain the color correction matrix T_s from T_c ($T_c = T_s \times T_w$). Decomposing the color correction matrix into two matrices: white-balance and the color space transformation (T_s) will have several advantages for editing tasks such as the white-balance modification.

According to [Kim et al. 2012], the color gamut mapping g can potentially be highly non-linear and challenging to model using parametric functions. We therefore use scattered point interpolation to model this mapping as done in [Kim

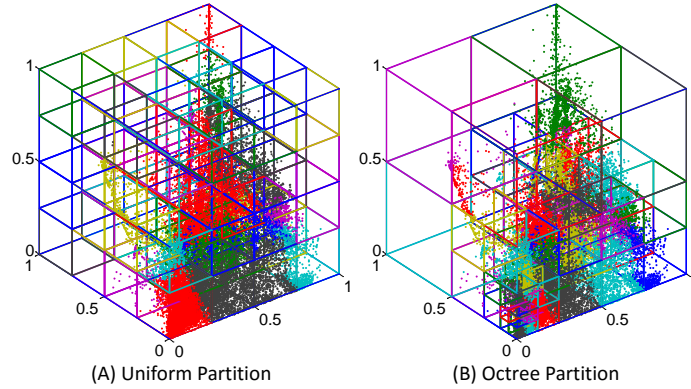


Figure 4.4: This figure shows an example of partition color space using uniform and octree approaches. The same number of bins $64 = 4^3$ is used for both two approaches.

et al. 2012]. These scattered points are sampled from the input image. We examined three different strategies to select the scattered points, namely: uniform sampling, k-means clustering, and octree partitioning. The mean values for each partition or cluster are chosen as the control points.

It is worth noting that the sRGB and RAW colors in an image are rarely distributed through the whole color space. Therefore, using uniform sampling for the color space has two disadvantages. The first is that many samples are empty and need to be ignored. This makes it hard to know the number of non-empty samples before hand and therefore challenging to efficiently control the exact number of non-empty samples. Second, the number of colors are not distributed equally, and non-empty samples may not represent a good usage of allocating control points. Figure 4.4-(a) shows an example using uniform sampling. As such, using something like a lattice regression [Lin et al. 2012] is not the right fit for this problem. Lattice regression (and the non-uniform version) assumes a dense sample over the full RGB color space - the input is multiple images of color charts. This needs to be

done for different settings of the camera. One possible solution is to use k-means clustering. However, the obvious drawback for k-means clustering is the required running-time since the number of points and clusters are relative large (more than 10^6 and 10^3 respectively). In this work, we adapted an octree algorithm proposed by Meagher [Meagher 1980] to partition the RGB color space. To do so, we introduce a slight modification to the traditional octree that is based on a depth-first mechanism, to one that uses a breadth-first approach. This breadth-first octree constructs allows us to control the number of non-empty partitions and sample more scatter points in dense regions (as shown in Figure 4.4-(b)). The details of the octree implementation is presented in the next section.

4.2.2 Modified Octree Partitioning

The basic approach for octree partitioning is to a depth-first search that explores as far as possible along each branch before backtracking. The implementation starts with a single bin surrounding all 3D input points. This bin is recursively subdivided into eight smaller bins. Recursion is stopped when a certain criteria is met. Three common conditions to stop bin subdivision are: 1) A bin contains fewer than a given number of points δ_p ; 2) A bin reaches the minimum size or volume δ_l ; 3) A bin reaches the maximum depth δ_d .

However, in our situation, we need to keep track of the number of non-empty bins and stop the partition process when it reaches a given number. Using the depth-first search strategy, some of the tree nodes may be divided many times while the others may not be divided although they are similar size and contain similar the number of points. To overcome this limitation, we modify the octree

partitioning to use a breadth-first search strategy. This involves using a queue Q to store all the current leaf-nodes. At each iteration, the node at the front of the queue Q is extracted and checked whether it satisfies one of the above stopping conditions or not. If the nodes need further division, the non-empty sub-bins of its will be added to the rear of the queue Q and the number of non-empty bins is updated. This process will be iterated until this number reaches the desired number of bins K . By doing so, bins having similar size and the number of points will be divided a similar number of times. The details for our modified octree partition are shown in Algorithm 1.

Table 4.2 shows the comparison among the three strategies to select the scattered points for modeling the gamut mapping function g . Here, the same number of bins ($4096 = 16^3$) is used in all three sampling methods. As can be seen, the running time for uniform sampling is the smallest but its reconstructed errors are highest since the number of non-empty bins is relatively small (around 7%). Using k-means can obtain reasonable reconstructed results but the running time is significantly high. Our modified octree sampling obtains the best reconstructed results and is $10\times$ faster than k-means clustering to compute. It is worth noting that the octree partitioning may not guarantee the exact number of returned non-empty bins as one or more subbin(s) (up to eight) are created at each division. However, it can reach very close to the given number within 8 control points.

4.2.3 Metadata Embedding

After all above parameters are estimated, they are embedded into the metadata of the JPEG file. The metadata structure in JPEG contains several segments. Each

Algorithm 1 Modified Octree Partition

Input: a set of n color points $\{p_i\}_{i=1}^n$, the desired number of bins K , the minimum capacity δ_p , the minimum size of bin δ_l , and the maximum depth of bin δ_d .

```
1:  $B \leftarrow [n]$                                 ▶ Array of Point Counts
2:  $C \leftarrow [\min(p_i), \max(p_i)]$             ▶ Array of Bin Connors
3:  $D \leftarrow [0]$                                 ▶ Array of Bin Depths
4:  $M_i \leftarrow 1$                                 ▶ Map of Point Bin
5: create an empty queue  $Q$ 
6:  $Q.enqueue(1)$                                 ▶ Insert the first bin
7:  $t \leftarrow 1$                                 ▶ Total number of non-empty bins
8: while ( $t < K$ )  $\wedge \neg Q.isEmpty()$  do
9:    $u \leftarrow Q.dequeue()$ 
10:   $L_u \leftarrow \min(C_u(4 : 6) - C_u(1 : 3))$ 
11:   $scd \leftarrow (B_u < \delta_p) \vee (L_u < \delta_l) \vee (D_u > \delta_d)$ 
12:  if  $scd$  then
13:    continue                                ▶ Move to the next bin
14:  end if
15:   $m \leftarrow (C_u(1 : 3) + C_u(4 : 6))/2$         ▶ The center
16:   $t \leftarrow t - 1$                             ▶ Remove the old bin
17:  for  $i = 1 : 8$  do
18:     $v \leftarrow length(D) + 1$                 ▶ New bin number
19:     $D_v \leftarrow D_u + 1$                     ▶ Increase depth
20:     $C_v \leftarrow getConnors(m, C_u)$ 
21:    Calculate  $mask$  which points belong in  $v$ 
22:     $M_{mask} \leftarrow v$ 
23:     $B_v \leftarrow countPointNumber(M, v)$ 
24:    if  $B_v > 0$  then
25:       $t \leftarrow t + 1$                         ▶ Insert the new bin
26:       $Q.enqueue(v)$ 
27:    end if
28:  end for
29: end while
```

Output: a map of point bins M .

	Uniform	K-means	Octree
Begin	4096	4096	4096
Return	283	4096	4091
Time(s)	0.53	101.70	11.18
RMSE	0.0037	0.0026	0.0023

Table 4.2: This table shows the three different strategies to select the scattered points for modeling the gamut mapping. These are uniform partition, k-means clustering, and our octree partitioning.

segment has capacity for different kind of data. These are delimited by two-byte codes called markers [Hamilton 1992; Brower et al. 2011]. One of these segments is the comment (COM) segment. A COM segment does not interfere with the image stored in the JPEG file. The maximum size of a COM segment is 64 KB.

We used the COM segment to store RAW data. Since the COM segment can only contain a text string, we need to encode these parameters into an array of characters, however, we must avoid the special character “null” (i.e. 0x00) as it denotes the end of the text comment. To avoid the null character in the sequence of characters, we used a simple and fast scheme as follows. The character sequence is converted into a sequence of binary bits. At every seventh bits, an additional bit 1 is inserted. This new bit stream is then converted to ASCII characters. By inserting this additional bit in this periodic manner the COM segment will not contain the null character. Figure 4.5 shows an example of this approach. By inserting this additional bit, the real storage size of metadata in the COM segment reduces to 56 KB (7/8 of the original size 64 KB). This is why the data allocation shown in Table 4.1 is only 56 KB.

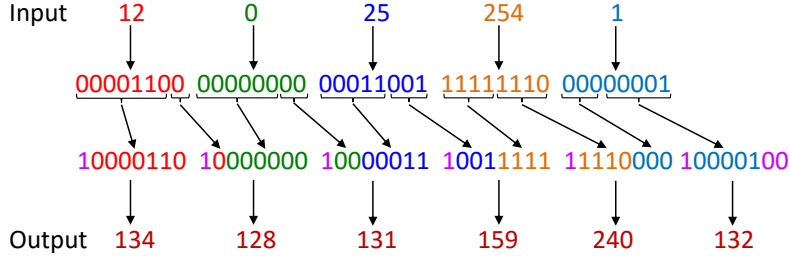


Figure 4.5: This figure shows an example of our encoding method which avoids null characters.

4.2.4 RAW Reconstruction

To reconstruct the RAW values, the metadata embedded in the JPEG file is first extracted and decoded by converting the text string to an bit string and then removing the additional bit pattern. We now have back all the parameterized data: the inverse white-balance matrix T_w^{-1} , the inverse color space transformation T_c^{-1} , the control points for the inverse gamut map g^{-1} and the control points for the inverse tone mapping f^{-1} . The RAW values are reconstructed by first applying the inverse tone-mapping f^{-1} to obtain the linearized sRGB image. Then the gamut mapping is applied. We adopt a linear tetrahedral interpolation [Kasson et al. 1993] to model the gamut mapping since the scattered points are in 3D color space. Next, the inverse color space transformation is applied and finally, the white-balance step is undone to obtain the reconstructed RAW image.

4.3 Experiments

In this section, we first compared our RAW reconstruction strategy with alternative techniques, including our method using k-means clustering, our method using the

octree, and the upsampling method by Yuan and Sun [Yuan and Sun 2011]. Yuan and Sun [Yuan and Sun 2011] do not provide code and we have re-implemented their approach. We do note that we have modified their approach to use the recent state-of-the-art method proposed by Ferstl et al. [Ferstl et al. 2013] to perform the image upsampling. Images used in this work were taken from various online data sets that contain RAW-sRGB pairs, including images from [Kim et al. 2012], [Cheng et al. 2014], [Nguyen et al. 2014a], and [Dang-Nguyen et al. 2015].

Figures 4.6 shows the results for images from various cameras and scenes. A jet map is used to show the error between the original and reconstructed RAW image. The root mean square error (RMSE) is also shown to quantitatively evaluate the results. For Yuan and Sun’s method [Yuan and Sun 2011], RAW images at resolutions of 1/2 of the original size are used for upsampling. For a fairer comparison, we also used a small-RAW image of resolution 100×90 which can be placed inside the 64 KB metadata. Table 4.3 shows the results in terms of RMSE. For each camera, 30 images were examined and the average RMSE was reported. The proposed octree partitioning approach provides RAW reconstructions with the lowest RMSE. Note that the error metric also includes quantization errors. This may seem counter intuitive, but the quantization does not introduce a huge problem. Since RAW values are quantized to 8-bit values (0 to 255), the different between quantization steps is 1 and the range of the image is 1 to 255. If we round the values during quantization, the maximum error will be a half of that (i.e. 0.5, or $0.5/255 \approx 0.2\%$). In general, the error will not be the max, but the average which is $0.25/255 \approx 0.001$, or 0.1%.

The proposed method does not attempt to remove compression artifacts that arise due to the lossy nature of JPEG. We assume that the input image is saved as

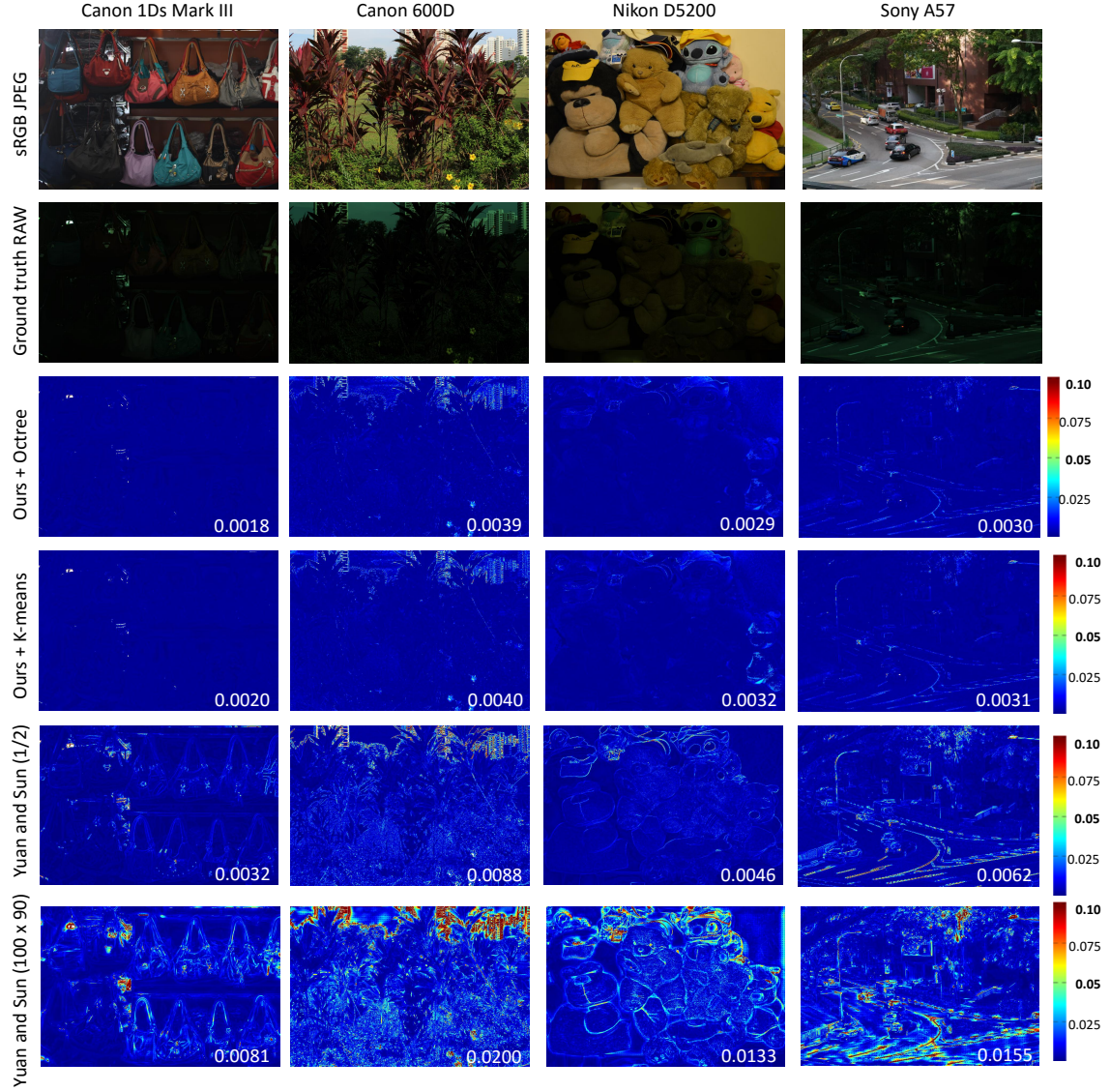


Figure 4.6: This figure shows comparisons between our approach and our implementation of the upsampling approach proposed by Yuan and Sun for various scenes and cameras (a Canon 1Ds Mark III, a Canon 600D, a Nikon D5200, and a Sony α 57). The white points on the difference maps indicate overexposed pixels with a value of 255 in any of the channels. The RMSEs for the each method are shown in the bottom right of each error map.

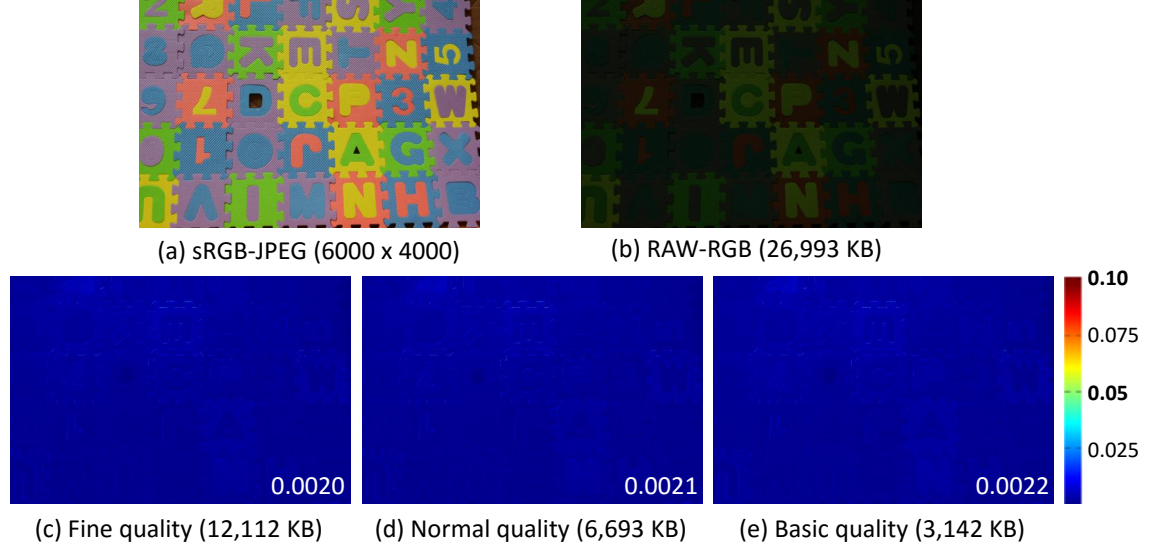


Figure 4.7: This figure shows an example of using different qualities of sRGB-JPEG images for reconstructing the RAW-RGB image. Here, three different qualities: fine, normal, and basic (which supports in Nikon cameras) are examined. The RMSEs for the each quality are shown in the bottom right of each error map.

Camera Name	Ours	[Yuan and Sun 2011] (1/2)	[Yuan and Sun 2011] (100 × 90)
Canon 1Ds	0.0018	0.0049	0.0135
Canon 600D	0.0038	0.0085	0.0191
Nikon D5200	0.0033	0.0078	0.0173
Sony α57	0.0020	0.0055	0.0150

Table 4.3: This table shows the comparison between our method and up-sampling method proposed by Yuan and Sun in terms of RMSE. For up-sampling method proposed by Yuan and Sun, RAW images at resolutions of 1/2 of the original size and 100 × 90 are used for upsampling.

a high-quality JPEG, however, for sake of completeness, we examine the effect of different JPEG qualities on the reconstructed RAW results. Most DSLR cameras support three image quality types, e.g.: fine, normal, and basic, that corresponds to the compression ratios (1/4, 1/8 and 1/16). In this experiment, we examined using different sRGB-JPEG quality images to reconstruct the RAW image. Figure 4.7 shows an example for a Nikon D5200 camera. Unsurprisingly, the quality does affect the reconstructed RAW-RGB images, however the overall difference between the different qualities is not significant.

4.4 Applications

We demonstrate the usefulness of the reconstructed RAW image data with two applications: white-balance and image deblurring. It is well known that having access to the original RAW image is advantageous for both these applications.

4.4.1 White-Balance Correction

As noted in the processing pipeline in Figure 4.2 in Section 4.2, white-balancing is a procedure that is applied early in the processing pipeline. Attempting to change white-balance in the sRGB image is challenging as it cannot undo the photo-finishing that has been applied to the sRGB image. In this experiment, we compared applying white-balance correction on our reconstructed RAW and the original sRGB-JPEG. The comparison results are shown in Figure 4.8. The input images are captured under wrong white-balance settings; while the ground truth images are captured under the proper settings. Here, the achromatic colors on the color checker boards are manually selected to use as the scene illumination color.

White-balance correction on the reconstructed RAW images is visually better than correction using the sRGB images.

4.4.2 Image Deblurring

Image deblurring assumes a linear image formation model in the form: $I_B = I \otimes h$, where I is the latent image, h is a blur kernel, and \otimes is a convolution operator. For blurred sRGB images, the relationship is not truly linear between I_B and I . Work by Tai et al [Tai et al. 2013] showed that the non-linear response of the camera changes the shape of the h to a spatially varying convolution making image deblurring even more challenging. Thus, it is desirable to deblur the image in the linear RAW space. We compared the deblurring method proposed in [Krishnan and Fergus 2009] on our reconstructed RAW and sRGB-JPEG images. This is done by applying a motion blur on a ground truth RAW image and then use the estimated parameters in the camera color pipeline to synthesize the blurred sRGB input images. Figure 4.9 shows the results of the deblurred sRGB and deblurred reconstructed RAW image. The signal-to-noise ratios (SNRs) are also reported at the bottom right of each image. Deblurring of the reconstructed RAW images gives superior results.

4.5 Discussion and Summary

We have described a method to encode the necessary metadata with the sRGB image for reconstructing a high-quality RAW image. Our approach produces a fully self-contained 100% compatible JPEG file that can be used as-is, not affecting any existing image workflows. This method can reconstruct the original RAW to within 0.3% error with only 64 KB overhead to the original JPEG file.

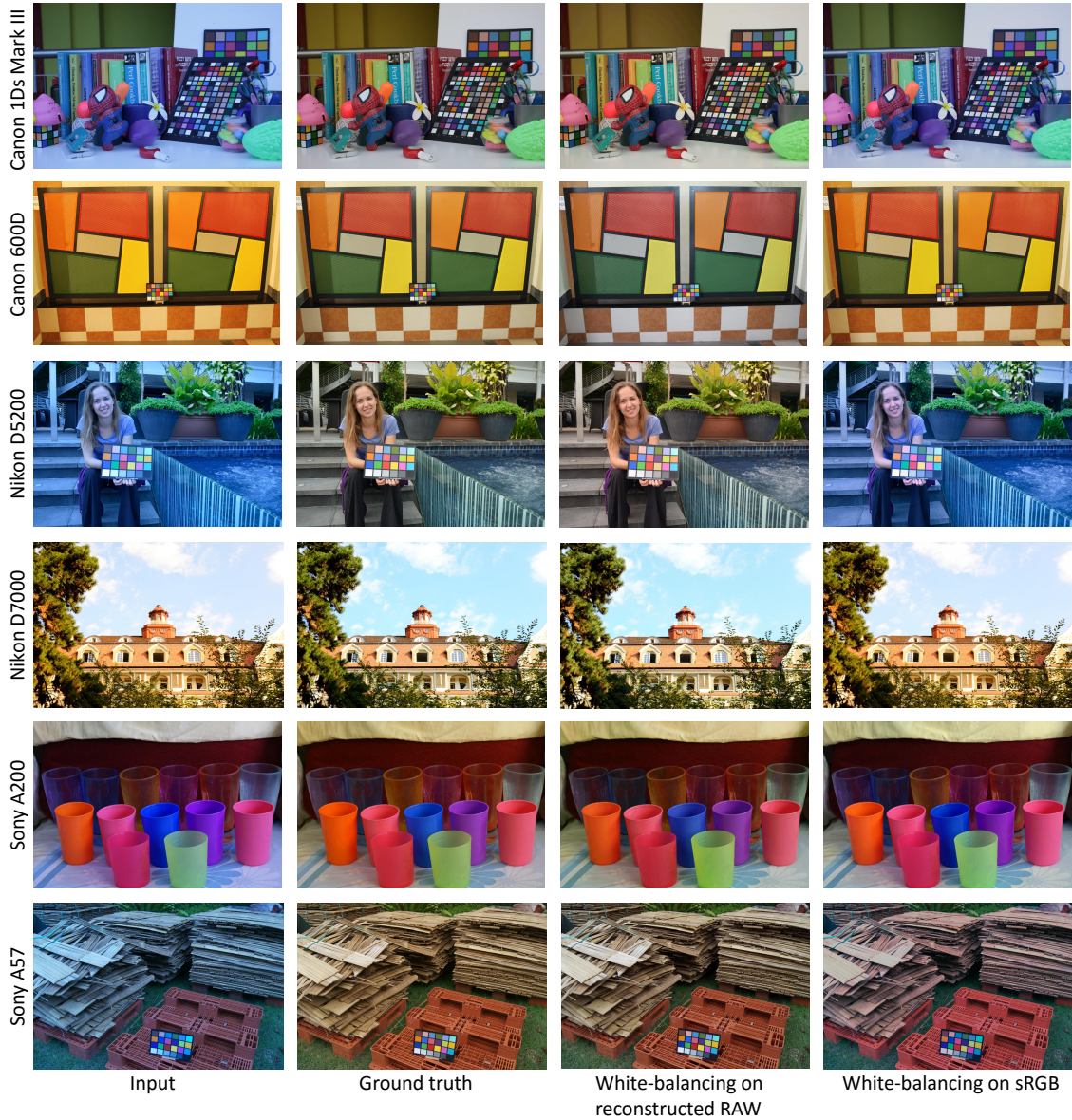


Figure 4.8: This figure shows examples on correcting white-balance for different cameras: a Canon 1Ds Mark III, a Canon 600D, a Nikon D5200, a Nikon D7000, a Sony α 200 and a Sony α 57. The first column is the input images captured under the wrong white-balance settings; the second column shows the ground truth images captured under the proper settings. The third column displays the results applied the white-balance correction on our reconstructed RAW images. The final column shows the results applied the white-balance correction directly on the sRGB-JPEG images.

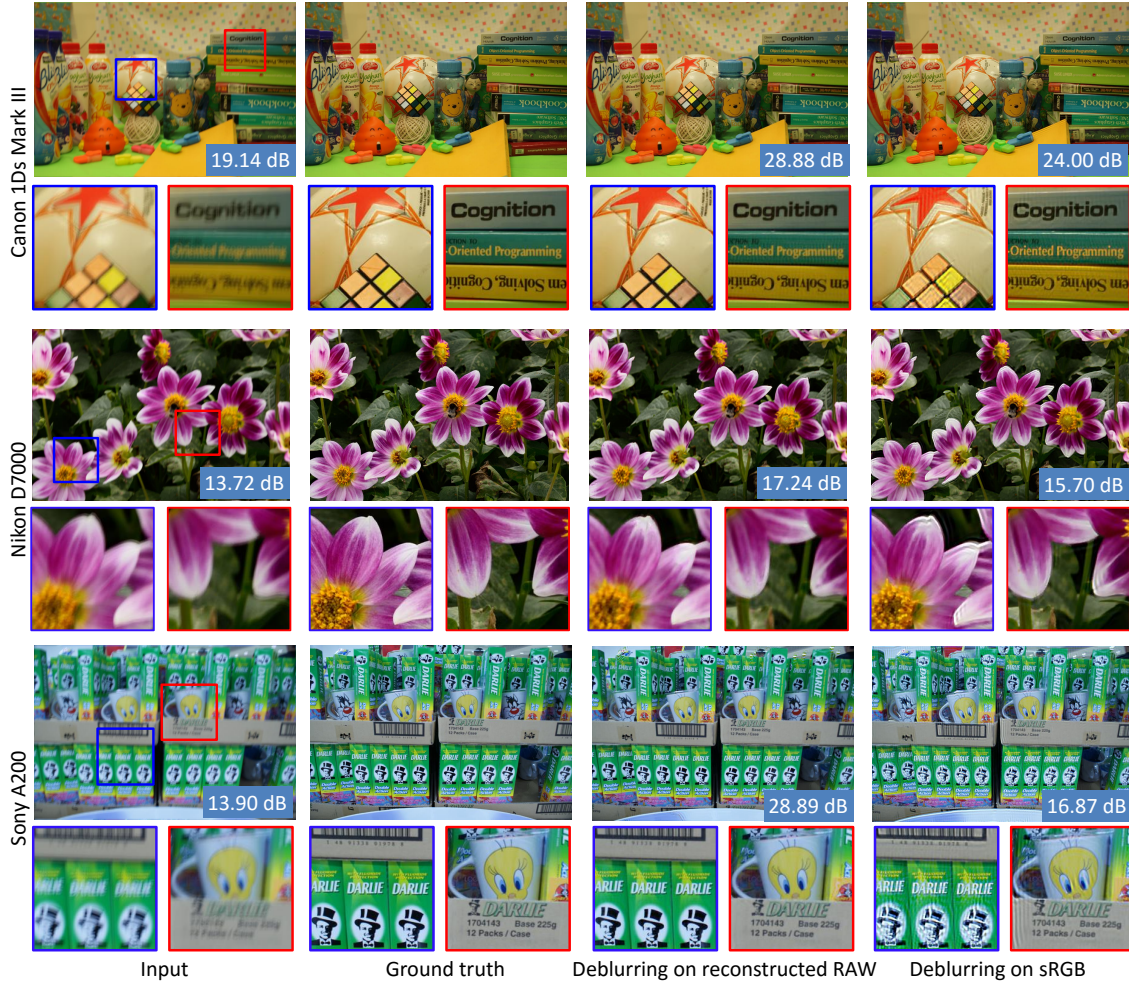


Figure 4.9: This figure shows examples for image deblurring for different cameras: a Canon 1Ds Mark III, a Nikon D7000, and a Sony α 200. A motion blur on the non-blurred ground truth RAW images is performed. The blurred sRGB image is synthesized using the parameterized color pipeline model. We applied our method to reconstruct the blurred RAW image, then deblurred it, and converted it back to the sRGB image. The first, third and fifth rows show the results, while the second, fourth and sixth rows show close-ups of particular regions. The signal-to-noise ratios (SNRs) were reported at the bottom right of each image.

One drawback of our method is that we cannot properly handle sRGB image values that are saturated or locally varying scene dependent. Upsampling methods such as Yuan and Sun [Yuan and Sun 2011] can better estimate these values given the availability of spatial RAW values in these regions. Future work would be to embed additional RAW values spatially to help in-paint or interpolate these regions. We also note that we have optimized our framework to minimize error for backward mapping from sRGB to RAW, however, for many photography tasks (such as our white-balance example), the forward mapping from RAW back to sRGB is needed. A topic worth further investigation would be to develop a method that considers the two-way reconstruction error.

Chapter 5

Color Transfer between a Pair of Images

In the two previous chapters, color mapping was investigated in a low level which was used for calibrating the raw-RGB spaces between different cameras, and in the mid-level which was used recovering the RAW image from the corresponding photo-finished sRGB image. In this chapter, we handle the color mapping at a higher level application. In particular, we present a new approach for color transfer between two images where the origins of the images and their corresponding image formation are not known. While this is a well-studied topic, our proposed 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

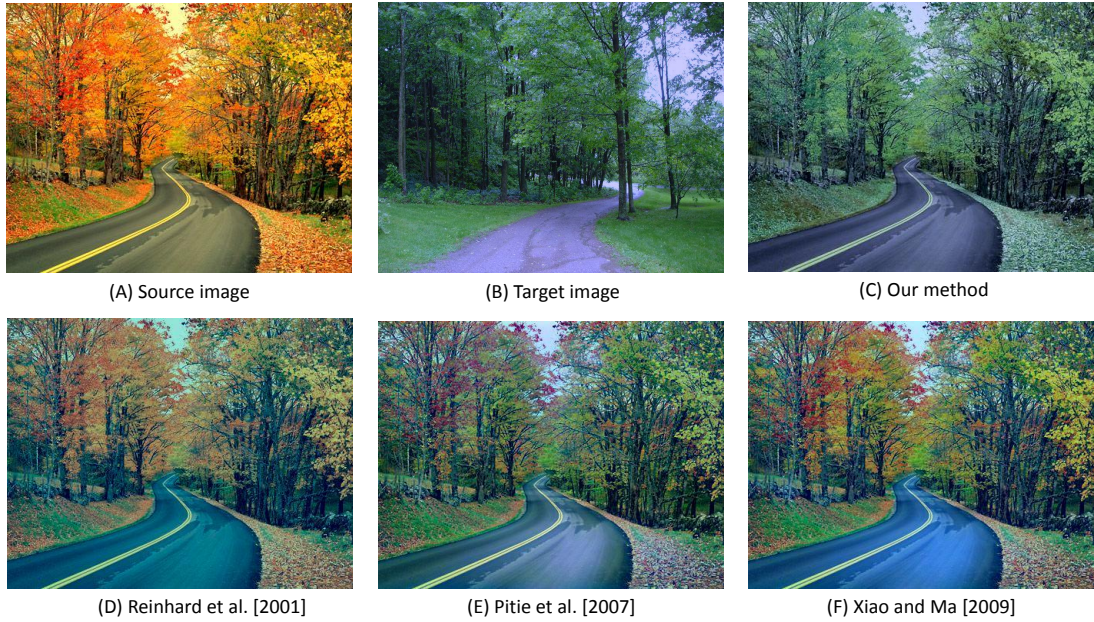


Figure 5.1: This figure compares color transfer results of several methods. Our method incorporates information about the source and target scene illuminants and constrains the color transfer to lie within the color gamut of the target image. Our resulting image has a more natural look and feel than existing methods.

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.

5.1 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. Figure 5.1 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. [Reinhard et al. 2001; Pitié et al. 2007; Xiao and Ma 2009]). Figure 5.1-(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 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 chapter, 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 Figure 5.1. 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 chapter is as follows: Section 5.2 overviews our approach and explains how we incorporate illumination estimation into the color mapping process and constrain the gamut; Section 5.3 shows experiments to evaluate our procedure and we conclude the chapter with a discussion and summary in Section 5.4.

5.2 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 . Figure 5.2 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 below.

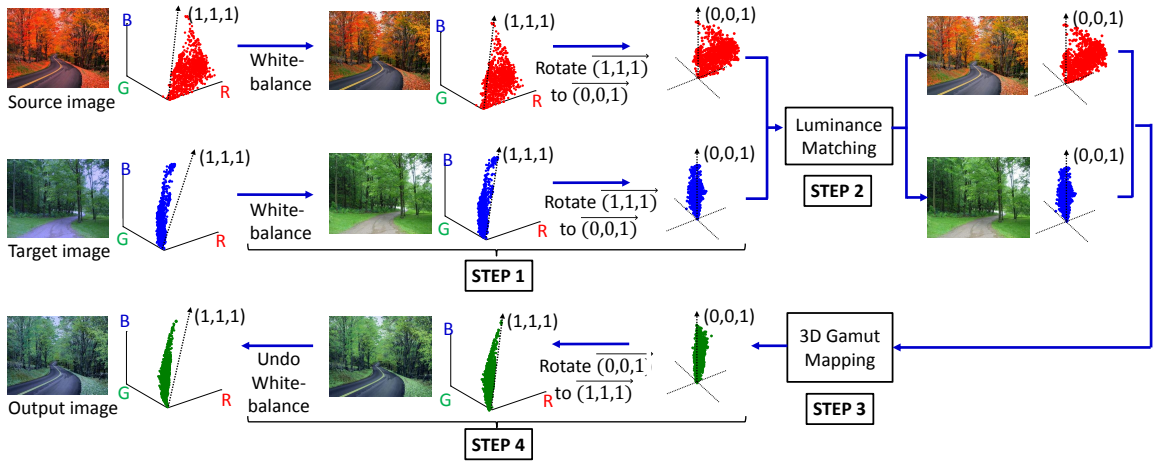


Figure 5.2: This figure shows our color transfer framework. Step 1: the “white” points of the source and target images are matched together using white-balancing. These are then rotated along the $(0,0,1)$ axis. Step 2: a gradient preserving technique is applied on the luminance channel (white-axis) 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).

5.2.1 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 image. 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 [Chakrabarti et al. 2012; Chong et al. 2007; Gijsenij et al. 2012]. In this work, we used the weighted Grey-Edge algorithm proposed in [Gijsenij et al. 2012] 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.

Figure 5.3 shows the importance of this step in removing the color cast and bias in the 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 (a pure diffuse white materials). 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 work were obtained using

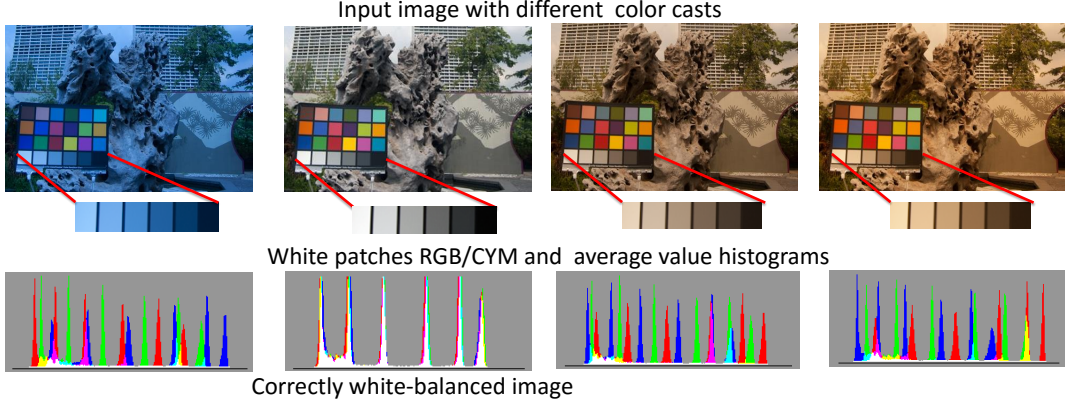


Figure 5.3: This figure shows the importance of proper white-balance in determining the proper scene luminance. A scene was captured with a color chart and white balanced with different settings. The achromatic patches on the color chart are extracted and their color channel histograms as well as overall 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.

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 matrix:

$$\mathbf{R} = \begin{pmatrix} \cos(\beta) & 0 & \sin(\beta) \\ 0 & 1 & 0 \\ -\sin(\beta) & 0 & \cos(\beta) \end{pmatrix} \begin{pmatrix} \cos(\alpha) & -\sin(\alpha) & 0 \\ \sin(\alpha) & \cos(\alpha) & 0 \\ 0 & 0 & 1 \end{pmatrix}, \quad (5.1)$$

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) chan-

nel and the other two axes represent chromaticity.

5.2.2 Matching brightness channel

The next step is to match the overall brightness between the two images. We only use the transformed luminance values for this step and we adopt Xiao et al.'s gradient preserving matching technique [Xiao and Ma 2009]. After this step, the output image would have the similar brightness histograms 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 Equation 5.2).

$$L_f = C_t^{-1}(C_s(L_s)), \quad (5.2)$$

where C_s and 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.

$$\left[I + \lambda(D_x^T D_x + D_y^T D_y) \right] L_o = L_f + \lambda(D_x^T D_x + D_y^T D_y) L_s, \quad (5.3)$$

where I is the identity matrix; D_x, D_y are two gradient matrices along to x , and y direction; λ is a regularization parameter. As shown by [Xiao and Ma 2009], the gradient regularization term helps to reduce halo artifacts that may arise in the histogram mapping.

5.2.3 Aligning the color gamut

To align the source color gamuts to the target resulting from previous step, the centers of gamuts of the source and the target images 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:

$$\begin{aligned} I_s &= I_s - \mu_s, \\ I_t &= I_t - \mu_t. \end{aligned} \tag{5.4}$$

The gamut mapping process is approximated by a linear transformation \mathbf{T} that includes a scale and a rotation (defined in Equation 5.5), which is used to align the gamut of the source image to that of the target image (Figure 5.4).

$$T = \begin{Bmatrix} s_1 \cos(\theta) & -s_1 \sin(\theta) & 0 \\ s_2 \sin(\theta) & s_2 \cos(\theta) & 0 \\ 0 & 0 & 1 \end{Bmatrix}, \tag{5.5}$$

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 T , we minimize for the following cost function:

$$f(T) = 2\mathcal{V}((T \times CH_s) \oplus CH_t) - \mathcal{V}(CH_t) - \mathcal{V}(T \times CH_s), \tag{5.6}$$

where CH_s , and CH_t are the full 3D convex hulls of the source and target image

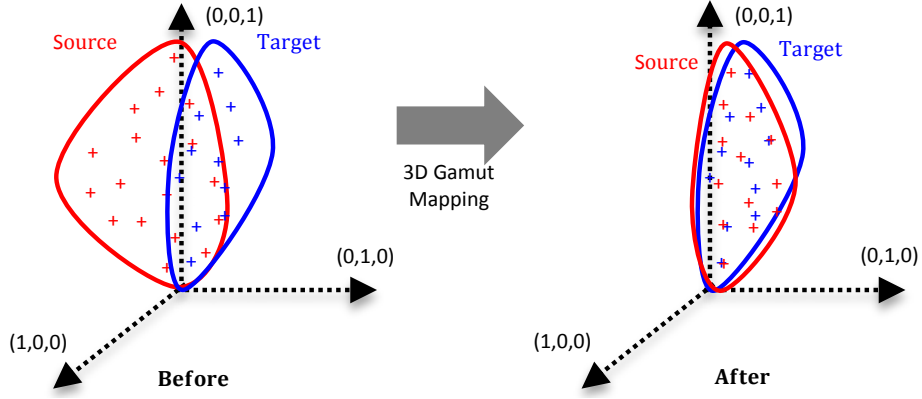


Figure 5.4: Our gamut mapping step to align the color distributions between two images.

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 Equation 5.6 is a non-convex function, a brute-force search is required to find the global optimum. We obtain an approximated 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:

$$I_o = TI_s + \mu_t. \quad (5.7)$$

5.2.4 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 \mathbf{R} defined in Equation 5.1 in the first step. The final step is to undo the white balancing by multiplying the point colors with the previously computed white point of the target image (r_{wt}, g_{wt}, b_{wt}) .

5.3 Experiments

5.3.1 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:

$$D(I_t, I_o) = (\mathcal{V}(CH_c) - \mathcal{V}(CH_t)) + (\mathcal{V}(CH_c) - \mathcal{V}(CH_o)), \quad (5.8)$$

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 Equation 5.8 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 our metric. However, the metric does reveal to which extent other methods produce results that are out of gamut with the target image.

5.3.2 Results

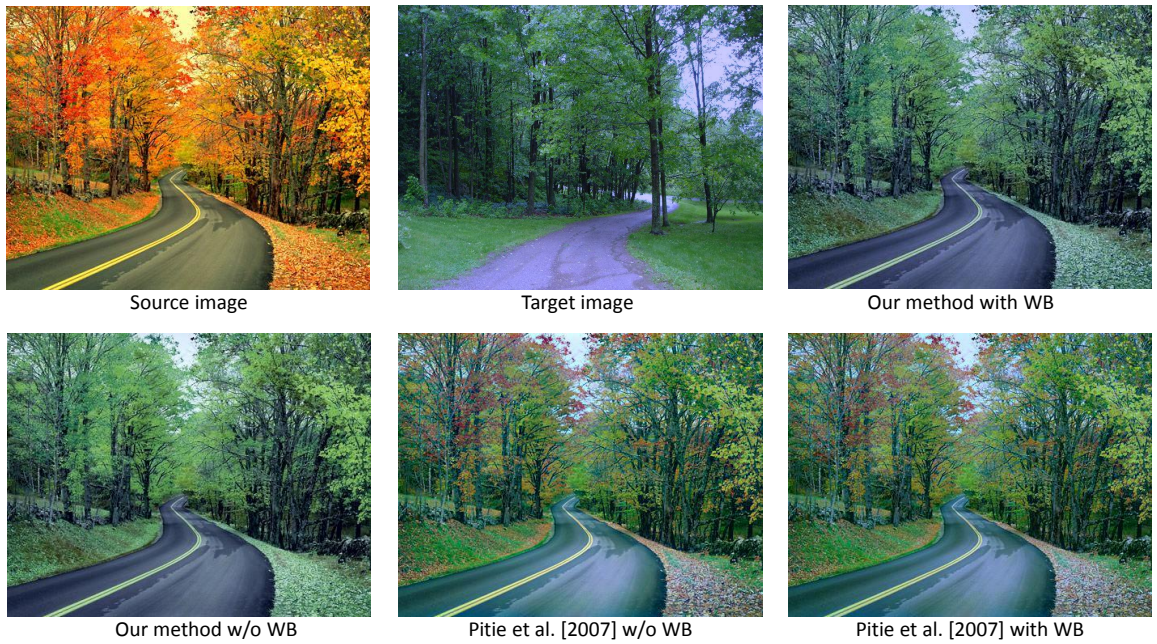


Figure 5.5: This figure shows the contribution of the gamut mapping and white-balancing in our framework. It is clear seen that the gamut mapping step help our method reduce out-of-gamut colors in comparison with the results from Petie et al.’s method. While the white-balancing step make the color cast of the output image close to that of the target image.

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 [Pitié et al. 2007] method by adding white-balancing as the first step. We use again the source and target images in Fig-

ure 5.1 for comparison. The results of all these approach are shown in Figure 5.5. It can be clearly seen that our method with the white-balancing step gives a better look and feel 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: [Reinhard et al. 2001], [Pitié et al. 2007], and [Xiao and Ma 2009]. The qualitative comparisons are shown in Figure 5.1, 5.6, 5.7, and 5.8. Figure 5.1 shows Example 1, while Figs. 5.6, 5.7, and 5.8 show Examples 2-10.

It can be observed that 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 algorithm proposed in [Gijsenij et al. 2012]. However, in a few cases like Examples 1 and 7, this algorithm did not perform satisfactorily. Therefore, these images required 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-gamut color than the other methods. Moreover, our approach arguably produces images with a closer look and feel to the target image than the other methods.

We also provide quantitative evaluation using the metric developed in Equa-

Example	Ours	[Reinhard et al. 2001]	[Pitié et al. 2007]	[Xiao and Ma 2009]
#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 5.1: The table shows the comparisons between all methods in terms of the difference between target and output gamut. The images for these examples are shown in Figs. 5.1, 5.6, 5.7, and 5.8.

tion 5.8. Table 5.1 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 dual core 3.10 GHz PC with 16.0 GB RAM. The comparison of execution time between all methods is shown in Table 5.2. 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 limitations. It can fail when the source and target images have complicated color distributions like in Figure 5.9. 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 Figure 5.9). Performing color transfer in a local manner may solve this case.

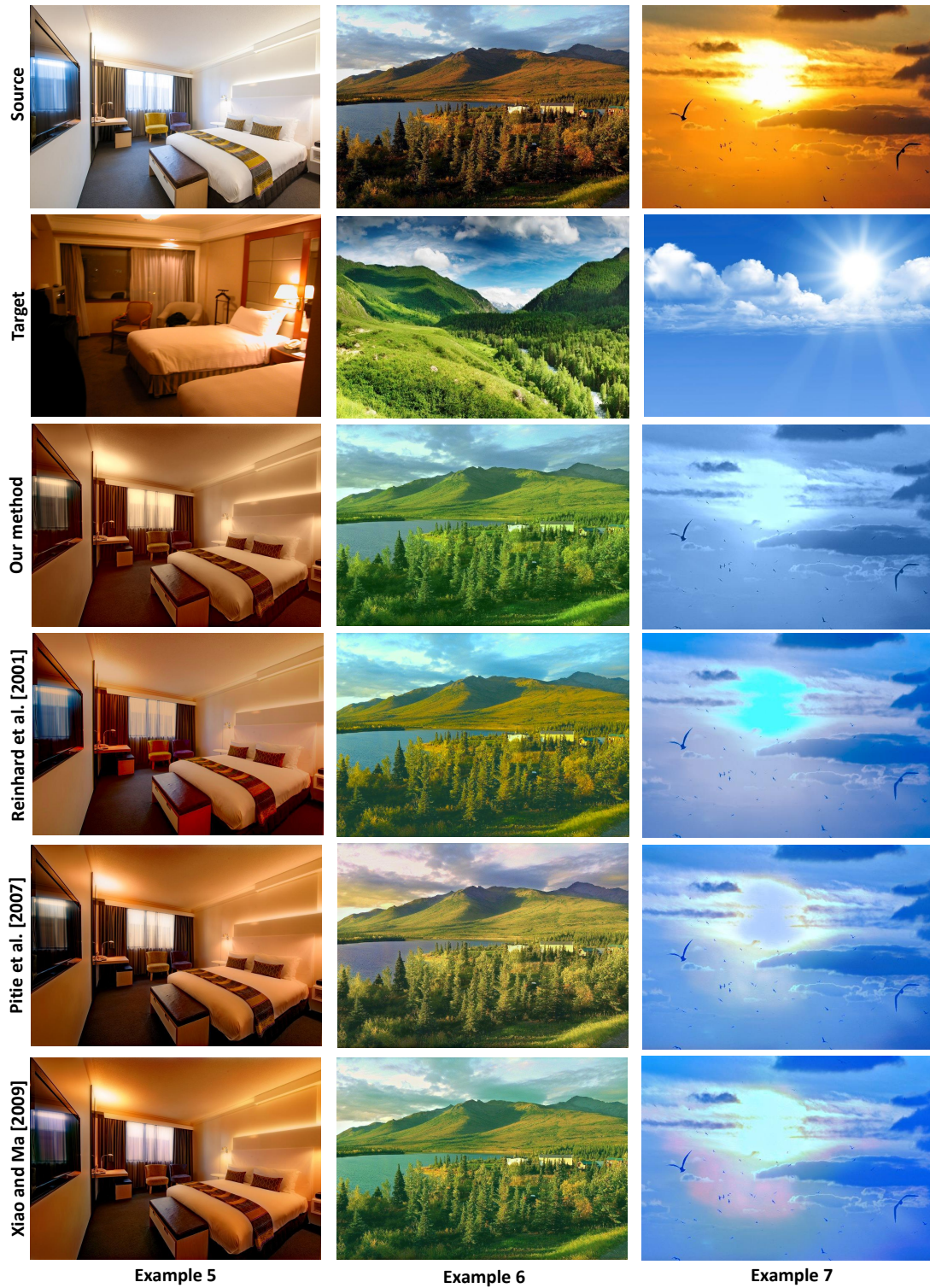


Figure 5.7: This figure shows Examples 5, 6, and 7 for comparisons between all methods.

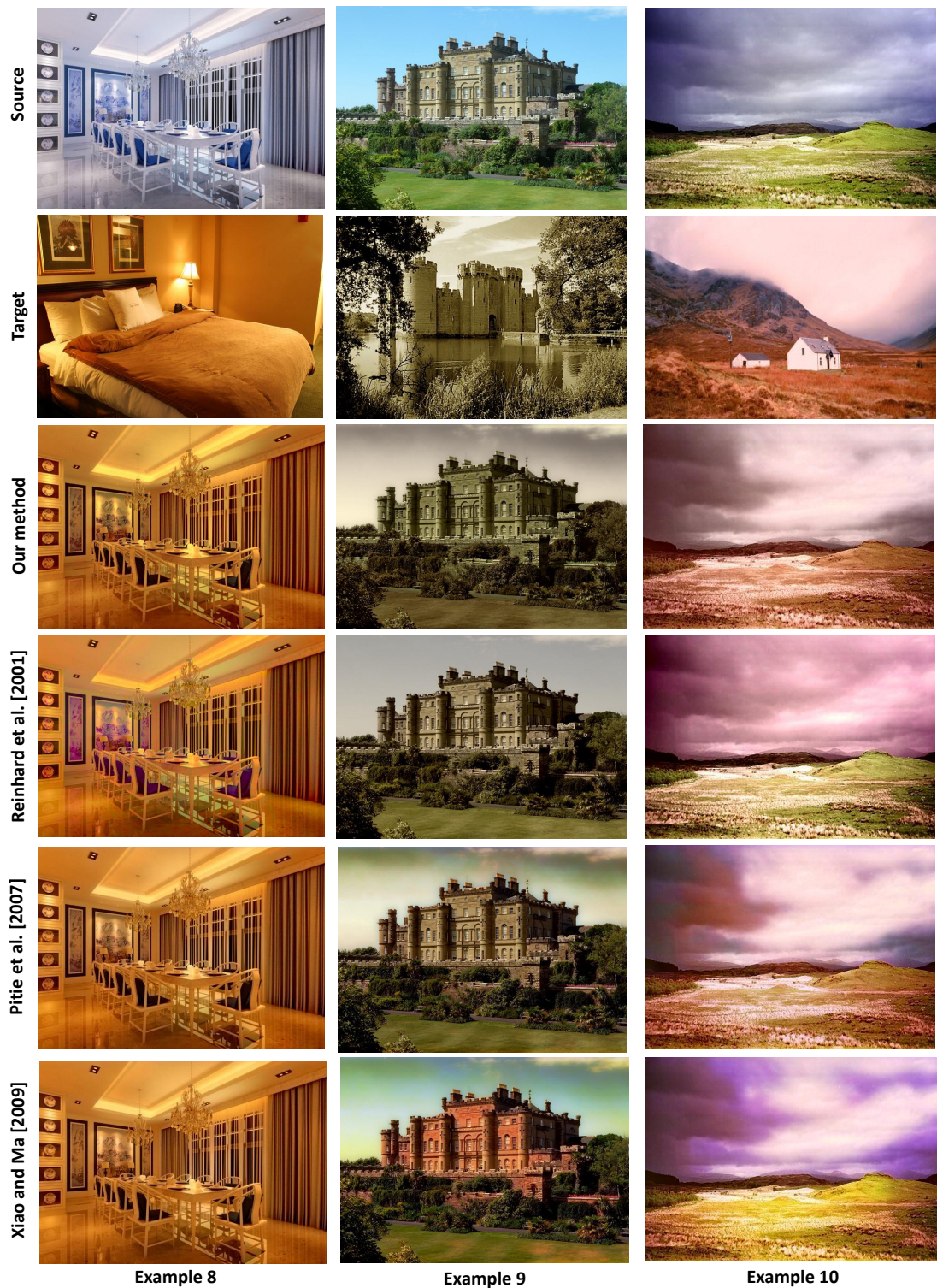


Figure 5.8: This figure shows Examples 8, 9, and 10 for comparisons between all methods.

Method	Relative performance
Ours	1.000
[Reinhard et al. 2001]	0.047
[Pitié et al. 2007]	2.193
[Xiao and Ma 2009]	1.823

Table 5.2: The 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 5.9: 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.

5.4 Discussion and Summary

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

will already lie along the $(0,0,0)$ to $(1,1,1)$ line in the sRGB colorspace 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.

Chapter 6

Conclusion and Future Directions

Although each previously mentioned chapter had a self-contained summary and discussion, this chapter concludes the thesis by giving a collective summary of the works described in previous chapters. We also describe several potential directions for future work.

6.1 Overall Summary

The goal of this thesis has been to address the problem of color mapping for camera color space calibration and color transfer between images. Chapter 2 provides the necessary background knowledge for this task as well as providing an overview of works related to this topic.

Chapter 3 described a state-of-the-art approach on estimating a mapping that can convert a RAW image of an arbitrary scene and illumination from one camera's RAW space to another. We introduced an illumination-independent mapping approach that can convert a RAW image of an arbitrary scene and illumination

from one camera’s RAW space to another. In stead of maintaining multiple transformations and categorizing input images to scene illumination, we proposed an illumination-independent method that uses white-balancing to canonicalize the illumination. This method allows us to perform the RAW-to-RAW mapping using only two linear transformations.

Chapter 4 described a method to encode the necessary metadata with the sRGB image for reconstructing a high-quality RAW image. Our approach produces a self-contained completely compatible JPEG file that can be used as-is, not affecting any existing image workflows. This method can reconstruct the original RAW to within a small amount of error with only 64 KB overhead to the original JPEG file.

Chapter 5 proposed a novel color transfer method that transfers color styles between a pair of images. We proposed a novel method for color transfer that considers the scene illumination in the color transfer process and constrains the result to fit within the target image’s gamut. As a result, the source illumination will be matched to the target illumination that can reduce the appearance difference between the source and target images. Moreover, we also explicitly enforced the color transform to produce a result that lies within the target image’s color gamut. Therefore, out-of-gamut colors, which give a strange appearance to the target image, can be avoided.

In summary, the thesis has examined the color mapping in three levels: low level where the mapping is between two RAW images for standardizing RAW images; middle level where the mapping is between RAW and photo-finished sRGB images for recovering RAW, and high level where the mapping is between two sRGB images for sharing the same “look and feel”.

6.2 Future directions

There are two potential future research directions aligned the works presented in this thesis. They are summarized in the following:

6.2.1 Harmonizing a group of images

Given a task such as designing a brochure, an designer may have a color theme in mind related to the graphics of the brochure. The designer would like to insert a number of images to make it more attractive, however, existing images often have different scene contents and also may have been taken under different lighting conditions. This results in images being incorporated into a brochure that results in visually noticeable color inconsistencies. Even though the designer spends time to find appropriate images, they may not go well or match the color theme of the brochure (an example as shown in Figure 6.1). This problem is related to the color transfer problem [Reinhard et al. 2001] that has recently drawn a large amount of research attention [Faridul et al. 2014]. However, most of these existing works handle a pair of images only which is difficult to extend to a group of images with an additional color theme constraint. Therefore, one promising direction is to investigate on the problem of harmonizing a group of images among themselves with an additional color theme from a given brochure theme.

6.2.2 Two-way reconstruction between RAW and sRGB

As mentioned at the end of Chapter 4, our method for reconstructing RAW from sRGB only considers backward mapping from sRGB to RAW since our goal is just

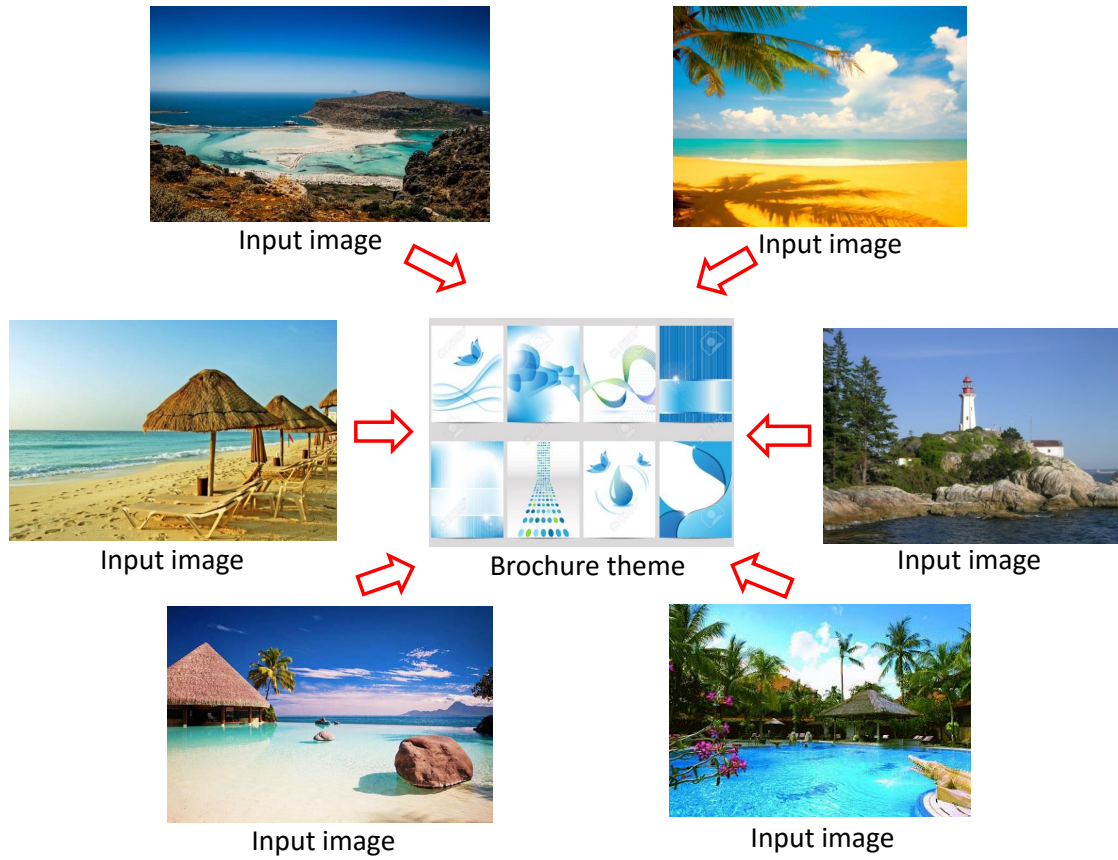


Figure 6.1: The figure shows an example of a group of input images for designing a brochure.

to obtain reconstructed RAW with as minimal error as possible. However, for many photography tasks (such as white-balance, deblurring), RAW images after being modified usually need to be converted back to sRGB for using in other applications. In these cases, the forward mapping from RAW back to sRGB is needed. Using the inversion of the backward mapping for the forward way can produce high error. Therefore, a topic worth further investigation would be to develop a method that considers the two-way reconstruction error.

Bibliography

- ABED, F. M., AMIRSHAHI, S. H., AND ABED, M. R. M. 2009. Reconstruction of reflectance data using an interpolation technique. *Journal of the Optical Society of America A (JOSA A)*. 23
- ADOBE. Digital negative (dng) specification. http://www.adobe.com/support/downloads/dng/dng_sdk.html. 22, 42
- AGAHIAN, F., AMIRSHAHI, S. A., AND AMIRSHAHI, S. H. 2008. Reconstruction of reflectance spectra using weighted principal component analysis. *Color Research & Application*. 23
- AN, X., AND PELLACINI, F. 2010. User-controllable color transfer. *Computer Graphics Forum (Proceedings of Eurographics 2010)*. 27
- ANDERSON, M., MOTTA, R., CHANDRASEKAR, S., AND STOKES, M. 1996. Proposal for a standard default color space for the internet - srgb. In *Color and Imaging Conference*. 18
- BARNARD, K., MARTIN, L., COATH, A., AND FUNT, B. 2002. A comparison of computational color constancy algorithms. ii. experiments with image data. *Transactions on Image Processing(TIP)*. 28

- BAY, H., TUYTELAARS, T., AND VAN GOOL, L. 2006. Surf: Speeded up robust features. In *European Conference on Computer Vision (ECCV)*. 25
- BRAINARD, D. H., AND WANDELL, B. A. 1986. Analysis of the retinex theory of color vision. *JOSA A* 3, 10, 1651–1661. 28
- BROWER, B., HINDS, A., HAMILTON, E., LEE, D., AND SULLIVAN, G. 2011. Information Technology: Digital compression and coding of continuous-tone still images: JPEG File Interchange Format (JFIF). *ISO/IEC*. 65
- BUCHSBAUM, G. 1980. A spatial processor model for object colour perception. *Journal of The Franklin Institute*. 28, 33
- BUHMANN, M. D. 2003. *Radial basis functions: theory and implementations*. Cambridge University Press. 35
- CHAKRABARTI, A., SCHARSTEIN, D., AND ZICKLER, T. 2009. An empirical camera model for internet color vision. In *British Machine Vision Conference (BMVC)*. 2, 3, 5, 23, 30
- CHAKRABARTI, A., HIRAKAWA, K., AND ZICKLER, T. 2012. Color constancy with spatio-spectral statistics. *Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*. 33, 41, 80
- CHAKRABARTI, A., XIONG, Y., SUN, B., DARRELL, T., SCHARSTEIN, D., ZICKLER, T., AND SAENKO, K. 2014. Modeling radiometric uncertainty for vision with tone-mapped color images. *Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*. 5, 20, 24, 25, 56, 57

- CHENG, D., PRASAD, D. K., AND BROWN, M. S. 2014. Illuminant estimation for color constancy: why spatial-domain methods work and the role of the color distribution. *Journal of the Optical Society of America A (JOSA A)*. 33, 67
- CHONG, H. Y., GORTLER, S. J., AND ZICKLER, T. 2007. The von kries hypothesis and a basis for color constancy. In *International Conference on Computer Vision (ICCV)*. 33, 41, 80
- COFFIN, D., 1997. DCRAW: decoding raw digital photos in linux. <https://www.cybercom.net/~dcoffin/dcraw/>. Accessed: 2015-10-05. 58
- COHEN-OR, D., SORKINE, O., GAL, R., LEYVAND, T., AND XU, Y.-Q. 2006. Color harmonization. In *ACM Transactions on Graphics*. 27
- COMANICIU, D., AND MEER, P. 2002. Mean shift: A robust approach toward feature space analysis. *Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*. 26
- DAI, S., HAN, M., XU, W., WU, Y., AND GONG, Y. 2007. Soft edge smoothness prior for alpha channel super resolution. In *Computer Vision and Pattern Recognition (CVPR)*. 24
- DANG-NGUYEN, D.-T., PASQUINI, C., CONOTTER, V., AND BOATO, G. 2015. Raise: A raw images dataset for digital image forensics. In *ACM Multimedia Systems Conference*. 67
- DAW, N. W. 1967. Goldfish retina: organization for simultaneous color contrast. *Science* 158, 3803, 942–944. 27

- DEBEVEC, P. E., AND MALIK, J. 1997. Recovering high dynamic range radiance maps from photographs. In *ACM Transactions on Graphics (Proceedings of ACM SIGGRAPH)*. 2, 21, 30
- DEBEVEC, P. E., AND MALIK, J. 2008. Recovering high dynamic range radiance maps from photographs. In *ACM SIGGRAPH 2008 classes*. 23, 56
- DIAZ, M., AND STURM, P. 2011. Radiometric calibration using photo collections. In *International Conference on Computational Photography (ICCP)*. 2, 21, 30
- FAIRCHILD, M. D. 2013. *Color appearance models*. John Wiley & Sons. vi, 12, 13
- FAIRMAN, H. S., BRILL, M. H., AND HEMMENDINGER, H. 1997. How the CIE 1931 color-matching functions were derived from Wright-Guild data. *Color Research & Application*. 16
- FARIDUL, H. S., POULI, T., CHAMARET, C., STAUDER, J., TRÉMEAU, A., REINHARD, E., ET AL. 2014. A survey of color mapping and its applications. In *Eurographics-State of the Art Reports*. 25, 96
- FATTAL, R. 2007. Image upsampling via imposed edge statistics. *ACM Transactions on Graphics (TOG)*. 24
- FERSTL, D., REINBACHER, C., RANFTL, R., RUETHER, M., AND BISCHOF, H. 2013. Image guided depth upsampling using anisotropic total generalized variation. In *International Conference on Computer Vision (ICCV)*. 67
- FINLAYSON, G. D., AND DREW, M. S. 1997. White-point preserving color correction. In *Color and Imaging Conference*. 5, 21

- FINLAYSON, G. D., AND SCHAEFER, G. 2001. Solving for colour constancy using a constrained dichromatic reflection model. *International Journal of Computer Vision (IJCV)*. 33
- FINLAYSON, G. D., AND TREZZI, E. 2004. Shades of gray and colour constancy. In *Color and Imaging Conference*, vol. 2004, Society for Imaging Science and Technology, 37–41. 28
- FISCHLER, M. A., AND BOLLES, R. C. 1981. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*. 25
- FORSYTH, D. A. 1990. A novel algorithm for color constancy. *International Journal of Computer Vision (IJCV)*. 33
- FREEMAN, W. T., PASZTOR, E. C., AND CARMICHAEL, O. T. 2000. Learning low-level vision. *International Journal of Computer Vision (IJCV)*. 24
- FUNT, B., AND BASTANI, P. 2014. Irradiance-independent camera color calibration. *Color Research & Application*. 5, 21
- FUNT, B., GHAFFARI, R., AND BASTANI, B. 2004. Optimal linear rgb-to-xyz mapping for color display calibration. In *Color and Imaging Conference*. 21, 22
- GIJSENIJ, A., GEVERS, T., AND VAN DE WEIJER, J. 2011. Computational color constancy: Survey and experiments. *Transactions on Image Processing (TIP)*. 28, 33, 41
- GIJSENIJ, A., GEVERS, T., AND VAN DE WEIJER, J. 2012. Improving color constancy by photometric edge weighting. *Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*. 80, 87

- GLASNER, D., BAGON, S., AND IRANI, M. 2009. Super-resolution from a single image. In *International Conference on Computer Vision (ICCV)*. 24
- GROSSBERG, M., AND NAYAR, S. 2003. What is the Space of Camera Response Functions? In *Computer Vision and Pattern Recognition (CVPR)*. 20
- GROSSBERG, M. D., AND NAYAR, S. K. 2003. Determining the camera response from images: What is knowable? *Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*. 23, 56
- GUILD, J. 1932. The colorimetric properties of the spectrum. *Philosophical Transactions of the Royal Society of London*. 14
- HACOHEN, Y., SHECHTMAN, E., GOLDMAN, D. B., AND LISCHINSKI, D. 2011. Non-rigid dense correspondence with applications for image enhancement. *ACM Transactions on Graphics (Proc. SIGGRAPH)*. 6
- HAMILTON, E. 1992. JPEG File Interchange Format. *C-Cube Microsystems*. 65
- HONG, G., LUO, M. R., AND RHODES, P. A. 2001. A study of digital camera colorimetric characterization based on polynomial modeling. *Color Res. & Application*. 5, 22, 35, 42, 49
- HOU, H. S., AND ANDREWS, H. 1978. Cubic splines for image interpolation and digital filtering. *Transactions on Acoustics, Speech and Signal Processing*. 24
- HUNG, P.-C. 1993. Colorimetric calibration in electronic imaging devices using a look-up-table model and interpolations. *Journal of Electronic imaging*. 5, 22

- HURLBERT, A. C., AND WOLF, C. J. 2002. Contribution of local and global cone-contrasts to color appearance: a retinex-like model. In *Electronic Imaging 2002*, International Society for Optics and Photonics, 286–297. 28
- HWANG, Y. B., LEE, J. Y., KWON, I. S., AND KIM, S. J. 2014. Color transfer using probabilistic moving least squares. In *Computer Vision and Pattern Recognition (CVPR)*. 6
- JIANG, J., LIU, D., GU, J., AND SUSSTRUNK, S. 2013. What is the space of spectral sensitivity functions for digital color cameras? In *Workshop on Applications of Computer Vision (WACV)*. 22
- KANAMORI, K., KAWAKAMI, H., AND KOTERA, H. 1990. Novel color transformation algorithm and its applications. In *Electronic Imaging*, 272–281. 5, 22
- KASSON, J. M., PLOUFFE, W., AND NIN, S. I., 1993. Tetrahedral interpolation technique for color space conversion. 66
- KIM, S. J., AND POLLEFEYS, M. 2008. Robust radiometric calibration and vignetting correction. *Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*. 20
- KIM, S. J., LIN, H. T., LU, Z., SÜSSTRUNK, S., LIN, S., AND BROWN, M. S. 2012. A new in-camera imaging model for color computer vision and its application. *Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*. 2, 3, 5, 20, 21, 23, 24, 25, 30, 56, 57, 60, 61, 67
- KRISHNAN, D., AND FERGUS, R. 2009. Fast image deconvolution using hyper-laplacian priors. In *Advances in Neural Information Processing Systems*. 71

- KULLBACK, S., AND LEIBLER, R. A. 1951. On information and sufficiency. *The Annals of Mathematical Statistics*. 49
- LAND, E. H., AND McCANN, J. J. 1971. Lightness and retinex theory. *JOSA* 61, 1, 1–11. 27
- LEVIN, A., LISCHINSKI, D., AND WEISS, Y. 2004. Colorization using optimization. In *ACM Transactions on Graphics*. 27
- LIN, S., AND ZHANG, L. 2005. Determining the radiometric response function from a single grayscale image. In *Computer Vision and Pattern Recognition (CVPR)*. 3, 20, 23
- LIN, S., GU, J., YAMAZAKI, S., AND SHUM, H.-Y. 2004. Radiometric calibration from a single image. In *Computer Vision and Pattern Recognition (CVPR)*. 3, 20
- LIN, H., KIM, S. J., SUSSTRUNK, S., AND BROWN, M. S. 2011. Revisiting radiometric calibration for color computer vision. In *International Conference on Computer Vision (ICCV)*. 20, 56
- LIN, H. T., LU, Z., KIM, S. J., AND BROWN, M. S. 2012. Nonuniform lattice regression for modeling the camera imaging pipeline. In *European Conference on Computer Vision (ECCV)*. 56, 61
- LOWE, D. G. 2004. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision (IJCV)*. 25
- MANN, PICARD, MANN, S., AND PICARD, R. W. 1995. On being ‘undigital’ with digital cameras: Extending dynamic range by combining differently exposed pictures. In *Proceedings of IS&T*. 23, 56

- MARTINEZ-VERDU, F., PUJOL, J., AND CAPILLA, P. 2003. Characterization of a digital camera as an absolute tristimulus colorimeter. *Journal of Imaging Science and Technology*. 5, 21, 22
- MEAGHER, D. J. 1980. Octree encoding: A new technique for the representation, manipulation and display of arbitrary 3-d objects by computer. Tech. rep., Image Processing Laboratory, Rensselaer Polytechnic Institute. 62
- MITSUNAGA, T., AND NAYAR, S. K. 1999. Radiometric self calibration. In *Computer Vision and Pattern Recognition (CVPR)*. 3, 23, 56
- NGUYEN, R. M. H., AND BROWN, M. S. 2016. Raw image reconstruction using a self-contained srgb-jpeg image with only 64 kb overhead. In *Computer Vision and Pattern Recognition (CVPR)*. 8
- NGUYEN, R. M. H., PRASAD, D. K., AND BROWN, M. S. 2014. Raw-to-raw: Mapping between image sensor color responses. In *Computer Vision and Pattern Recognition (CVPR)*. 8, 67
- NGUYEN, R. M., PRASAD, D. K., AND BROWN, M. S. 2014. Training-based spectral reconstruction from a single rgb image. In *European Conference on Computer Vision (ECCV)*. 23
- NGUYEN, R., KIM, S., AND BROWN, M. 2014. Illuminant aware gamut-based color transfer. In *Computer Graphics Forum*. 9
- OLIVEIRA, M., SAPPA, A. D., AND SANTOS, V. 2011. Unsupervised local color correction for coarsely registered images. In *Computer Vision and Pattern Recognition (CVPR)*. 6

- PAL, C., SZELISKI, R., UYTENDAELE, M., AND JOJIC, N. 2004. Probability models for high dynamic range imaging. In *Computer Vision and Pattern Recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on*, vol. 2, IEEE, II-173. 3
- PITIE, F., KOKARAM, A. C., AND DAHYOT, R. 2005. N-dimensional probability density function transfer and its application to color transfer. In *International Conference on Computer Vision (ICCV)*. 27
- PITIÉ, F., KOKARAM, A. C., AND DAHYOT, R. 2007. Automated colour grading using colour distribution transfer. *Computer Vision and Image Understanding*. 6, 27, 77, 86, 87, 88, 92
- PORIKLI, F. 2003. Inter-camera color calibration by correlation model function. In *International Conference on Image Processing (ICIP)*. 22
- POULI, T., AND REINHARD, E. 2011. Progressive color transfer for images of arbitrary dynamic range. *Computers & Graphics*. 6, 27
- PRASAD, D. K., NGUYEN, R., AND BROWN, M. S. 2013. Quick approximation of cameras spectral response from casual lighting. In *International Conference on Computer Vision Workshops (ICCVW)*. 22
- RASMUSSEN, C. E. 2006. Gaussian processes for machine learning. 36
- REINHARD, E., AND POULI, T. 2011. Colour spaces for colour transfer. In *Computational Color Imaging*. 1-15. 27

- REINHARD, E., ADHIKMIN, M., GOOCH, B., AND SHIRLEY, P. 2001. Color transfer between images. *IEEE Computer Graphics and Applications*. 4, 6, 25, 27, 77, 87, 88, 92, 96
- REINSCH, C. H. 1967. Smoothing by spline functions. *Numerische Mathematik*. 59
- SAUVAGET, C., VITTAUT, J.-N., SUAREZ, J., BOYER, V., AND MANUEL, S. 2010. Automated colorization of segmented images based on color harmony. *Journal of Materials Processing Technology*. 27
- SHAO, F., JIANG, G., YU, M., AND CHEN, K. 2007. A content-adaptive multi-view video color correction algorithm. In *International Conference on Acoustics, Speech and Signal Processing*. 26
- SHI, L., AND FUNT, B. 2012. Maxrgb reconsidered. *Journal of Imaging Science and Technology* 56, 2, 20501–1. 28
- SUN, J., SUN, J., XU, Z., AND SHUM, H.-Y. 2008. Image super-resolution using gradient profile prior. In *Computer Vision and Pattern Recognition (CVPR)*. 24
- SÜSSTRUNK, S., BUCKLEY, R., AND SWEN, S. 1999. Standard rgb color spaces. In *Color and Imaging Conference*. 17
- TAI, Y.-W., JIA, J., AND TANG, C.-K. 2005. Local color transfer via probabilistic segmentation by expectation-maximization. In *Computer Vision and Pattern Recognition (CVPR)*. 6, 26
- TAI, Y.-W., CHEN, X., KIM, S., KIM, S. J., LI, F., YANG, J., YU, J., MATSUSHITA, Y., AND BROWN, M. 2013. Nonlinear camera response functions and image deblurring:

- Theoretical analysis and practice. *Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*. 71
- THÉVENAZ, P., BLU, T., AND UNSER, M. 2000. Image interpolation and resampling. *Handbook of Medical Imaging, Processing and Analysis*. 24
- URBAN, P., DESCH, M., HAPPEL, K., AND SPIEHL, D. 2010. Recovering camera sensitivities using target-based reflectances captured under multiple LED-illuminations. In *Workshop on Color Image Processing*. 22
- VAN DE WEIJER, J., GEVERS, T., AND GIJSENIJ, A. 2007. Edge-based color constancy. *IEEE Transactions on image processing* 16, 9, 2207–2214. 28
- WANG, Q., SUN, X., AND WANG, Z. 2010. A robust algorithm for color correction between two stereo images. In *Asian Conference on Computer Vision (ACCV)*. 26
- WELSH, T., ASHIKHMIN, M., AND MUELLER, K. 2002. Transferring color to greyscale images. In *ACM Transactions on Graphics*. 27
- WEN, C.-L., HSIEH, C.-H., CHEN, B.-Y., AND OUHYOUNG, M. 2008. Example-based multiple local color transfer by strokes. In *Computer Graphics Forum*. 27
- WRIGHT, W. D. 1929. A re-determination of the trichromatic coefficients of the spectral colours. *Transactions of the Optical Society*. 14
- XIAO, X., AND MA, L. 2006. Color transfer in correlated color space. In *ACM International Conference on Virtual Reality Continuum and its Applications*. 27
- XIAO, X., AND MA, L. 2009. Gradient-preserving color transfer. In *Computer Graphics Forum*. 6, 27, 77, 82, 87, 88, 92

- XIONG, Y., SAENKO, K., DARRELL, T., AND ZICKLER, T. 2012. From pixels to physics: Probabilistic color de-rendering. In *Computer Vision and Pattern Recognition (CVPR)*. 2, 20, 21, 30, 56
- YAMAMOTO, K., KITAHARA, M., KIMATA, H., YENDO, T., FUJII, T., TANIMOTO, M., SHIMIZU, S., KAMIKURA, K., AND YASHIMA, Y. 2007. Multiview video coding using view interpolation and color correction. *Transactions on Circuits and Systems for Video Technology*. 26
- YUAN, L., AND SUN, J. 2011. High quality image reconstruction from RAW and JPEG image pair. In *International Conference on Computer Vision (ICCV)*. 24, 56, 67, 69, 74
- ZAIDI, Q., SPEHAR, B., AND DeBONET, J. 1997. Color constancy in variegated scenes: role of low-level mechanisms in discounting illumination changes. *JOSA A* 14, 10, 2608–2621. 28