A NEW IN-CAMERA COLOR IMAGING MODEL FOR COMPUTER VISION

LIN HAI TING

NATIONAL UNIVERSITY OF SINGAPORE

2013

A NEW IN-CAMERA COLOR IMAGING MODEL FOR COMPUTER VISION

LIN HAI TING
(*B.Sc., Renmin University of China, 2008*)

A THESIS SUBMITTED FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY

DEPARTMENT OF COMPUTER SCIENCE
NATIONAL UNIVERSITY OF SINGAPORE

2013

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.

Signature: _____ **Date:** _____

To my parents and wife.

Acknowledgments

I would like to express my deepest thanks and appreciation to my advisor Michael S. Brown for his motivation, enthusiasm, patience and brilliant insights. He is always supportive and kind. His extremely encouraging advices always rekindle my passion during the hard times in my research. I could not have asked for a finer advisor.

I feel tremendously lucky to have had the opportunity to work with Dr. Seon Joo Kim, and I owe my greatest gratitude to him. He initiated this work and continuously dedicated his passion and enlightening thoughts into this project, guiding me through the whole progress. Without him, this work would not have been possible.

I am grateful to the members of my committee Dr. Leow Wee Kheng and Dr. Terence Sim, for their effort, encouragement and insightful comments. Thanks also goes to Dr. Dilip Prasad for his careful review of this manuscript and helpful feedbacks on improving the writing.

Sincere thanks to my collaborators: Dr. Tai Yu-Wing and Dr. Lu Zheng. As seniors, you both have helped me tremendously, in working out the ideas, conducting the experiments and also have provided me thoughtful suggestions in every aspects.

I thank my fellow graduate students in NUS Computer Vision Group: Deng Fanbo, Gao Junhong and Liu Shuaicheng. Thank you for the inspiring discussions, for the overnight hard workings before deadlines and for the wonderful time we have spent together. I also would like to thank staffs and friends for their help during my experiments. Whenever I asked for it, they were always so generous to allow me to let their precious cameras go through numerous heavy testings.

I am heartily thankful to my other friends who appeared in my life during my Ph.D journey. Your constant supports, both physical and spiritual,

are the best gifts to me. Because of you, this foreign city becomes so memorable.

Last but certainly not least, I would like to express my great gratitude to my parents, for all the warmth, care and perpetual love you have given to me. Thanks also go to my two lovely elder sisters, for their warming concern and support. And of course, I am grateful to my wife. Since we first met, you have always been with me through good and bad times, encouraging me, supporting me and making my days so joyful.

Contents

Summary

Many computer vision algorithms, such as photometric stereo, shape from shading and image matching, assume that cameras are accurate light measuring devices which capture images that are directly related to the actual scene radiance. Digital cameras, however, are much more than light measuring devices; the imaging pipelines used in digital cameras are well known to be nonlinear. Moreover, the primary goal of many cameras is to create visually pleasing pictures rather than to capture accurate physical descriptions of the scene.

In this thesis, we present a study of the in-camera image processing through an extensive analysis of an image database collected by capturing images of scenes under different conditions with over 30 commercial cameras. The ultimate goal is to investigate if image values can be transformed to physically meaningful values and if so, when and how this can be done. From our analysis, we found a glaring limitation in the conventional imaging model employed to determine the nonlinearities in the imaging pipeline (i.e. radiometric calibration). In particular, the conventional radiometric models assume that the irradiance (RAW) to image intensity (sRGB) transformation is attributed to a single nonlinear tone-mapping step. However, this tone-mapping step alone is inadequate to describe saturated colors. As a result, such color values are often mis-interpreted by the conventional radiometric calibration methods.

In our analysis, we found that the color mapping component which includes gamut mapping has been missing in previous models of imaging pipeline. In this thesis, we describe how to introduce this step into the imaging pipeline based on Radial Basis Functions, together with calibration procedures to estimate the associated parameters for a given camera model. This allows us to model the full transformation from RAW to

sRGB with much more accuracy than demonstrated by prior radiometric calibration techniques.

Furthermore, an efficient nonuniform lattice regression calibration scheme is also proposed in order to speed up the in-camera color mapping process. The results demonstrate that this nonuniform lattice provides errors comparable to using an RBFs, but with computational efficiency which is an order of magnitude faster than optimized RBFs computation.

In addition, we demonstrate how our new imaging pipeline model can be used to develop a system that converts an sRGB input image captured with the wrong settings to an sRGB output image that would have been recorded under different and correct camera settings. The results of real examples show the effectiveness of our model.

This work, to our best knowledge, is the first to introduce gamut mapping into the imaging pipeline modeling. The proposed model achieves a new level of accuracy in converting sRGB images back to the RAW responses. Acting as a fundamental modeling of in-camera imaging pipeline, it should benefit many computer vision algorithms.

List of Tables

List of Figures

Chapter 1

Introduction

In computer vision, digital cameras are used as input instruments for electronically perceiving the scene. Many computer vision algorithms assume cameras are accurate light measuring devices which capture images that are directly related to the actual scene radiance. Fig. 1.1 shows a simple image formation process [21]. The intensities in an output image are considered to be proportional (up to digitization errors) to the scene irradiance reflected by the objects. Representative algorithms adopting this assumption include photometric stereo, shape from shading, image matching, color constancy, intrinsic image computation, and high dynamic range imaging.

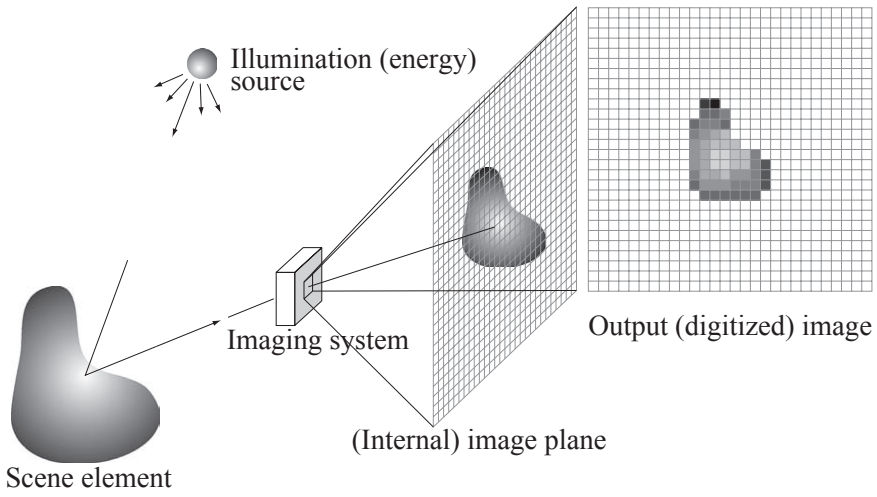


Figure 1.1: The digital image formation process. (Image from [21].)

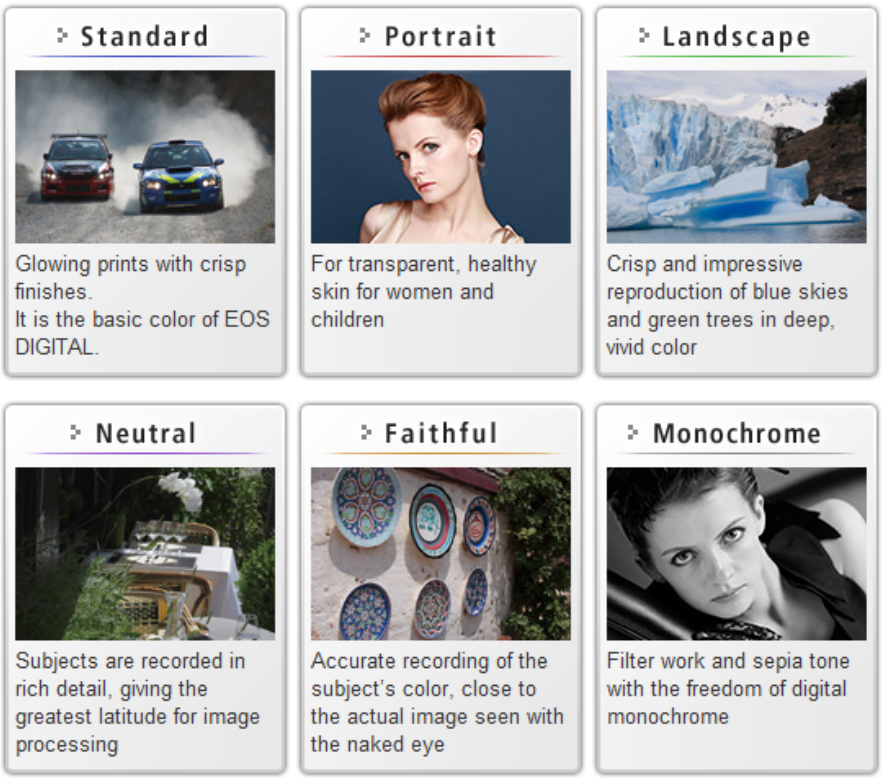


Figure 1.2: Picture styles of Canon EOS DIGITAL cameras. (Snapshot from Canon official site [5].)

However, digital cameras are much more than light measuring devices. This is evident from the variety of complicated functions available on consumer cameras. Typically, for a single-lens reflex (DSLR) camera, users can try to achieve desired effects when shooting by correctly setting options such as picture style[1], white balance, etc. A web page snapshot of preset picture styles available with Canon EOS DIGITAL is shown in Fig. 1.2 [5], where portrait style is introduced as "for transparent, healthy skin for women and children" and landscape style is introduced as "crisp and impressive reproduction of blue skies and green trees in deep, vivid color". White balance is another option that dramatically affects the outputs. Fig. 1.3 demonstrates

[1]Picture style refers to the photofinishing feature of Canon cameras to produce optimized pictures under specific scenes, such as portrait and landscape. Other camera manufacturers offer similar photofinishing styles, e.g. Nikon's "Image Optimizer" and Sony's "Creative Style". For simplicity, we collectively refer to these functions as *picture style*.

Incandescent Sunny Shade

Figure 1.3: Images of different white balance settings from a Nikon DSLR camera.

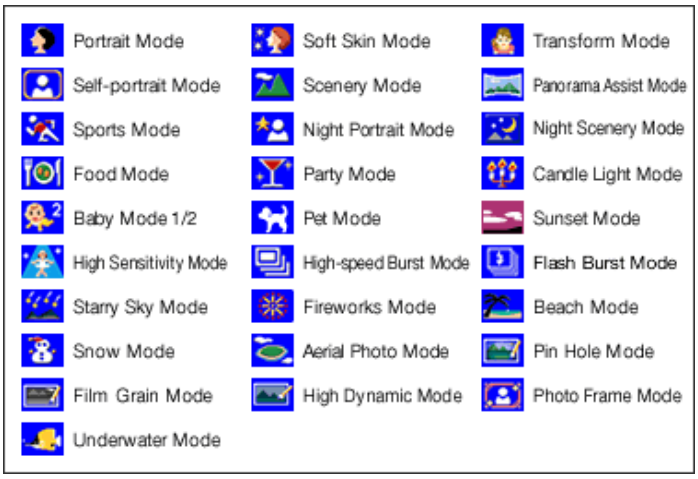


Figure 1.4: Different scene mode settings in a particular point-and-shoot camera Lumix DMC-ZS8(TZ18).(Snapshot from [50].)

the images of an outdoor scene with different white balance settings from a Nikon camera. While checking a point-and-shoot camera, we could even find more various options about the scene mode as shown in Fig. 1.4.

These various image rendering options reveal the complexity of the in-camera imaging pipeline, and also indicate that the primary goal of commercial cameras is to create visually pleasing pictures rather than to capture accurate physical descriptions

Figure 1.5: Color comparison between different cameras. The images are taken with the same settings including aperture, exposure, white-balance and picture style. The variation in the colors of the images is evident.

of the scene. Furthermore, each camera manufacturer has its own secret recipe to achieve this goal. It is well known among professional photographers that the overall color impressions of different cameras such as Canon and Nikon cameras are different. Fig. 1.5 compares the images from Canon, Nikon and Sony cameras with the same aperture, exposure, white balance and picture style shooting at the same indoor scene. In these images, there exist noticeable color differences in balloon region, background wall and skins.

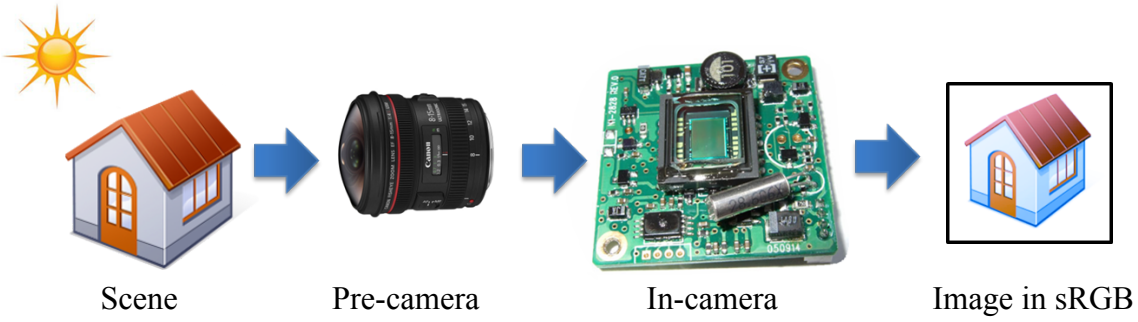


Figure 1.6: Summarization of image formulation process of a modern camera.

Fig. 1.6 summarizes the image formulation process that a modern camera appears to have. The scene irradiance transits through the lens, filtered by lens filter and partially absorbed. The amount of light falling on the sensor is controlled by the combination of shutter speed and aperture. These filtering or controls are in the pre-camera process. While in the in-camera process, the image sensor responds to the exposure and results in digital RAW pixel values. These RAW values are then manipulated on board (referred to as in-camera), realizing the rendering functions as mentioned above. Finally the color image in a standard color space such as sRGB color space is produced.

Image senors, such as charge-coupled device (CCD) and complementary metal-oxide-semiconductor (CMOS), convert photons into electronic voltage and finally the analog data are converted into digital RAW values. These digital RAW values are guaranteed to be linear [7] to the amount of incident light with the response properties of the sensor compensated. They are the most reliable linear descriptions of the scene from the shooting camera. Compared to the linear RAW values, the final color image in sRGB is highly nonlinear. sRGB is the abbreviation of standard RGB color space which is created by HP and Microsoft in 1996 for use on monitors, printers, and the Internet. Due to the overwhelming domination of monitors in digital image display, sRGB color space is the common color space supported by cameras in which the final images are represented. More details about sRGB color space could be found in Chapter 2.

Due to the variety of on-board processing, the natural question is what are the pixel values of output images reflecting about the scene? Can these values be transformed to physically meaningful values, ideally the RAW values, and if so, when and how can this be done? In the next section, detailed objectives of this work are specified.

1.1 Objectives

For the past decades, many researchers have been working on how to recover the relative scene irradiance from images [40, 12, 23, 39, 24, 36, 37, 34, 32, 33, 7]. These prior approaches has formulated the in-camera imaging pipeline as a mapping function, namely the response function of the camera, which maps the amount of light

collected by the image sensor to image intensities. We refer to this group of work as traditional imaging models.

In traditional imaging models, the focus has been on the response function estimation per color channel. The models are extended to color images in a relatively simple way. This results in unsatisfactory modeling of the in-camera processing. We specify the gaps in the traditional imaging models as follows:

- Response function-based formulation is relatively an oversimplified model of the in-camera imaging pipeline.
- Most of the current calibration techniques estimate the response function of each channel independently, instead of treating the RGB as a whole. This may lead to wrong conclusions when applied to color imaging.
- Many researchers accept the assumption that the response function is a fixed property for a given camera model. However, some researchers [7] disagree with that. This disagreement is due to the lack of systematic verification of the assumption.

The main aim of the study presented in this thesis is to propose a general model of in-camera imaging pipeline, so that a better understanding can be gained on the behavior of a camera in producing color images from its RAW responses. The specific objectives of this research were to:

- conduct thorough experiments to verify the assumption about the response function over a number of cameras from different camera companies.
- propose a generic, more sophisticated, and more accurate model for color imaging pipeline so that the main behavior of cameras in producing color images could be well represented. The relative scene irradiance should be accurately recovered from images by the inverse computation based on this model.
- develop another practical other than theoretically optimal representation for the model to achieve efficient evaluations in real applications.
- apply our model to practical photography problems such as white balance (WB) correction, and by this application, to further demonstrate the accuracy of our model.

The mathematical model of the in-camera processing (RAW to sRGB) proposed in this study should have significant impact on the imaging pipeline representation. All main imaging steps could be found in our model as separate components. In this way, it is shown clearly the fundamental differences between camera's RAW images and its sRGB outputs, which facilitates computer vision (CV) algorithm designers to select the optimal inputs for their specific applications. Further more, since color imaging is the basic means of obtaining vision information of the scene in CV, a better modeling of the camera should contribute to the whole CV community, and the ability of reversing sRGB back to RAW using our model should benefit those applications that rely on the availability of physical scene irradiance.

In this work, we focus on examining the behavior of a camera under Manual mode. It is understood that modern cameras are equipped with very powerful computing systems. With such computing ability, many extra operations for enhancing the image results can be performed on board. These extra operations are enabled when the cameras are set to other more complicated modes, such as "auto mode" where the camera automatically chooses the "optimal" settings and operations for users. These operations bring additional complexities into the imaging pipeline. Focusing only on "manual mode" enables us to conduct our experiments under full control. Furthermore, it also eliminates the extra disturbing elements. This elimination contributes to the establishment of a compact model explaining the core processing of the imaging pipeline. For more information about manual mode and other modes, please refer to the section 2.3.3 scene dependency and camera settings of Chapter 2.

1.2 Contributions

In correspondence to the objectives, we made the following contributions:

- We collected more than 10,000 images (both sRGB and RAW if applicable) from 31 cameras ranging from DSLR cameras to point and shoot cameras under different settings, including different picture styles and white balances, and conducted analysis on the collected data to verify the assumption about the response function being a fixed property of a certain camera (in other words, being scene independent).

- We proposed a generic and accurate model for color imaging pipeline. The critical step of gamut mapping was uniquely introduced and modeled as a Radial Basis Functions [4, 6]. Both forward (RAW to sRGB) and backward (sRGB to RAW) processes were modeled together with calibration procedures to estimate the associated parameters for a given camera model. Our results achieved much more accuracy than demonstrated by prior radiometric calibration techniques.

- Another compact and efficient representation of the imaging pipeline was proposed in order to speed up the in-camera color imaging process for practical applications. In this representation, we proposed a novel nonuniform lattice regression method to fit the underlying transforming function from sRGB to RAW and inverse.

- We demonstrated how our new imaging pipeline model can be used to develop a system that converts an sRGB input image captured with the wrong settings to an sRGB output image that would have been recorded under different and correct camera settings. Those settings include white balance and picture style settings.

1.3 Road map

In Chapter 2, more background information is provided. We describe our data collection and analysis in Chapter 3. The details of our proposed in-camera imaging model based on Radial Basis Functions and the experimental results are described in Chapter 4. Chapter 5 presents the non-uniform lattice regression technique used in formulating our model. Furthermore, applications of our model in photo editing are exhibited in Chapter 6. Chapter 7 discusses and concludes the work.

Chapter 2

Background

This chapter presents technical background information about in-camera imaging pipeline. In section 2.1, the general descriptions of stages in the pipeline are presented. Related topics about color, color spaces and gamut mapping are discussed in section 2.2. Section 2.3 reviews previous work on radiometric calibration.

2.1 Camera pipeline

Although the on board processes may be different in different camera models, they still follow a scheme of several generic stages. These stages include RAW responding of image sensor, white balancing, demosaicing, sharpening, color space transformation, color rendering, re-quantization and compression [7].

Scene radiance comes through the camera lens, followed by the color filters and hits the camera's image sensor, causing linear RAW responses. Generally, these color filters above the photosensors are arranged according to a pattern named Bayer pattern. Bayer pattern is a particular arrangement of the red, green and blue color filters over a square grid of photosensors, where 50% of the filters are green filters, 25% are red and the other 25% are blue. Fig. 2.1 shows an example of this pattern. Due to the presence of color filters, for each pixel, only the response value of one color channel is recorded. Therefore, demosaicing is needed to interpolate the missing values of each pixel for all three channels to generate a full color image. White balancing is applied to balance the three color components so that white objects appear white in the image. Sharpening is used for enhancing image details.

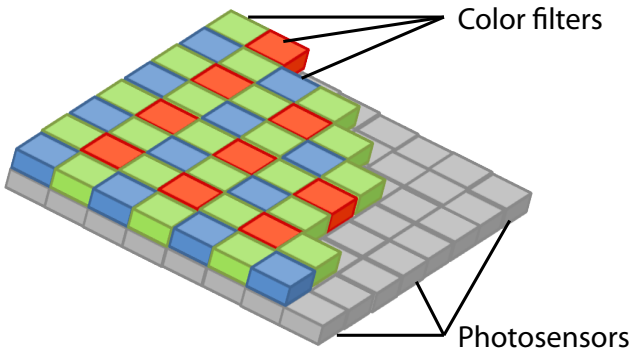


Figure 2.1: An example of Bayer pattern.

Demosaicing, white balancing, and sharpening, are generally applied directly on the RAW values which are in the cameras' RAW space. This RAW space is almost unique for each camera model. Those RAW values need to be transformed to standard color spaces, for example, the CIE XYZ color space, and finally transformed to sRGB or Adobe sRGB color space. Color rendering, which refers to how cameras modify the tristimulus result from the previous stages in order to represent them in the final output color space of limited gamut, is the most critical step in the imaging pipeline. It determines the final appearance of the image colors. Finally, this image will be quantized, compressed and then saved as a JPEG file.

These various stages affect the final output image to different extent. While sensor's RAW response, white balance, color space transformation, and color rendering are critical elements in generating the final outputs, demosaicing, sharpening, requantization, and compression are treated as introducing noise to the true values. In this work, we investigate those critical elements in order to understand the relationship between the final output image and the physical scene irradiance.

2.2 Color representation and communication

Before we examine the in-camera imaging pipeline of generating color images, we need to understand "color". Although color seems so familiar to us, the perception of color in our mind involves complicated physical and neural processes. Therefore, before discussing the representation and/or communication of color, we first investigate what

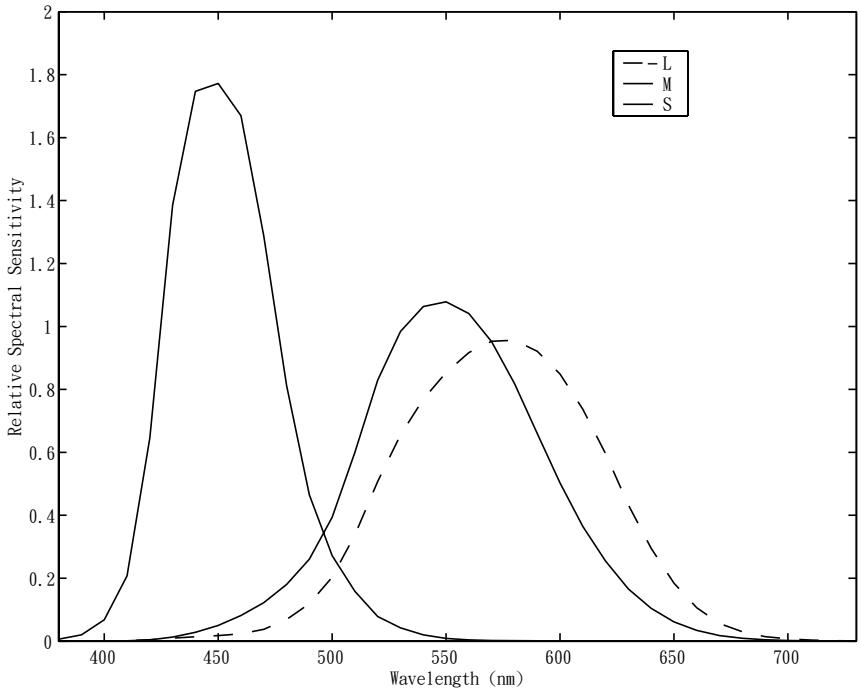


Figure 2.2: Relative spectral sensitivities of S, M and L cones. Image from [52].

is color.

2.2.1 Tristimulus

Interestingly, color is not a characteristic of an object. The perception of color is actually the neural system's interpretation of the signals sensed by the eyes. On the retina of the eyes, there are two kinds of light-sensitive photoreceptors: rods and cones. While rods contribute to the perception of shades of gray only and perform normally under very low light levels such as starlight, cones are the cells responsible for our color perception in normal vision.

There are three types of cones, namely S, M and L cones with their spectral sensitivities peaking at short, medium and long wavelengthes respectively. Fig. 2.2 shows the estimates of the effective sensitivities of the different cones. Their response

c_i to the incident light could be calculated as [52]:

$$c_i = \int_{\lambda_{min}}^{\lambda_{max}} s_i(\lambda) l(\lambda) d\lambda, \quad i = 1, 2, 3 \tag{2.1}$$

where λ represents the wavelength, λ_{min} and λ_{max} denote the lower and upper bound of the visible wavelengths, $s_i(\lambda)$ is the sensitivity of the cone of the i-th type, i.e. the curves in Fig. 2.2, and $l(\lambda)$ represents the spectral distribution of the light incident on the retina.

With sufficiently high sampling rate, Eq. 2.1 can be numerically written as

$$c_i = \sum_{j=0}^{N-1} s_i(\lambda_j) l(\lambda_j) \Delta\lambda = \mathbf{s}_i^T \mathbf{l}, \quad i = 1, 2, 3 \tag{2.2}$$

$$\text{where } \mathbf{s}_i = \Delta\lambda \cdot \begin{bmatrix} s_i(\lambda_0) \\ s_i(\lambda_1) \\ \vdots \\ s_i(\lambda_{N-1}) \end{bmatrix} \text{ and } \mathbf{l} = \begin{bmatrix} l(\lambda_0) \\ l(\lambda_1) \\ \vdots \\ l(\lambda_{N-1}) \end{bmatrix}.$$

The set $\{\lambda_j\}_{j=0}^{N-1}$ are uniformly spaced wavelengthes over the range $[\lambda_{min}, \lambda_{max}]$ with $\lambda_j = \lambda_0 + j\Delta\lambda$. Note that $\{\mathbf{s}_i\}_{i=1}^{3}$ are cone sensitivity vectors which absorb the influence of the scaling factor $\Delta\lambda$. Eq. 2.2 can be further compactly written in matrix-vector notation as

$$\mathbf{c} = \mathbf{S}^T \mathbf{l}, \tag{2.3}$$

where $\mathbf{c} = [c_1, c_2, c_3]^T$ and $\mathbf{S} = [\mathbf{s}_1, \mathbf{s}_2, \mathbf{s}_3]$.

This 3×1 vector $\mathbf{c}$ is known as a tristimulus vector. The array of $\mathbf{c}$'s from different cones are the input for later neural processes. The final color perception formed in the mind depends on many other factors, such as viewing condition, scene arrangement. As an illustration of how sophisticated the human vision system is, an extended checker shadow illusion originally published by Edward H. Adelson [1] is shown in Fig. 2.3. We perceive the two grids A and B as different patches, but actually they are identical in color, including the center brown dots. Although two same responses could be treated as different colors when conditions differ, by associating the tristimulus vector with a well defined standard condition, we could still uniquely specify a color

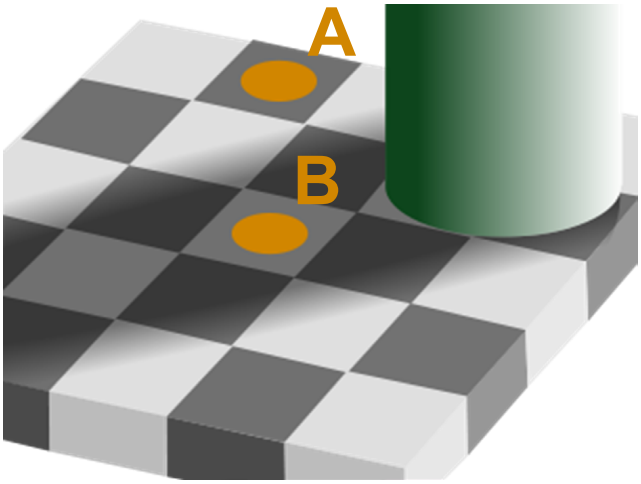


Figure 2.3: Checker shadow illusion [1]. Grids A and B have the same intensity although they are perceived to be different.

using vector $\mathbf{c}$. This concept served as the inspiration of the standard color space. In the next subsection, we discuss some color spaces based on this mathematical representation of color.

2.2.2 Color spaces

In this subsection, the basis of color matching is first explained. Next, color spaces are introduced and a linear relationship between different color spaces is further derived. Finally, several example color spaces including CIE XYZ, CIE RGB, CIE xyY and sRGB are briefly presented.

Color matching

From the previous subsection, we know that the tristimulus vector uniquely specifies a color. Two spectra $\mathbf{l}_A$ and $\mathbf{l}_B$ have the same color, i.e. $\mathbf{c}_A = \mathbf{c}_B$, if the following equation holds:

$$\mathbf{S}^T\mathbf{l}_A = \mathbf{S}^T\mathbf{l}_B. \tag{2.4}$$

Since inherently the color vector is three dimensional, any given color/tristimulus

c could be matched using linear combination of three color primaries, which is

$$\mathbf{c} = \mathbf{S}^T\mathbf{l} = \mathbf{S}^T\mathbf{P}\mathbf{a}(\mathbf{l}) \tag{2.5}$$

$$\text{and } \mathbf{a}(\mathbf{l}) \doteq (\mathbf{S}^T\mathbf{P})^{-1}\mathbf{S}^T\mathbf{l} \tag{2.6}$$

where $\mathbf{P} = [\mathbf{p}_1, \mathbf{p}_2, \mathbf{p}_3]$ is the matrix of three color primaries $\mathbf{p}_1$, $\mathbf{p}_2$, and $\mathbf{p}_3$. In definition, $\mathbf{p}_1$, $\mathbf{p}_2$, and $\mathbf{p}_3$ are colorimetrically independent, i.e. $\mathbf{S}^T\mathbf{p}_1$, $\mathbf{S}^T\mathbf{p}_2$, and $\mathbf{S}^T\mathbf{p}_3$ are linearly independent, which asserts the existence of $(\mathbf{S}^T\mathbf{P})^{-1}$. The 3×1 vector $\mathbf{a}(\mathbf{l})$ is the combination weights for the three primaries in matching the color of spectrum **l**. This phenomenon of color matching by using three color primaries is well known as trichromacy.

Color spaces and their relationships

Based on trichromacy, given three primaries, a 3D color space could be defined. A point in this color space represents the color matched by the weighted combination of the three primaries. Any spectrum **l** has its corresponding point $\mathbf{a}(\mathbf{l})$ in this space. Specifically, the unit intensity monochromatic spectra $\{\mathbf{e}_i\}_{i=1}^N$, ($\mathbf{e}_i$ is a N-vector with a one in the ith position and zeros elsewhere), have their corresponding points $\{\mathbf{a}_i | \mathbf{a}_i = (\mathbf{S}^T\mathbf{P})^{-1}\mathbf{S}^T\mathbf{e}_i\}_{i=1}^N$. With the help of $\mathbf{a}_i$'s, the corresponding point for spectrum **l** can be calculated as:

$$\begin{aligned}\mathbf{a}(\mathbf{l}) &= (\mathbf{S}^T\mathbf{P})^{-1}\mathbf{S}^T\mathbf{l} \\ &= (\mathbf{S}^T\mathbf{P})^{-1}\mathbf{S}^T[\mathbf{e}_1\mathbf{e}_2\ldots\mathbf{e}_N]\mathbf{l} \\ &= [\mathbf{a}_1\mathbf{a}_2\ldots\mathbf{a}_N]\mathbf{l} \\ &= \mathbf{A}^T\mathbf{l},\end{aligned} \tag{2.7}$$

where

$$\mathbf{A} = [\mathbf{a}_1\mathbf{a}_2\ldots\mathbf{a}_N]^T = \mathbf{S}(\mathbf{P}^T\mathbf{S})^{-1}, \tag{2.8}$$

and the columns of matrix **A** are referred to as the color-matching functions (CMFs) associated with the primaries **P**. Furthermore, with simple derivation, the CMFs **A**

and the primaries $\mathbf{P}$ are correlated to each other:

$$\mathbf{A}^T\mathbf{P} = \mathbf{I}. \tag{2.9}$$

Assuming another set of primaries $\mathbf{Q}$ and the corresponding CMFs matrix $\mathbf{B} = \mathbf{S}(\mathbf{Q}^T\mathbf{S})^{-1}$, the corresponding points in these two color spaces for spectrum $\mathbf{l}$, $\mathbf{a_A}(\mathbf{l})$ and $\mathbf{a_B}(\mathbf{l})$ respectively, are related as:

$$\mathbf{a_A}(\mathbf{l}) = \mathbf{A}^T\mathbf{l} = \mathbf{A}^T\mathbf{Q}(\mathbf{B}^T\mathbf{l}) = \mathbf{A}^T\mathbf{Q}\mathbf{a_B}(\mathbf{l}), \tag{2.10}$$

since

$$\mathbf{A}^T = (\mathbf{S}^T\mathbf{P})^{-1}\mathbf{S}^T = (\mathbf{S}^T\mathbf{P})^{-1}(\mathbf{S}^T\mathbf{Q})(\mathbf{S}^T\mathbf{Q})^{-1}\mathbf{S}^T = \mathbf{A}^T\mathbf{Q}\mathbf{B}^T. \tag{2.11}$$

From Eq. 2.10, we can see that the transformation between two color spaces of different primaries is a linear transformation, which is $\mathbf{A}^T\mathbf{Q}$.

This primaries-based color space definition is natural in describing the color space of output electrooptical devices, such as displays and projectors, and is convenient as well in expressing transformations between spaces of different sets of primaries. For those devices, the primaries are naturally the physical spectra produced by the devices that are finally seen by human eyes. However, for the input optoelectronic devices, such as scanners and digital cameras, which respond to the physical spectra and generate digital images, it is not clear what the primaries are when we treat the digital values as in a primaries based color space. The basic spectral property of those devices are the spectral sensitivities, as analogous to the sensitivities of the cones. However, the digital images should be finally seen on an output device, which makes it a necessity to relate the color spaces of input devices to the primaries based color spaces.

Let the spectral sensitivities of an input device be $\mathbf{S}_d$. This can be related to the cone sensitivities $\mathbf{S}$ using linear transformation plus a residual as follows:

$$\mathbf{S}_d^T = \mathbf{T}\mathbf{S}^T + \mathbf{S}_\Delta^T, \tag{2.12}$$

where $\mathbf{T}$ is a 3×3 transformation matrix which is usually of full rank and $\mathbf{S}_\Delta$ is the residual sensitivity matrix. Assuming $\mathbf{T}$ is invertible, the color of a spectrum $\mathbf{l}$ will

be

$$\mathbf{c} = \mathbf{S}^T\mathbf{l} = \mathbf{T}^{-1}(\mathbf{S}_d^T - \mathbf{S}_\Delta^T)\mathbf{l} = \mathbf{T}^{-1}\mathbf{c}_d - \mathbf{T}^{-1}\mathbf{S}_\Delta^T\mathbf{l}, \tag{2.13}$$

where $\mathbf{c}_d = \mathbf{S}_d^T\mathbf{l}$ is the digital response of the spectrum $\mathbf{l}$ of the device. From Eq. 2.13, we can see that unless $\mathbf{S}_\Delta = 0$, the correspondence between $\mathbf{c}$ and $\mathbf{c}_d$ will be spectral dependent and is hard to recover since the information after the sensing procedure is only the convoluted response $\mathbf{c}_d$. This condition $\mathbf{S}_\Delta = 0$ was introduced as the Luther condition [38, 28] back in the 1927, which is rarely satisfied due to manufacturing reasons. Therefore a certain color management technique is required to adjust the colors to the "right" positions to account for the residual part. This will be further explained in section 2.2.3. With the approximation $\mathbf{S}_\Delta = 0$, the relation between the color space of an input device and any other space of primaries $\mathbf{P}$ will be:

$$\mathbf{c} = \mathbf{S}^T\mathbf{l} = \mathbf{T}^{-1}\mathbf{S}_d^T\mathbf{l} = \mathbf{S}^T\mathbf{P}((\mathbf{S}^T\mathbf{P})^{-1}\mathbf{T}^{-1}\mathbf{S}_d^T\mathbf{l}), \tag{2.14}$$

where we can see the corresponding CMFs is a linear combination of the sensitivities $\mathbf{S}_d\tilde{\mathbf{T}}$, where $\tilde{\mathbf{T}} = \mathbf{T}^{-T}(\mathbf{P}^T\mathbf{S})^{-1}$. Since the primaries and the CMFs are correlated to each other (see Eq. 2.9), we can directly treat $\mathbf{S}_d$ as the CMFs and calculate its corresponding primaries as $\tilde{\mathbf{P}} = \mathbf{S}_d({\mathbf{S}_d}^T\mathbf{S}_d)^{-1}$. Having $\tilde{\mathbf{P}}$ as the primaries and based on Eq. 2.11, in our proposed model in later chapters, a linear transformation is adopted between the camera raw space and a standard color space.

Color space examples

One of the earliest mathematically defined and most widely used standard basic color spaces is the CIE 1931 XYZ color space, which is derived from CIE RGB color space by modifying its primaries so as to avoid negative RGB values. The CMFs of CIE RGB color space were directly constructed from experiments where each monochromatic test primary was matched by normal observers through the adjustment of the combination amounts of the three CIE RGB primaries [56]. In this way, the estimation of cone sensitivity matrix $\mathbf{S}$, which is difficult to measure directly , was avoided. Fig. 2.4 shows the CMFs of CIE RGB and CIE XYZ color spaces. Note that there are negative values in the CMFs of CIE RGB color space, which do not exist in that of CIE XYZ color space.

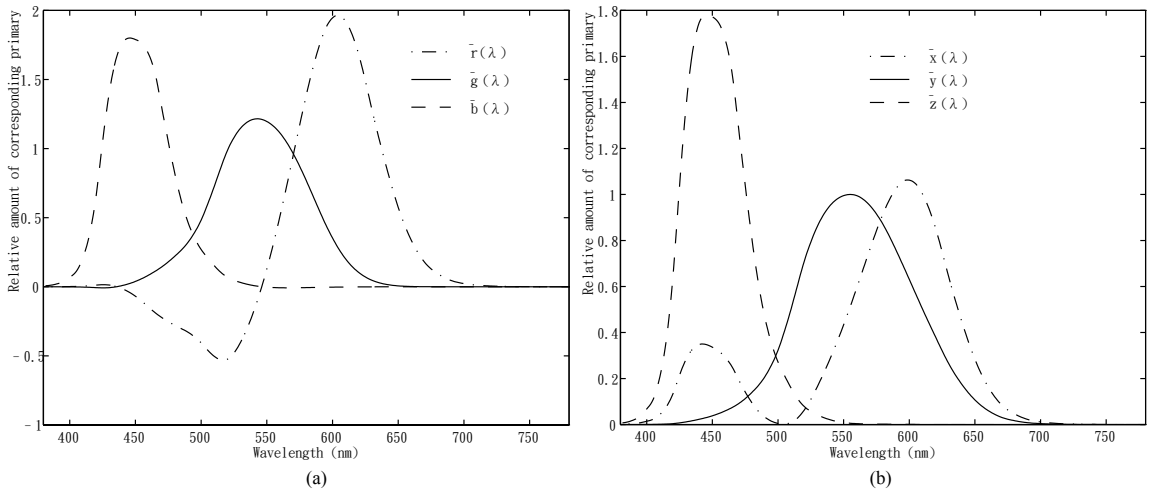


Figure 2.4: (a) CIE RGB and (b) CIE XYZ color matching functions. Images from [52].

In CIE XYZ color space, all perceivable colors are represented in the non-negative region. With the normalization stated below, a more intuitive color space, which divides the concept of color into brightness and chromaticity, is derived:

$$\begin{aligned} x &= \frac{X}{X+Y+Z}, \\ y &= \frac{Y}{X+Y+Z}, \\ z &= \frac{Z}{X+Y+Z} = 1-x-y, \\ Y &= Y, \end{aligned} \tag{2.15}$$

where x, y components specify the chromaticity and Y indicates the level of "brightness". This is the well known CIE xyY color space. With this space, chromaticities alone could be easily plotted in 2D figure. Fig. 2.5 shows the chromaticity diagram. In this diagram, the colored horseshoe-shaped region is the gamut of human vision.

Another color space extensively used for monitors, printers and the internet is the standard RGB color space, in abbreviation, sRGB. It is also a common space in which modern cameras represent their final digital images. The sRGB color space is well

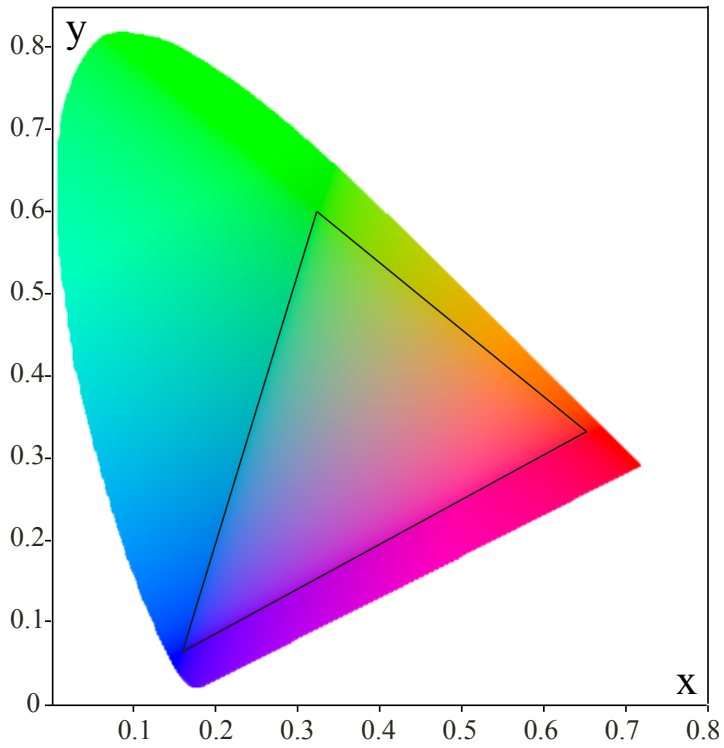


Figure 2.5: The CIE 1931 color space chromaticity diagram. The triangle indicates the gamut of sRGB color space.

designed for monitor displays: it uses primaries of the ITU-R recommendation [31] which is the same as in studio monitors; it defines a nonlinear mapping between the intensity of each primary and the actual number recorded in order to compensate the gamma response of a CRT monitor.

Accordingly, the transformation from CIE XYZ color space to sRGB color space involves two steps:

1. Linear transformation

$$\begin{bmatrix} R_{linear} \\ G_{linear} \\ B_{linear} \end{bmatrix} = \begin{bmatrix} 3.2406 & -1.5372 & -0.4986 \\ -0.9689 & 1.8758 & 0.04715 \\ 0.0557 & -0.2040 & 1.0570 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} \tag{2.16}$$

where R_{linear}, G_{linear}, B_{linear} are the intermediate parameters before nonlinear gamma compensation. They are considered as in linear sRGB color space. Note that in Eq. 2.16, X, Y and Z are normalized values by scaling the original XYZ with a suitable value and then clipping them into the range $[0, 1]$. This determines the maximum original XYZ values that sRGB can represent. The linear transformation matrix reflects the original XYZ values which map to the white color in sRGB and will change if another XYZ values are specified as the white color. In this Eq. 2.16, the white color in sRGB corresponds to CIE standard illuminant D65 [48], i.e. the chromatic $x = 0.3127$, $y = 0.3290$. Unlike CIE XYZ space, the range of each dimension of sRGB space is $[0, 1]$ instead of $[0, +\infty)$.

2. Gamma compensation

For each channel $C_{linear} \in \{R_{linear}, G_{linear}, B_{linear}\}$, the final $C_{srgb} \in \{R_{srgb}, G_{srgb}, B_{srgb}\}$ is calculated as:

$$C_{srgb} = \begin{cases} 12.92C_{linear}, & C_{linear} \leq 0.0031308 \\ (1+a)C_{linear}^{1/2.4} - a, & C_{linear} > 0.0031308 \end{cases} \tag{2.17}$$

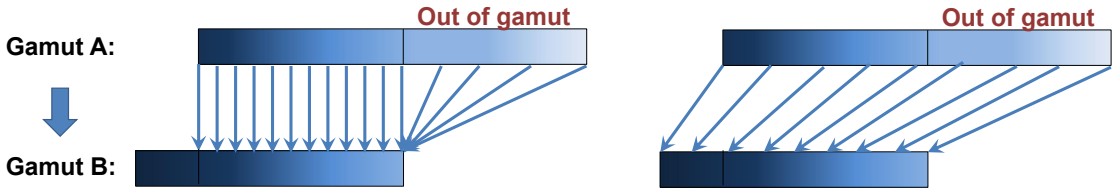
where $a = 0.055$.

Due to the limit of dimension range (which is necessary in practice since negative or infinite amount of light is impossible to produce) and its adopted primaries, the possible colors represented by the sRGB space are inside the triangle shown in Fig. 2.5. This set of colors is considered as the gamut of the sRGB color space.

Besides the aforementioned color spaces, many other color spaces have been proposed for different purposes, such as CIELUV, CIELAB and Adobe RGB color spaces. While these color spaces are important ones, they are out of the scope of this thesis.

2.2.3 Gamut mapping

As mentioned in the previous subsection, the gamut of a color space is the range of colors that can be represented in that space by definition. It is determined by the three primaries and the data range of each dimension. The gamut can also be associated with a certain device. Ideally, the device has its primaries and dimension range, which define the gamut. This may be subject to other factors such as viewing condition. For example, the gamuts of printed image under low illumination and high illumination are different. We consider the effects of those factors as modifying the



(a) Gamut clipping (b) Gamut compression

Figure 2.6: A 1D illustration of the comparison between examples of (a) gamut clipping and (b) gamut compression. This mapping is from a 1D gamut A to a 1D gamut B.

primaries and/or dimension ranges.

When reproducing colors from one gamut/device to another, these gamuts probably mismatches to each other, which means their color representation abilities differ. Therefore, gamut mapping is required in order to reproduce the colors in the target gamut. In this case, if a color point of the original gamut is located outside the target gamut, called out-of-gamut color, this color needs to be assigned to another color point within the target gamut so that it can be reproduced as effectively as possible, despite the mismatch.

The simplest way of assigning out-of-gamut colors to those within the gamut is gamut clipping. Gamut clipping basically maps those out-of-gamut colors to the nearest colors inside the target gamut and keeps those originally inside colors untouched. It is a preferable approach where accurate reproduction [44] is the gamut mapping intent, since it changes only the out-of-gamut colors due to gamut differences. However, most of the cases desire perceptually pleasing reproduction which requires the mapping to be smooth and continuous in nature. In other words, the variation between the colors, not the actual colors, should be preserved. In order to do that, gamut compression algorithms are proposed to continuously compress the outside colors inside or expand the inside colors toward the boundary of the target gamut. Unlike gamut clipping, this compression is applied to all original colors. Fig. 2.6 shows the comparison between these two gamut mapping algorithms. Other gamut mapping algorithms such as combining the aforementioned two approaches are possible. For example, a composite gamut mapping algorithm could be developed with a core gamut defined to be untouched and colors outside the core gamut being compressed [3].

In the case of photography, while rendering the colors in the standard color space such as sRGB color space from the RAW data of the image sensor, pleasant reproduction is the right choice. Furthermore, cameras, as color input devices, require gamut mapping to compensate the color mismatches due to the unfulfillment of condition $\mathbf{S}_\Delta = 0$ (see Eq. 2.13). This compensation, although is not enough to completely solve the spectral dependence issue, should be continuous in the working color space. We show the importance of this gamut mapping step in modeling the in-camera imaging pipeline in the following chapters.

2.3 Previous work

As mentioned in Chapter 1, there is little work about directly modeling the imaging pipeline. In the traditional imaging models, radiometric calibration forms the critical part of formulating the imaging pipeline. We briefly explain what radiometric calibration is in subsection 2.3.1 and how it was solved traditionally in subsection 2.3.2. The solving methods are classified into two categories according to the required input: calibration from multiple images with different exposures and calibration from a single image. Finally, a discussion about scene dependency and camera settings is presented in subsection 2.3.3.

2.3.1 Radiometric calibration formulation

In radiometric calibration, the nonlinearity in the camera pipeline is captured by the response function (f), which maps the relative amount of light collected by each sensor pixel (irradiance e) to pixel intensities (I) of the output image. Mathematically:

$$I_x = f(e_x), \tag{2.18}$$

where x is the pixel location. Eq. 2.18 can be extended to deal with color as follows [7]:

$$\begin{bmatrix} I_{rx} \\ I_{gx} \\ I_{bx} \end{bmatrix} = \begin{bmatrix} f_r(e_{rx}) \\ f_g(e_{gx}) \\ f_b(e_{bx}) \end{bmatrix}, \tag{2.19}$$

$$\mathbf{e}_x = \begin{bmatrix} e_{rx} \\ e_{gx} \\ e_{bx} \end{bmatrix} = \mathbf{T}\mathbf{E}_x. \tag{2.20}$$

The term $\mathbf{T}$ is a 3×3 transformation that captures both the transformation from the camera's color space ($\mathbf{E}_x$) to a linearized standard color space, i.e. linearized sRGB color space ($\mathbf{e}_x$) and white balancing.

The radiometric mapping f is the most critical part of the imaging pipeline and is almost always nonlinear due to the design factors built into digital cameras for a variety of reasons, including compressing the dynamic range of the imaged scene (tone-mapping), accounting for nonlinearities in display systems (gamma correction), mimicking the response of films, or to create aesthetic effects [24, 37]. When the response function f is known, the image intensities can be inverted back to relative scene radiance values. This inversion facilitates physics-based photometric analysis of the scene. While knowing the response function, as Eq. 2.20 shows, the whole imaging pipeline modeling could be completed by using the transformation $\mathbf{T}$.

2.3.2 Radiometric calibration algorithms

Calibration from multiple images with different exposures

To estimate this response function, many conventional radiometric calibration algorithms rely on multiple images of a scene taken with different exposures. Assuming constant radiance, which implies constant illumination, a change in intensity is explained by a change in exposure. Given a pair of images with an exposure ratio of k', the response function f is computed by solving the following equation from intensity values (I, I') at corresponding points:

$$\frac{f^{-1}(I'_x)}{f^{-1}(I_x)} = \frac{e'_x}{e_x} = k'. \tag{2.21}$$

In log domain, let $g = logf^{-1}$ and $K' = logk'$, the Eq. 2.21 can be written as:

$$g(I'_x) - g(I_x) = K'. \tag{2.22}$$

The main difference among various calibration methods is the model used to represent the response function. In the last few decades, many different models have been proposed in order to have an accurate and practical estimation of the response function. By assuming that the exposure ratios are known and the response curve can be modeled as gamma curve, Mann and Picard (1995) [40] proposed the problem and the perspective of recovering high dynamic range by combining multiple exposed images. Although this method is limited due to the simplicity of the function form, it has inspired more work following this direction. Debevec and Malik (1997) [12] proposed a non-parametric model and represented f as a vector. By imposing a smoothness constraint and assuming known exposure ratios, they recovered this vector which maps the intensity from 0 to 255 back to relative scene irradiance values. They also showed how high dynamic range (HDR) map can be generated based on the radiance response function and how HDR can be applied to generate realistic motion blur. In the work of Mitsunaga and Nayar (1999) [41], the response function was modeled as a low degree polynomial. An iterative way was employed to refine the function and rough exposure ratios. Grossberg and Nayar (2004) [24] introduced a new model called the empirical model of response (EMoR) which extracts the principle components for the response function from the database of existing response functions using principle component analysis (PCA). This model takes advantage of the knowledge about the shapes of response functions and thus enables recovery of the curve from a few data samples. Other than the work in [33], which explained the color as having the same response function for all three channels but with different exposure level per channel, those methods mentioned above do not deal with color and compute the response function independently per channel.

While different radiometric calibration methods vary in either how the response function is modeled and/or computed, all methods share a common view that it is a fixed property of a given camera model. In fact, this view was exploited to compute the radiometric response function by applying statistical analysis on images downloaded from the web in [34]. One exception is the work in [49] where the response function was modeled differently per image using a probabilistic approach. Another

exception is the recent work in [7] where the goal was to provide an analysis of the factors that contribute to the color output of a camera for internet color vision. They proposed a 24-parameter model to explain the imaging pipeline and the parameters are iteratively computed using available RAW data. Through their analysis, they suggested that the color rendering function f is scene-dependent. They further suggested that fixed nonlinearities per channel/camera as used in traditional radiometric calibration are often inadequate.

The majority of models mentioned above were extended to color in relatively simple ways. Furthermore, their main assumption that the response function is a fixed property is challenged by the work in [7].

Calibration from a single image

If the process is scene dependent as mentioned in [7], traditional radiometric calibration assuming fixed response function would be inadequate since f is altered while the exposure is changed. The only option would be to use single-image based radiometric calibration methods.

Instead of using multiple images, Farid (2001) [14] treated the radiometric nonlinearity as a gamma correction and computed the gamma correction from a single image through a minimization process. In this method, the fact that gamma correction introduces specific higher-order correlations in the frequency domain was exploited. Thus by detecting and minimizing these correlations, the gamma correction was estimated. However this method also suffers from the limitation of a simple model which is usually violated in practice.

In the work by Lin et al. (2004) [36], the color distributions of local edge regions in a single image was used for computing the response function. In observation, the measured colors across edges illustrate a non-linear distribution in color space while they should have a linear distribution if only simple blending occurs. By linearizing the non-linear distributions, they recovered the response function which is responsible for the non-linearity. This idea was extended in Lin and Zhang's subsequent work which deals with a single grayscale image by using the histograms of edge regions (Lin and Zhang, 2005 [37]). While the single image calibration algorithms are conceptually the best choice, they are sometimes unstable because they rely on edge regions, which are sensitive to noise and may be distorted due to further processing such as

sharpening onboard camera.

Gap

These calibration algorithms mentioned above are all based on formulating the cameras response functions defined in Eq. 2.18 and 2.19, which are independent in different color channels. The nonlinearity is only explained by this channel independent radiometric mapping f, without examining the effects from gamut mapping and color manipulation, which act on color as a whole. However, the effects of these steps on the final output are so significant that if ignored, the fitting errors will not be satisfying (the results can be found in the later chapters, e.g. Fig. 4.5). Furthermore, the work in [7] suggested that fixed nonlinearities per channel/camera as used in traditional radiometric calibration are often inadequate and we have to adopt scene dependent response functions to optimally reduce the fitting error.

2.3.3 Scene dependency and camera settings

Before moving forward, it is important to clarify the issue of scene dependency of the in-camera imaging process. If the process is scene dependent as mentioned in [7], traditional radiometric calibration would be inadequate and the only option would be to use single-image based radiometric calibration methods. However, these single image calibration algorithms are sometimes unstable due to their reliance on edge regions, which are sensitive to noise and may altered by further sharpening.

There are generally two color rendering strategies with regards to how digital cameras convert image sensor RAW responses to the final output: the *photofinishing* model and the *slide or photographic reproduction* model [27]. The digital camera community defines color rendering as the operations that apply the tone/color reproduction aims for the imaging system and change the state of the image from a scene-referred image state to an output-referred image state [30]. Color rendering transforms may include tone and gamut mapping to account for the dynamic range and color gamut of the output color image encoding (e.g., sRGB, Adobe RGB), compensation for differences in the input and output-viewing conditions, and other color adjustments (e.g., selective hue, saturation) to account for color reproduction preferences of the human observer.

The intent of the "photofinishing" color rendering operations is to produce a pleasing image that is not solely dependent on the exposure received by the image sensor. In this model, the imaging pipeline varies the color rendering based on the captured scene, possibly in a spatially varying manner. The auto-mode in cameras will trigger the photofinishing model as well as the optimizers such as Dynamic Lighting Optimizer on the Canon EOS550D and D-Range Optimizer in Sony α-200. Different photofinishing methods can also be associated with scene modes, e.g. Party, Landscape, Nighttime, etc. This scene dependent processing usually results in a spatially variant rendering, hence even one set of response functions per image, which are unvaryingly applied to different regions of the image, would not solve the linearization problem. For images produced under such settings, it is very difficult to convert image values to physically meaningful values.

The photographic reproduction model, on the other hand, uses fixed color rendering. For most high-end cameras, it is possible to set the camera in this photographic mode by turning the camera settings to manual and turning off all the settings pertaining to photofinishing, such as Dynamic Lighting Optimizer. For the remainder of this thesis it is assumed that the algorithms discussed are intended to work in the photographic reproduction mode.

Being aware of the missing piece in the traditional formulation and facing the suggestion of scene dependent response functions, we investigate the imaging pipeline and propose a new model which incorporates the gamut mapping and color manipulation. Based on this insight, we conducted thorough data collection and analysis in the next chapter. The question that for a certain camera model, whether the response function is a fixed property or not is also answered in Chapter 3 before we describe the details of our new model in Chapter 4.

Chapter 3

Data collection and analysis

In the previous chapter, we introduced technical background information about the imaging pipeline and reviewed the related work on radiometric calibration. In this chapter, we describe the collection step of the image data in section 3.1 and show the key observations of our analysis in section 3.2.

3.1 Data collection

For the analysis, we collected more than 10,000 images from 31 cameras ranging from DSLR cameras to point-and-shoot cameras. Images were taken in manual mode under different settings including white balance, aperture, and shutter speed. Images were also collected under different lighting conditions: indoor lighting and/or outdoor cloudy condition. Images were captured three times under the same condition to check the shutter speed consistency. RAW images were also saved if the camera supports RAW and the RAW files were rendered using the software *dcraw* [9]. The command used was *dcraw -v -D -4 -T*. We additionally use the database in [7] which includes over 1000 images from 35 cameras. Cameras from most of the major manufacturers are included as shown in Fig. 4.3. Although the cameras used for data collection are not uniformly distributed among the manufacturers, they reflect the reality of certain manufacturers being more popular than others.

The target objects for our dataset are two Macbeth ColorChecker charts, specifically a 24-patch chart and a 140-patch chart. There are several reasons why these color charts were used for our analysis. First, since the patches are arranged in a

regular grid pattern, we can automatically extract colors from different patches with simple registration. Second, measurements from different pixels within a patch can be averaged to reduce the impact of image noise on the analysis. Finally, these color charts include a broad spectrum of colors and different gray levels, which facilitate radiometric calibration and color analysis.

3.2 Data analysis

Using the conventional radiometric model, pairs of intensity measurements at corresponding patches in two differently exposed images constitute all the necessary information to recover the radiometric response function of a camera [23]. These pairs can be arranged into a plot that represents the *brightness transfer function* (BTF [33]), which can be formulated from Eq. 2.21 as

$$I'_x = \tau_k(I_x) = f(k' f^{-1}(I_x)), \tag{3.1}$$

where τ_k is the BTF, f is the response function, and k' is the exposure ratio. The BTF describes how image intensity changes with respect to an exposure change under a given response function. If the response function is a fixed property of a camera and the model in Eq. 2.18 is valid, the BTF should be the same for all pairs of images that share the same exposure ratio regardless of other camera settings and lighting conditions. Notice that even if we consider the color transformation in Eq. 2.19, the BTFs should still remain the same for the same exposure ratio as long as the color transformation remains unchanged between images, i.e.:

$$\frac{f^{-1}(I'_{cx})}{f^{-1}(I_{cx})} = k' \frac{\mathbf{t}'_c \mathbf{E}_\mathbf{x}}{\mathbf{t}_c \mathbf{E}_\mathbf{x}} = k', \quad \textit{if } \mathbf{t}_c = \mathbf{t}'_c. \tag{3.2}$$

In the above equation, $\mathbf{t}_c$ is a row of the color transformation $\mathbf{T}$ that corresponds to the color channel c.

To validate the model in Eq. 2.18 and the assumption that the response f is a fixed property of a camera, we compare the BTFs of different cameras under different settings. Representative examples from two cameras are shown in Fig. 3.1 for clarity. In the figure, each point represents the change in brightness for a given patch between the image pair.

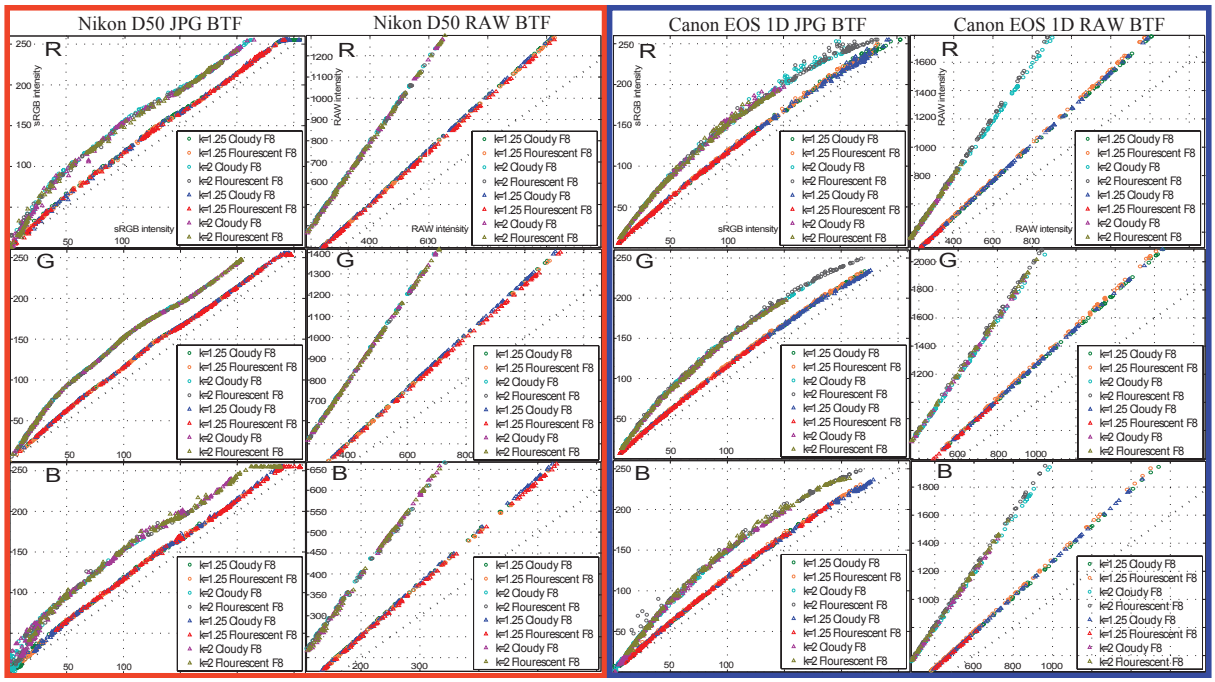


Figure 3.1: Brightness transfer functions for Nikon D50 and Canon EOS-1D. Each plot includes several BTFs with different exposure ratios (1.25 and 2.0), different lighting environments (◯: outdoors, △: indoors), and different white balance settings (cloudy and fluorescent). The key observation from these plots is that the BTFs of sRGB images with the same exposure ratio exhibit a consistent form aside from outliers and small shifts. For better viewing, please zoom the electronic PDF.

Through our analysis of the database, we made several key observations, which can also be observed in Fig. 3.1. The BTFs of a given camera and exposure ratio exhibit a consistent shape up to slight shifts and a small number of measurement outliers. BTFs recorded in the green channel are generally more stable than in the other channels and have a smaller amount of outliers. Also, the appearance of shifts and outliers tends to increase with larger exposure ratios.

The shifts can be explained with the inconsistency of the shutter speed. In our experiments, we control the exposure by changing the shutter speed[1], and it is well known that the shutter speeds of cameras may be imprecise [26]. In particular, we have found that shutter speeds of cameras with high total shutter-usage count tend to be less accurate, as observed from measurement inconsistency over repeated image captures under the same setting. We should note that we can rule out the illumination change as a cause because of our illumination monitoring from other cameras under the same conditions capturing repeated images and the consistent BTFs result from those other cameras. As these shifts also exist in RAW image BTFs, onboard camera processing can also be ruled out.

We found that some outliers, though having intensity values well within the dynamic range of the given color channel, have a 0 or 255 intensity value in at least one of the other channels. These clipped values at the ends of the dynamic range do not accurately represent the true scene irradiance.

One significant reason for outliers observed is that when a camera's color range extends beyond that of the sRGB gamut, gamut mapping is needed to convert colors from outside the sRGB gamut to within the gamut for the purpose of sRGB representation [27, 30, 45]. The entire color which has one of its RGB components deviated from the corresponding overall BTF curve is considered as an outlier. As seen in Fig. 3.2, we can observe the vast majority of outliers in our dataset have high color saturation levels and lie close to the boundary of the sRGB color gamut. This gamut mapping essentially produces a change in color for points outside the sRGB gamut, and if out-of-gamut colors are shifted in different ways between different exposures, the color transformation becomes different ($\mathbf{t}_c \neq \mathbf{t}'_c$ in Eq. 3.2) between the two images. Thus, these points become outliers positioned off from the BTF. This effect of gamut mapping becomes more significant with larger exposure ratios, since

[1]We use shutter speed to control exposure because changing the aperture could result in spatial variation of irradiance due to vignetting and depth-of-focus.

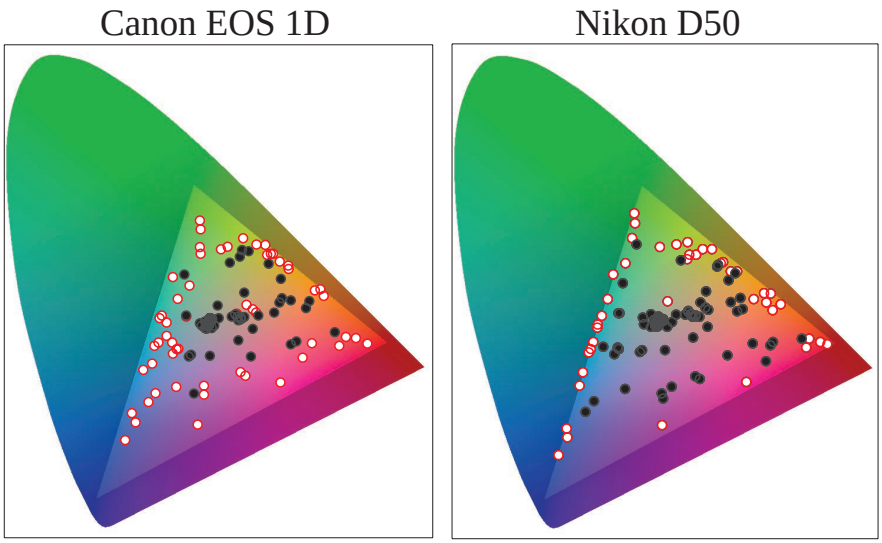


Figure 3.2: Positions of color points in the sRGB chromaticity gamut. Inliers (filled with black) are surrounded by outliers (filled with white). Outliers (as observed in Fig. 3.1) are color points with high saturation levels, and lie close to the boundary of the sRGB gamut.

the out-of-gamut colors need a greater displacement in color space to move into the sRGB gamut.

To summarize, the observations imply that factors such as shutter speed inaccuracies and gamut mapping have to be considered in order to compute the radiometric response function accurately. Most importantly, the observations show that less saturated colors can be modeled with the conventional radiometric model (Eq. 2.18) and be linearized accurately. This confirms that the response is fixed for a certain camera model. However, we have shown that the conventional model has an essential limitation in representing the nonlinear color mapping in the imaging pipeline and highly saturated colors will not be linearized accurately with the model in Eq. 2.18 by inverting f. We address this problem in the next chapter by proposing our new model.

Chapter 4

New in-camera imaging model

In the previous chapter, we examine the in-camera image processing through an extensive analysis of an image database collected by capturing images of scenes under different conditions with over 30 commercial cameras. From our analysis, we found a glaring limitation in the conventional imaging model employed to determine the nonlinearities in the imaging pipeline (i.e. radiometric calibration). In particular, the conventional radiometric models assume that the irradiance (RAW) to image intensity (sRGB) transformation is attributed to a single nonlinear tone-mapping step. However, this tone-mapping step alone is inadequate to describe saturated colors. As a result, such color values are often mis-interpreted by the conventional radiometric calibration methods.

In this chapter, we propose our new model and describe how to introduce this gamut mapping step into the imaging pipeline (section 4.1), together with calibration procedures to estimate the associated parameters for a given camera model (section 4.2). This allows us to model the full transformation from RAW to sRGB with much more accuracy than demonstrated by prior radiometric calibration techniques. Results are shown in section 4.3. Section 4.4 concludes this chapter.

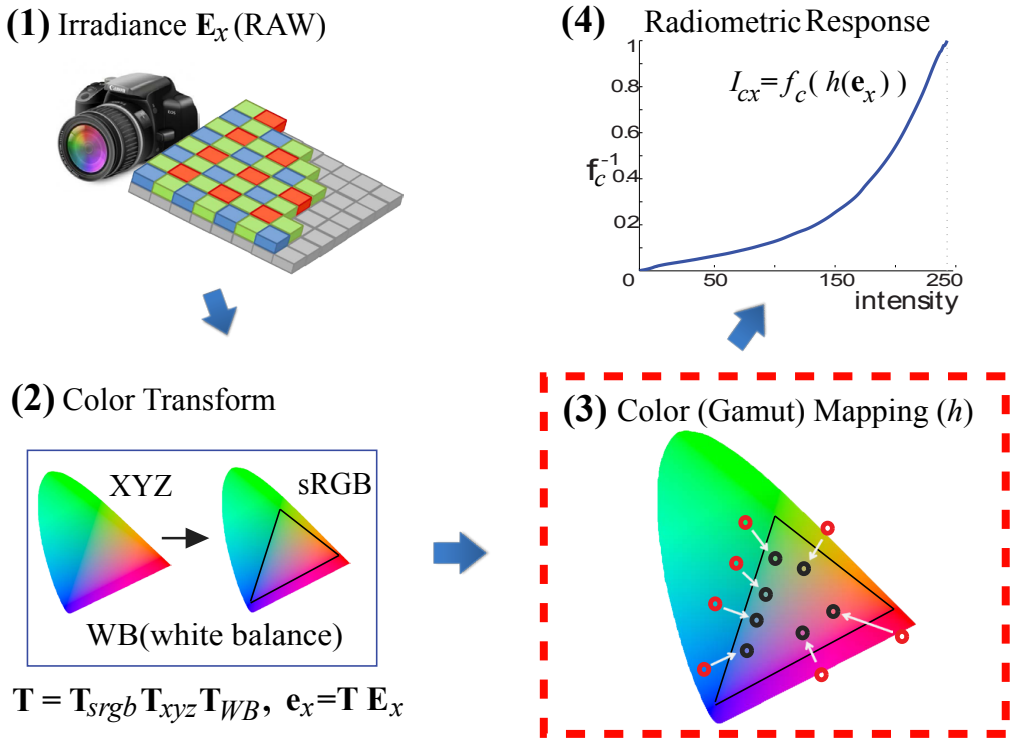


Figure 4.1: A new radiometric model: the color (gamut) mapping process h [30] is added to overcome the limitation of the conventional model.

4.1 Model formulation

Based on our observation, we introduce the following model for the imaging pipeline inside digital cameras, which is illustrated in Fig. 4.1.

$$\begin{aligned} \begin{bmatrix} I_{rx} \\ I_{gx} \\ I_{bx} \end{bmatrix} &= \begin{bmatrix} f_r(e_{rx}) \\ f_g(e_{gx}) \\ f_b(e_{bx}) \end{bmatrix}, \\ \text{where} \begin{bmatrix} e_{rx} \\ e_{gx} \\ e_{bx} \end{bmatrix} &= h(\mathbf{T}_s \mathbf{T}_w \mathbf{E}_x). \end{aligned} \tag{4.1}$$

$\mathbf{E}_x = [E_{rx}, E_{gx}, E_{bx}]^T$ is the irradiance, which can be recorded as a RAW image in certain digital cameras(We assume that the RAW value $\mathbf{E}_x$ is demosaicked (i.e. the color filter array values are interpolated) and is linearly related to the actual irradiance as shown in [7]. The RAW BTFs in Fig. 3.1 also show the linearity of $\mathbf{E}_x$). In our model, the RAW values are first white balanced by a 3×3 diagonal matrix $\mathbf{T}_w$. Then the white balanced RAW values, defined in the camera's color space, are transformed to the linear sRGB space by a 3×3 matrix $\mathbf{T_s}$. Having the linear transformation decomposed to $\mathbf{T}_w$ and $\mathbf{T_s}$ allows more flexibility in designing applications compared to having a single transformation that combines both factors. Notice that the white balance $\mathbf{T}_w$ could actually be applied at a different stage to the same effect, e.g. after the color space transformation $\mathbf{T}_s$ and after the function h in Equation 4.1. We place white balancing as the first operator in the imaging pipeline based on empirical data from our experiments: in all cameras that we tested, the order in Eq. 4.1 yielded the best results. Next, the nonlinear color gamut function h:$\mathbb{R}^3 \rightarrow \mathbb{R}^3$ is applied and then the final image in the nonlinear sRGB space is computed with the camera response function f.

A noticeable difference between the new model in Eq. 4.1 and the conventional model in Eq. 2.18 is the addition of color transformations, especially the nonlinear color gamut mapping function h. In digital cameras, both tone mapping and gamut mapping are employed to transform the colorimetry of the source image to one that produces a visually pleasing image on the actual reproduction medium [27]. The tone mapping by the camera response function f aims to compress the dynamic range

of the luminance recorded from the imaged scene. The gamut mapping (h) acts on the color itself and brings the colors that are outside the sRGB gamut to within the gamut. That is, when a camera's color range extends beyond that of the sRGB gamut, gamut mapping converts colors outside the sRGB gamut to inside the gamut for the purpose of sRGB representation. The gamut mapping process is usually nonlinear with greater transformation of more highly saturated colors near and beyond the boundary of the gamut. In addition to color range compression, gamut mapping may also include different transformations for some specific color ranges, e.g. to make the sky more blue and to make skin tone more vivid. By incorporating this nonlinear color mapping h in the pipeline, our model in Eq. 4.1 describes the in-camera imaging more accurately than the conventional model. Note that in our new model, the color space transformation $\mathbf{T}_s$ is fixed per camera model, the white balance parameter $\mathbf{T}_w$ is fixed per white balance setting of a specific camera, and the response function f and the color mapping h are fixed per picture style of a camera.

4.2 Model calibration based on Radial Basis Functions (RBFs)

Converting a given sRGB image to its RAW representation requires knowledge of the model parameter values in Eq. 4.1, namely of f, $\mathbf{T}_s$, $\mathbf{T}_w$, and h. To calibrate these values, we assume that we are given a number of training images taken by the camera under varied settings with different exposures, white balance, and picture styles. We also assume that the RAW images associated with these training images are provided as well [1]. For each camera model, we compute the color space transform $\mathbf{T}_s$, a matrix $\mathbf{T}_w$ for each white balance preset, and f and h per picture style.

4.2.1 Camera Response Function Estimation

We first compute the camera response function f from a set of images taken with varying exposures. At first glance, using a conventional radiometric calibration procedure does not look feasible due to the presence of h in Eq. 4.1. However, for color points ($\mathbf{p}$) that satisfy $h(\mathbf{p}) = \mathbf{p}$, the following equation holds between a pair of image

[1] For modern DSLR cameras, they can record both RAW and sRGB images per shooting.

intensity values varied by the exposure ratio k':

$$f_c^{-1}(I'_{cx})/f_c^{-1}(I_{cx}) = k', c \in \{r, g, b\}. \tag{4.2}$$

Eq. 4.2 represents the basic principle of traditional radiometric calibration methods, and any of them can be used to compute f. Here, we use the method in [24] which is based on a PCA model of camera responses:

$$g_c(I_{cx}) = \log(f_c^{-1}(I_{cx})) = g_0(I_{cx}) + \sum_{n=1}^{M} \eta_n(I_{cx})\xi_{cn} \tag{4.3}$$

where g_0 is the mean response function, η_n's are the PCA basis functions of the response curve in the log domain and ξ_{cn}'s are the PCA parameters. Given multiple images with different exposures, the response function can be computed linearly by putting the model in Eq. 4.3 into the log version of Eq. 4.2:

$$g_c(I'_{cx}) - g_c(I_{cx}) = K', \tag{4.4}$$

where $g_c = \log f_c^{-1}$ and $K' = \log k'$.

The key is then to find a set of points ($\mathbf{p}$) that satisfy $h(\mathbf{p}) = \mathbf{p}$. In other words, we need to find points that do not get transformed by the gamut mapping function. Since the main purpose of the gamut mapping is to bring the color points outside the gamut into the inside, we assume that colors with low saturation are not significantly transformed by the gamut mapping. Therefore, we only use points with a saturation value (S in HSV color space) below a threshold (β) to compute the response function. Additionally, points with 0 or 255 in any of the color channels are rejected as outliers.

4.2.2 Color Transformation Matrix Estimation

After computing the camera response function f, we can convert the image values to linearized sRGB values. Then the linear color transformation matrices $\mathbf{T}_w$ and $\mathbf{T}_s$ are computed also by using the points with low color saturation that are not affected by the gamut mapping. The white balance matrix $\mathbf{T}_w$ is a diagonal matrix defined per white balance setting, and the color space transformation matrix $\mathbf{T}_s$, which aligns the camera's color space with the sRGB space, is defined per camera. We compute

the $\mathbf{T}_w$'s and $\mathbf{T}_s$ that minimize the following error function:

$$\sum_{i=1}^{M}\sum_{j=1}^{N}\left\|\mathbf{T}_s^{-1}\mathbf{X}_{ij}-\mathbf{T}_{w_i}\mathbf{E}_{ij}\right\|^2, \tag{4.5}$$

where M is the number of white balance settings, N is the number of color points used for estimation, $\mathbf{X}_{ij}$ is the linearized sRGB values computed from the inverse response functions ($\mathbf{X}_{ij}=[f_r^{-1}(I_{r,ij}), f_g^{-1}(I_{g,ij}), f_b^{-1}(I_{b,ij})]^T$), and $\mathbf{E}_{ij}$ denotes the RAW image values ($\mathbf{E}_{ij}=[E_{r,ij}, E_{g,ij}, E_{b,ij}]^T$) that correspond to $\mathbf{X}_{ij}$.

In the minimization Eq. 4.5, the backward space transformation matrix $\mathbf{T}_s^{-1}$ has 9 unknowns and the M white balance diagonal matrices $\mathbf{T}_w$'s have total $3M$ unknowns. We need $N \geq (9+3M)/(3M)$ data points to solve the problem (number 3 in the denominator is because one data point has 3 channels). The forward space transformation $\mathbf{T}_s$ is then calculated as $(\mathbf{T}_s^{-1})^{-1}$. However, before showing the solving method, we have to discuss the scale ambiguity between $\mathbf{T}_w$'s and $\mathbf{T}_s$ first. Assume $\mathbf{T}_w^*$'s and $\mathbf{T}_s^*$ are the ground truth parameters, i.e.

$$(\mathbf{T}_s^*)^{-1}\mathbf{X}_{ij}-\mathbf{T}_{w_i}^*\mathbf{E}_{ij}=\mathbf{0}, \text{for all } i \text{ and } j, \tag{4.6}$$

then for any diagonal 3×3 matrix $\mathbf{R}$, including all zero matrix $\mathbf{R}=\mathbf{0}$, by multiplying $\mathbf{R}$ to both sides of Eq. 4.6, we have $\mathbf{R}(\mathbf{T}_s^*)^{-1}\mathbf{X}_{ij}-\mathbf{R}\mathbf{T}_{w_i}^*\mathbf{E}_{ij}=\mathbf{0}$, which means $\mathbf{T}_s^{-1}=\mathbf{R}(\mathbf{T}_s^*)^{-1}$ and $\mathbf{T}_{w_i}=\mathbf{R}\mathbf{T}_{w_i}^*$ also minimize Eq. 4.5. In order to resolve the $\mathbf{R}$'s scale ambiguity, we fix one of the white balance matrices $\mathbf{T}_{w_{i^o}}=\mathbf{I}$ and solve for the other unknowns. This $\mathbf{T}_{w_{i^o}}$ is selected to correspond to the lighting condition under which the input dataset for this optimization is taken. While the problem of Eq. 4.5 is a quadratic minimization problem, the unknowns can be calculated by setting the first order partial derivative of Eq. 4.5 into zero and solving for the unknowns by matrix calculations.

We note that few camera models such as the Canon EOS-1D and Nikon 200D provide the white balancing scale factors for each channel ($\mathbf{T}_w$) in its image metadata (EXIF). For those cameras, we can compute the color space transformation $\mathbf{T}_s$ just by incorporating $\mathbf{T}_w$ from this metadata into Eq. 4.5.

4.2.3 Color Gamut Mapping Function Estimation

With the camera response function f and the linear color transformations of $\mathbf{T}_w$ and $\mathbf{T}_s$ computed, the last step in our calibration procedure is to solve for the color gamut mapping function h in Eq. 4.1. The gamut mapping function is a key element in defining the color characteristics of a camera. This nonlinear mapping can be vastly different among cameras as shown in Fig. 4.4, making colors in one camera more vivid and colors in another camera look softer. Designing a single parametric model that can describe the gamut mapping functions on different camera models is challenging. We therefore opted for a nonparametric approach to model the gamut mapping function based on scatter point interpolation using radial basis functions (RBFs).

Among several variants of RBFs, we adopt the following form [4, 6] to model the inverse gamut mapping function $h^{-1}(\mathbf{X}) = [h_r^{-1}(\mathbf{X}), h_g^{-1}(\mathbf{X}), h_b^{-1}(\mathbf{X})]^T$:

$$h_c^{-1}(\mathbf{X}) = p_c(\mathbf{X}) + \sum_{i=1}^{N} \lambda_{ci}\phi(\|\mathbf{X} - \mathbf{X}_i\|), c \in \{r, g, b\}, \tag{4.7}$$

where $\mathbf{X} = [f_r^{-1}(I_r), f_g^{-1}(I_g), f_b^{-1}(I_b)]^T$, color points $\mathbf{X}_i$ are the control (or center) points of the RBFs, and N is the number of control points. The λ_{ci}'s are the weights for the basis function ϕ, and we chose $\phi(r) = r$ as the basis function. For the polynomial term $p_c(\mathbf{X})$, we set it as a linear polynomial $p_c(\mathbf{X}) = \mathbf{a}_c^T\tilde{\mathbf{X}}$ where its unknown coefficients $\mathbf{a}_c = [a_{c1}, a_{c2}, a_{c3}, a_{c4}]^T$ and $\tilde{\mathbf{X}} = [1, \mathbf{X}^T]^T$.

Using a gamut mapping based on RBFs is preferable in this problem. RBFs has several advantages over the other scattered point interpolation methods [29], including distance weighted [55, 53] or triangular interpolation methods [17, 47, 46, 15]. These advantages include the following: RBFs accepts irregularly spaced data points as its input, while does not require tessellation or tetrahedrization over the input data points; RBFs extrapolates beyond the range of input data points naturally; RBFs is proven to be the smoothest solution in interpolating the scattered data points [13]; and lastly RBFs do not require the function form of the interpolation surface to be explicitly specified, only the the basis functional ϕ needs to be provided.

With data from the given sRGB-RAW image pairs and the computed matrices $\mathbf{T}_w$ and $\mathbf{T}_s$, the corresponding instance of a control point $\mathbf{X}_i'$ is given by $\mathbf{X}_i' = h^{-1}(\mathbf{X}_i) = \mathbf{T}_s\mathbf{T}_w\mathbf{E}_i$, where $\mathbf{E}_i$ is the RAW value of the control point. Note that all points

regardless of their saturation values are used in this stage in contrast to the previous steps where only points with low saturation were used to compute for f and $\mathbf{T}$'s. From a set of control point pairs $(\mathbf{X}_i, \mathbf{X}'_i)$, the parameters of the RBFs for channel c ($c \in \{r, g, b\}$) in Eq. 4.7, $\boldsymbol{\lambda}_c = [\lambda_{c1}, \lambda_{c2}, ..., \lambda_{cN}]^T$ and $\mathbf{a}_c$, are computed as follows [6]:

$$\begin{pmatrix} \mathbf{D} - 8N\pi\rho\mathbf{I} & \tilde{\mathbf{P}} \\ \tilde{\mathbf{P}}^T & \mathbf{0}_{4\times4} \end{pmatrix} \begin{pmatrix} \boldsymbol{\lambda}_c \\ \mathbf{a}_c \end{pmatrix} = \begin{pmatrix} \mathbf{P}' \\ \mathbf{0}_{4\times3} \end{pmatrix}_c, \tag{4.8}$$

where $\mathbf{D}$ is an $N \times N$ matrix with $\mathbf{D}_{ij} = \|\mathbf{x}_i - \mathbf{x}_j\|$, $\tilde{\mathbf{P}}$ is an $N \times 4$ matrix with the i-th row being $\tilde{\mathbf{X}}_i^T$, and $\mathbf{P}'$ is an $N \times 3$ matrix with the i-th row as $\mathbf{X}_i'^T$. Operator $(\cdot)_c$ extracts the c column of a matrix. The parameter ρ balances smoothness of the RBFs against fidelity to the data. This linear system of format $\mathbf{Ax} = \mathbf{b}$ can be solved directly based on the pseudoinverse: $\mathbf{x} = (\mathbf{A}^T\mathbf{A})^{-1}\mathbf{A}^T\mathbf{b}$ (use command *ldivide* in matlab).

With the computed parameters, the inverse gamut mapping at any point ($h^{-1}(\mathbf{X})$) is evaluated by Eq. 4.7 (Fig. 4.4). The overall performance of the RBF relies on the selection of the control points. While we could use all possible points from the training data as control points, this would be inefficient since the evaluation time grows with the number of control points. Additionally, a larger number of control points could also lead to over-fitting. We instead use a greedy algorithm similar to the one used in [6] to select a small subset of control points from a large number of available points that maintains the desired accuracy. The number of control points used in this work varies from 3000 to 5000. As previously mentioned, the gamut mapping function h is computed per camera picture style and the training data set for each picture style contains images taken from all the white balance settings. Having data from different white balance settings is necessary to have the color points well distributed throughout the color space in the training data. In most of our experiments, we use 70 image pairs per picture style for the training: 7 different white balance settings with 10 RAW-sRGB pairs per each setting.

4.2.4 Calibrating Cameras without RAW support

Thus far, computing the color transformations $\mathbf{T}_w$, $\mathbf{T}_s$, and h relied on having the associated RAW image for each sRGB image in the training set. However, there are

many cameras that do not provide RAW images, especially point-and-shoot cameras. Therefore, a calibration scheme for cameras without RAW support is necessary to broaden the applicability of our work.

For those cameras without RAW support, we use a RAW image of the same scene from another camera as a reference. In this case, Eq. 4.1 changes to:

$$\begin{bmatrix} e_{rx} \\ e_{gx} \\ e_{bx} \end{bmatrix} = h(\mathbf{T}_s \mathbf{T}_w \mathbf{T}_c \mathbf{E'}_x). \tag{4.9}$$

The 3×3 matrix $\mathbf{T}_c$ is a transformation that approximates the transformation between the color space of two different cameras. This linear approximation is justified by the intent of camera manufactures to make the Luther condition satisfied (please refer to Eq. 2.13). $\mathbf{E'_x}$ contains the RAW values given by the reference camera. For cameras without RAW images, the different color transformations are combined into one transformation ($\mathbf{T}_{z_i} = \mathbf{T}_s \mathbf{T}_{w_i} \mathbf{T}_c$), which is computed as the one that minimizes the following error using quadratic programming:

$$\sum_{i=1}^{M} \sum_{j=1}^{N} \left\| \mathbf{T}_{z_i}^{-1} \mathbf{X}_{ij} - \mathbf{E'}_{ij} \right\|^2. \tag{4.10}$$

After computing the $\mathbf{T}_{z_i}$'s, the gamut mapping function h is computed just as explained in Section 4.2.3. While an image of a camera cannot be converted back to its own RAW image with this approach, it can still be transformed to sRGB images with different settings as described in the next section.

4.3 Experimental results

4.3.1 Radiometric Response Function Estimation

We first compare the performance of the response function estimation (Section 4.2.1) against the conventional approach [24] upon which we have built our algorithm.

Fig. 4.2 shows an example of the outliers detected by our algorithm and the response functions recovered by the two methods. The cross-talks are those points with at least one of its RGB channels being 0 or 255. Note that the only difference

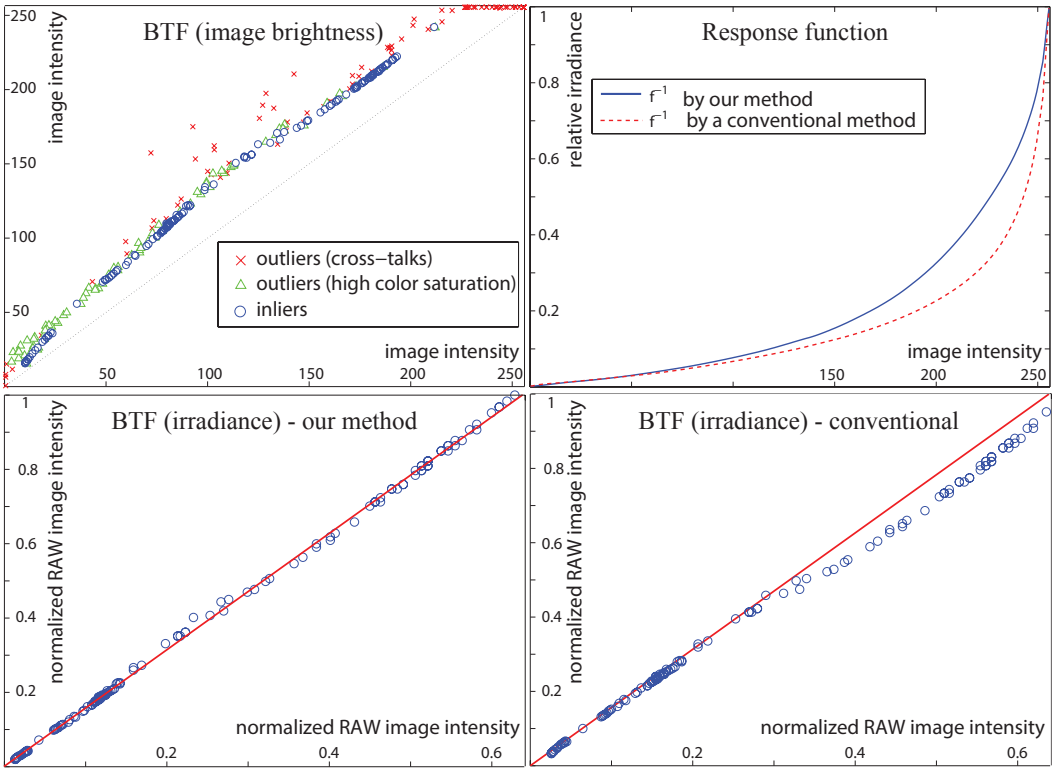


Figure 4.2: A BTF, estimated response function, and linearization results for the blue channel of Nikon D40 using our radiometric calibration algorithm with outliers removed and a conventional method [24]. With our method, the outliers are effectively removed for more accurate calibration.

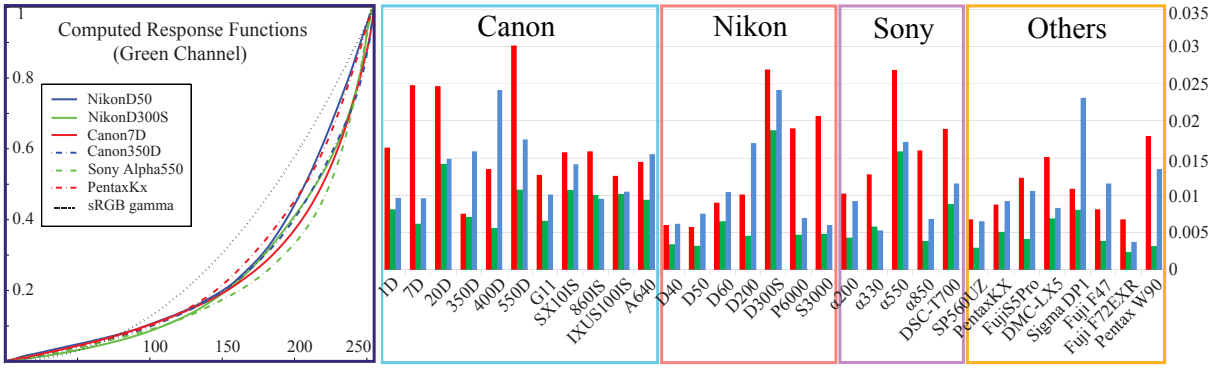


Figure 4.3: Inverse radiometric response functions for a set of cameras in the database and mean linearization errors (δ in Eq. 4.11) for all cameras in the database. High errors indicate the degree of color mapping in the in-camera image processing. Images from cameras with high errors will not be linearized accurately with the conventional radiometric model and calibration, hence a new imaging model is necessary. Using the color mapping (h), the linearization errors converge very close to zero (the maximum among the cameras in the database above is 0.004). Using the The bar colors indicate different color channels.

between the two methods is the existence of the outlier removal procedure. There is a significant difference in the estimations and the proposed algorithm for removing the outliers clearly outperforms on the linearization results.

A few selected inverse response functions computed using our algorithm for some cameras in our database are shown in Fig. 4.3. Note that the responses differ from the sRGB gamma curve commonly used for linearization in some color vision work. For a quantitative evaluation of the response estimation, we use the following measure per channel to gauge the accuracy of linearization from Eq. 2.21:

$$\delta_c = \sqrt{\frac{\sum_{n=1}^{N}\sum_{x\in A}||k'_n f^{-1}(i^n_{cx}) - f^{-1}(i^{n'}_{cx})||^2}{N|A|}}, \tag{4.11}$$

where N is the number of image pairs, A is the set of all image points, and $|A|$ is the size of the set A. To compute δ for each camera, we use all available sets of images in the database for the particular camera, not just the ones used for calibration. This is to verify that a response function computed under a specific condition can be used to accurately linearize images captured under different settings such as the lighting

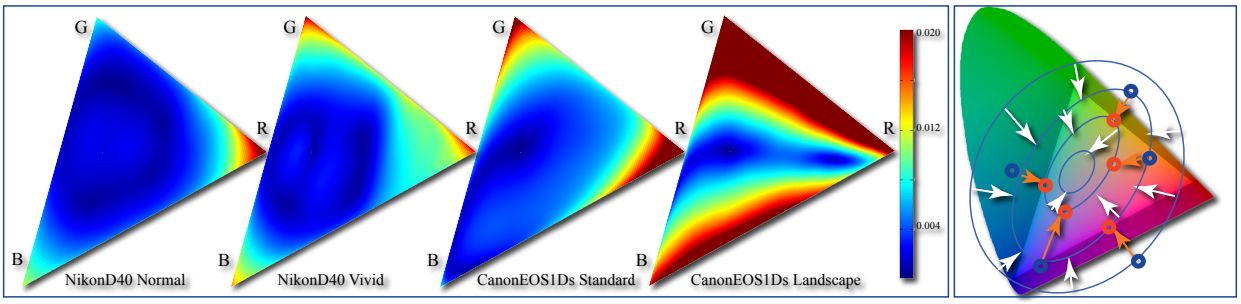


Figure 4.4: (Left) Gamut mapping functions (h) display large variations depending on the mode and the camera. The colors in the map indicate the color displacement magnitude of the gamut mapping at a specific color ($|||[r,g,b]^T - h([r,g,b]^T)||$). Interesting to note is that our estimate of the "Landscape" mode of Canon's picture style matches the description by Canon: "Landscape expresses hues from green to blue more vividly than the Standard settings". (Right) The gamut mapping function is estimated with scatter point interpolation via radial basis functions. The rings and white arrows show the interpolated color mapping function h and the colored dots and arrows indicate control points.

condition and the white balance setting.

Fig. 4.3 (right part) plots the δ's for all cameras in the database. We can see that for many cameras in our database, the image can be linearized very well with an average error of less than 1%. Note that outliers were included for the statistics in Fig. 4.3. If we exclude outliers from the computation, δ converges almost to 0 in many cameras. So the δ in Fig. 4.3 is related to the amount of outliers, or the degree of color mapping h in the in-camera image processing. For the cameras with high δ's, the gamut mapping is applied to points well within the sRGB gamut as opposed to other cameras where it applies only to points close to the boundary of the gamut.

4.3.2 Color Mapping Function Estimation

Next, we evaluate the performance of the color mapping function (h) estimation and the overall accuracy of the new imaging model (Eq. 4.1). The 3D color mapping functions (h) for the Nikon D40 and the Canon EOS-1D are shown as slices in Fig. 4.4. The results confirm the existence of gamut mapping in the in-camera imaging process and the need to include the color mapping function in the radiometric model. The

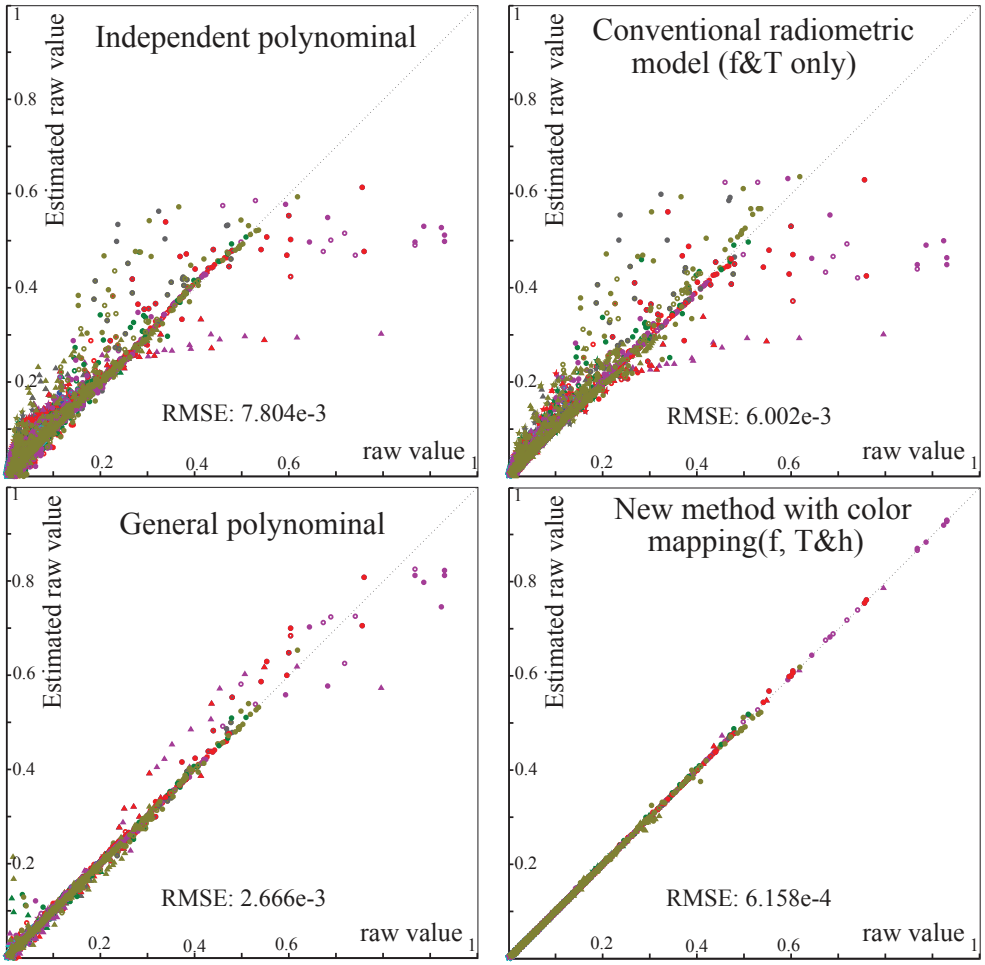


Figure 4.5: Performance of mapping image values to RAW values (Canon EOS-1D) with different techniques: using the technique in [7] with the independent polynomial model per channel, using f and T in Eq. 4.1 without h, the all-channel 3D polynomial model in [7], and the new method with h. Using our new model, images can be mapped back to RAW accurately.

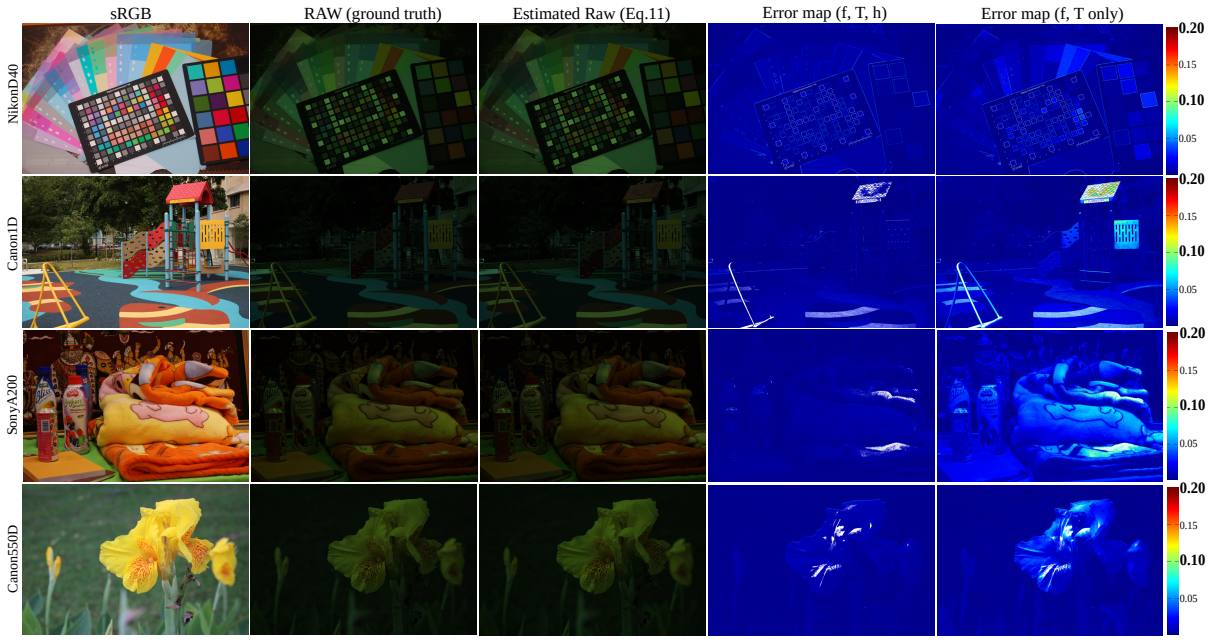


Figure 4.6: Mapping images to RAW. Our method for mapping images to RAWs works well for various cameras and scenes. The white points on the difference maps indicate pixels with a value of 255 in any of the channels which are impossible to linearize. The RMSE's for the new method and the conventional method from the top to the bottom are (0.006, 0.008), (0.004, 0.010), (0.003, 0.017), and (0.003, 0.007). Notice that the errors are high in edges due to demosaicing. For Nikon cameras, the difference in performance between the new method and the conventional method is not as big as other cameras because the effect of the gamut mapping is not as severe as the others (see Fig. 4.5 (a)).

performance of our new imaging model and its calibration procedures for converting image values to RAW is shown in Fig. 4.5. In the figure, we compare the results from four different techniques given a number of sRGB-RAW image pairs. The first method is the implementation of the algorithm from [7] where f is modeled as a 6^{th} order polynomial per channel. The second method computes the RAW just from f and $\mathbf{T}$, which are computed as described in Section 4.2 without the color mapping function h. Next, we computed f as a 3D polynomial function as described in [7]. Finally, the last method computes RAW from Eq. 4.1 with the color mapping function included. As can be seen, the image can be mapped backed to RAW accurately by including the color mapping function in the radiometric model and approximating the mapping function with radial basis functions. In addition, our results show that in-camera color manipulation introduces nonlinearities that cannot be sufficiently modeled by a 3D polynomial function [7].

Fig. 4.6 shows the results of applying the calibrated model to convert images of real scenes back to RAW responses for various scenes and cameras. The estimates of RAW images are compared with the ground truth RAW images. Note that the estimates are purely from the pre-calibrated values of f, h, and $\mathbf{T}$ and the ground truth RAW images are used only for evaluation purposes. Using the new model and the calibration algorithms introduced in Section 4.2, we can accurately convert the image values to RAW values even for the highly saturated colors in the scene.

Our system is implemented in C++ and we have two implementations for evaluating the RBFs gamut mapping function h (Eq. 4.7). One implementation evaluates the RBFs of each image on the fly and takes 15 seconds on average to compute the color transfer, which includes the backward (h^{-1}) RBFs evaluations. The running time of this implementation can be shortened by using a fast RBFs evaluation method as in [22]. The other implementation is based on lookup tables which saves computation time while increasing the amount of memory usage. The color transfer in the experiments depends only on the color values (RGB) of each point in the image and is therefore a deterministic process. This allows lookup tables to be built for both the forward and the inverse process by sampling the RAW and the sRGB color space and precomputing the color transfers for each of the sampled points. With the lookup tables, photo refinishing takes less than a second.

4.4 Conclusion

We have presented our new in-camera imaging model based on an extensive analysis of a large image database. One of the key contributions is identifying the need for a color (gamut) mapping step in the in-camera image processing model. The inclusion of this step covers the limitations present in the conventional imaging model and calibration methods. By considering color mapping in the imaging process, we achieve much higher accuracy than previous approaches in the radiometric response function estimation, and also realize the conversion of a given sRGB image to RAW using our calibration scheme.

For calibration, we relied on a simple assumption that the colors with low saturation are not affected by the gamut mapping. With this assumption, the response function and the linear color space transformations were first computed by using the points filtered by a threhold (β) on the color saturation level. While this simple approach provided satisfactory results for our rather controlled dataset (color charts), a more robust approach based on an iterative scheme may reduce the reliance on a hard threshold for more general databases of images.

Note that the color gamut mapping function h may not be invertible depending on the gamut mapping method employed by the camera. For instance, many color points will be mapped to a single color if a camera employs a clipping strategy. However, we rarely observed such instances in our experiments (about 0.2% of total observations). When such instances occurred, we chose the median value as the control point to approximate the inverse.

While we estimate f and h separately during the calibration, one could also consider combining the two functions into a single $\mathbb{R}^3 \rightarrow \mathbb{R}^3$ function that directly maps white balanced RAW values to nonlinear sRGB values. In principle, the radial basis functions should be able to model this. However, in our experiments we obtained better results when we used two separate functions. Our intuition is that this initial linearization of the RGB space using the function f reduces the complexity of the color mapping function h. This allows h to appear smoother and require less control points for the scatter point interpolation. Modeling f separately from h is also desirable since f can be still computed from multiple images and used for linearization when RAW images are not available.

Recall that the underlying assumption for this work is that cameras are operat-

ing under the photographic reproduction mode, which can be achieved by capturing images in the manual mode and turning off features for scene dependent rendering. We leave the investigation of what and how much scene dependent processing is done in images under the photofinishing mode to the future work. The analysis on the photofinishing mode together with the analysis done in this work will suggest a direction for the internet color vision research [7, 25, 34, 35] in the future.

Chapter 5

Non-uniform lattice regression for in-camera imaging modeling

In the previous chapter, we proposed a new in-camera imaging model that included a 3D color mapping function h based on radial basis functions (RBFs). Using an RBFs model with several of thousand of control points provides a smooth gamut mapping function, however, this design decision incurs a high computational cost. In this chapter, we describe a method to construct a sparse lookup table (LUT) that is effective in modeling the camera imaging pipeline that maps a RAW camera image to its sRGB output or the inverse process. We show how to construct a LUT using a novel nonuniform lattice regression method that adapts the LUT lattice to better fit the underlying 3D function. Our LUT method offers not only a performance speedup of an order of magnitude faster than RBFs, but also a compact mechanism to describe the imaging pipeline.

While describing the method, we take the modeling of the forward imaging pipeline only as the main target. The inverse process could be modeled with an effortless modification, since this LUT method is designed for any 3D function. The remainder of this chapter is organized as follows: an introduction is presented in section 5.1. Section 5.2 gives a brief overview of the lattice regression and related work; Section 5.3 describes our nonuniform regression algorithm; Section 5.4 demonstrates results obtained using our LUT approach followed by the conclusion and discussion in Section 5.5.

5.1 Introduction

As mentioned before, while existing methods were overall effective in modeling the color mapping process, some RGB colors could not be mapped well using the conventional model based on per-channel tone-mapping. Our new model presented in chapter 4 addressed this issue by proposing to add a 3D gamut-mapping function in the imaging pipeline. This new imaging model was shown to be significantly more accurate at modeling the color mapping process than conventional approaches. In addition, the introduction of the color-gamut mapping step made it possible to model different picture styles (e.g. landscape, portrait, vivid, etc). The model is expressed as:

$$\begin{bmatrix} I_{rx} \\ I_{gx} \\ I_{bx} \end{bmatrix} = \begin{bmatrix} f_r(e_{rx}) \\ f_g(e_{gx}) \\ f_b(e_{bx}) \end{bmatrix},$$
$$\text{where} \begin{bmatrix} e_{rx} \\ e_{gx} \\ e_{bx} \end{bmatrix} = h(\mathbf{T}_s \mathbf{T}_w \mathbf{E}_x). \tag{5.1}$$

In this pipeline, the RAW sensor values $\mathbf{E}_x = [E_{rx}, E_{gx}, E_{bx}]^T$ are first white-balanced by a 3×3 diagonal matrix $\mathbf{T}_w$. Then the white-balanced RAW values, defined in the camera's color space, are transformed to the linear sRGB space by a 3×3 matrix $\mathbf{T}_s$. The color mapping function h ($\mathbb{R}^3 \rightarrow \mathbb{R}^3$) is applied afterwards, followed by a final compression by the radiometric response functions $f_c, c \in \{R, G, B\}$. Fig. 5.1 shows a diagram of this pipeline.

Due to the difficulty of using a general model, such as a polynomial, for the color mapping function h, scatter point interpolation via radial basis functions (RBFs) was used to model h. While the RBFs model was shown to be effective, it has a drawback in terms of the computational cost because its evaluation requires computing distances to all control points as illustrated in Fig. 5.1. Although a full-resolution look-up table (LUT) could help in the speed up, this dense LUT required a large memory (over 220MB) and hours to generate.

In this chapter, we propose an improved method to significantly speed up the in-camera color mapping process, i.e. how RAW sensor values are mapped to the

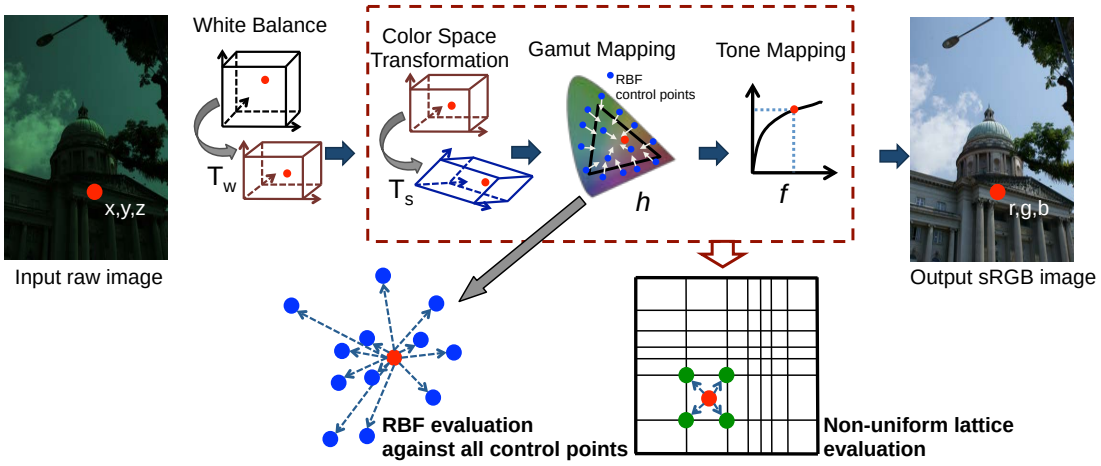


Figure 5.1: The in-camera color processing pipeline. One significant performance bottleneck is the use of RBFs to model the 3D gamut-mapping function. RBFs evaluation requires computing distances to all control points in the RBFs. The contribution of this chapter is to replace several steps in the imaging pipeline with a single sparse LUT based on a nonuniform lattice layout.

standard RGB (sRGB) outputs. Specifically, we introduce a sparse 3D LUT which defines a lattice of control points that are used to interpolate the color mapping. This method requires no distance computation. Instead, it directly indexes the color points to the appropriate LUT cells based on the lattice's structure as illustrated in Fig. 5.1. The general trade-off for using a LUT in lieu of a more complex function is a reduction in accuracy due to the lower-resolution of the control points. To address this issue, we have developed an adaptive lattice regression algorithm that modifies the lattice layout in a nonuniform manner to produce a more accurate estimation of the color mapping function. Moreover, we show that our nonuniform lattice regression method is effective enough to combine the color transformations (T_s) and radiometric functions (f_c) into the LUT. Our nonuniform lattice regression method gives performances comparable to the method based on the RBFs, but requires a fraction of the time to evaluate.

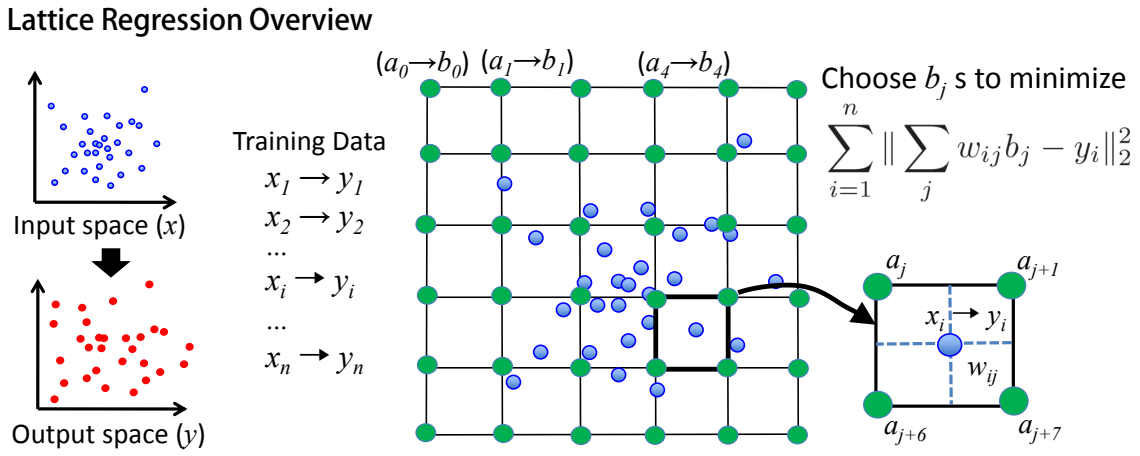


Figure 5.2: An overview of lattice regression using uniform node placement. The lattice a is defined in the input space. Each lattice node, a_i has a corresponding output value b_i. The idea of lattice regression is to compute the control points b_i that interpolate the training-data based on a given interpolation scheme which define the weights w_{ij}.

5.2 Uniform lattice regression

Before describing our algorithm, we introduce the closely related uniform lattice regression in this section. Given a set of RGB colors x in an input color space and a set of points y in the output color space, a LUT is defined over the input color space as a lattice a, which has vertices $\{a_1, a_2, ..., a_m\}$. This effectively divides the input color space into cells with a_i's as the vertices. The corresponding values in the output color space for the lattice a is denoted as $b = \{b_1, b_2, ..., b_m\}$. The LUT is applied in the following manner. An input point x_i finds its corresponding cell in a and then interpolates the output value $\hat{y}_i$. For linear interpolation, which is commonly used, the output estimation is given as $\hat{y}_i = \sum_{j=1}^{m} w_{ij} b_j$, where w_{ij}'s are the interpolation weights satisfying the following: a) $w_{ij} \geq 0$, the equality holds if a_j is not a vertex of the cell containing x_i, b) $\sum_{j=1}^{m} w_{ij} a_j = x_i$ and c) $\sum_{j=1}^{m} w_{ij} = 1$. Fig. 5.2 shows a diagram (drawn in 2D for sake of simplicity) of this type of LUT.

The unknown variables in this approach are the lattice output points b which must be computed based on a set of training sample pairs $(x_i, y_i), i = 1, 2, .., n$. Assuming a fixed space lattice a and linear interpolation function, the output b can be solved

as [20]:

$$\begin{aligned}\hat{b} &= \arg\min_b \sum_{i=1}^{n} (\hat{y}_i - y_i)^2 \\ &= \arg\min_b \sum_{i=1}^{n} \| \sum_j w_{ij} b_j - y_i \|_2^2 \\ &= \arg\min_b \|Wb - y\|_2^2. \end{aligned} \tag{5.2}$$

where W is the weighting matrix with its i-th row being $[w_{i1}, w_{i2}, ..., w_{im}]$ and $y = [y_1, y_2, ..., y_n]^T$.

This basic approach assumes that the training set is well distributed over the input space. Garcia and Gupta [18, 19] suggested adding a smoothness regularization on b to achieve a more natural extrapolation when some cells contained limited number of training samples.

Their proposed objective becomes [19]:

$$\hat{b} = \arg\min_b \|Wb - y\|_2^2 + \lambda J_K(b), \tag{5.3}$$

where $J_K(b) = b^T K b$ is the thin-plate regularizer with a $m \times m$ matrix K that depends only on the type of interpolation basis function and lattice dimensions, and λ is a weighting scalar.

The easiest design is to distribute a_i's equally in each dimension of the input color space. While distributing a_i equally enables a simple array-style access to the cell needed to perform interpolation, it may waste vertex quota in the region where the transformation is quite flat or allocates too few vertices in regions where the transformation is complicated.

Therefore, nonuniform sampling strategies have been introduced. Most notable are Chang et al. [8] and Monga et al. [42]. Although totally free placement of the latex vertices enables the possibility of using less vertices, extra time is introduced during evaluation for sub-volume access [54, 43]. Other approaches such as [8] tried to resolve the sub-volume access using constrained vertices placement. However, this approaches required the space transformation function or its high resolution approximation to be known. In our case, only a sparse set of sample points are known. Moreover, none of the above approaches mentioned can be easily regularized, which is important in our

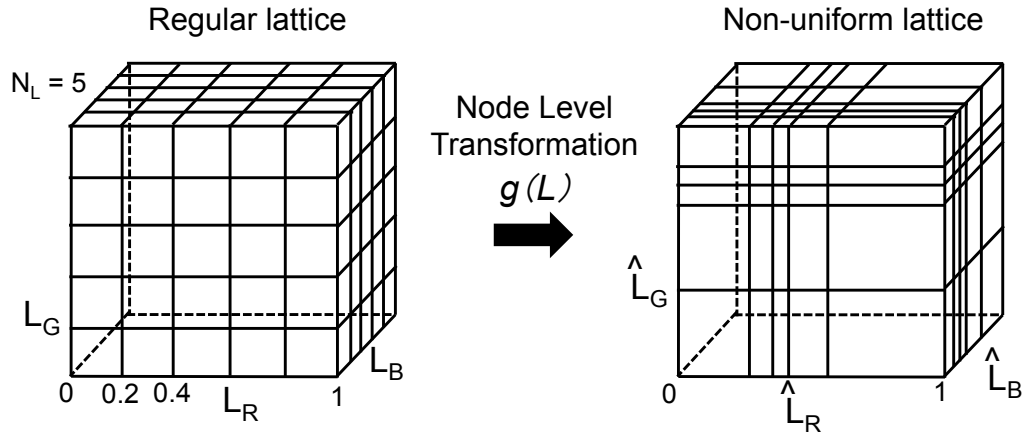


Figure 5.3: Our algorithm computes a node level transformation function that transforms a regular lattice to a non-uniform lattice by moving the node levels along different dimensions.

setting.

We propose our own simple, but effective, nonuniform lattice regression scheme based on the objective of reducing fitting error.

5.3 Model formulation based on non-uniform lattice regression

In our formulation, we use the term, *node level*, to denote how the lattice dimensions are divided given the number of lattice nodes N_L along each dimension. Since the input and output spaces are normalized, the node levels for a given dimension spread out in the range $[0, 1]$. For example, if N_L is 6, the node levels, u_i, for a uniform lattice for a dimension will be $0, 0.2, 0.4, 0.6, 0.8$, and 1. In our approach, N_L is the same for all three color channels., however, the algorithm can be easily extended to have different N_L for different channels.

The crux of our approach is to construct a node level transformation function, g, using the smoothed error histogram of the sample points computed based on the uniform lattice. For dimension c, the node level transformation function g_c transforms an input node level u_i to another level along the lattice dimension, i.e. we are

determining how to adjust the lattice grids to reduce the overall interpolation error of the lattice. In essence, the function g_c maps the uniform lattice node levels to the non-uniform lattice node levels (see Fig. 5.3). The details of our formulation are described in the following.

Uniform Lattice Initialization

Our algorithm first initializes a uniform lattice using the following equation:

$$\hat{b} = \arg\min_b \|Wb - y\|_2^2 + \lambda_S \|Sb\|_2^2 + \lambda_A \rho(b) + \lambda_K J_K(b), \tag{5.4}$$

which is extended from Eq. 5.3 by adding a first-order derivative smoothness regularization $\|Sb\|_2^2$ and a boundary constraint $\rho(b)$. This boundary constraint extension is necessary since in our particular case, data samples are rare in a large portion of the input color space especially near the boundary where the color saturation level is very high. The first-order regularization matrix $S = [S_R; S_G; S_B]$, where each row of $N_L^2(N_L - 2) \times m$ matrix S_c, $c \in \{R, G, B\}$, contains only three nonzero entries. Assuming nodes $a_{i_{1c}}$ and $a_{i_{3c}}$ are neighbors to $a_{i_{2c}}$ along the dimension c, there exists one row in S_c with the nonzero values $\frac{-d(i_{3c},i_{2c})}{d(i_{1c},i_{2c})+d(i_{3c},i_{2c})}$, $\frac{-d(i_{1c},i_{2c})}{d(i_{1c},i_{2c})+d(i_{3c},i_{2c})}$ and 1 at corresponding positions respectively, where $d(i, j)$ is the distance between nodes a_i and a_j. The boundary constraint is $\rho(b) = \sum_{c\in\{R,G,B\}}((b_1)_c^2 + \sum_{i\in\Upsilon_c}((b_i)_c - 1)^2)$, where Υ_c is the set of a_i nodes whose c coordinates equal to 1. Operator $(\cdot)_c$ extracts the c coordinate of a vector or c column of a matrix. Here we also assume that a_1 represents the origin $[0, 0, 0]$. In matrix form, $\rho(b) = \sum_{c\in\{R,G,B\}} \|A_c(b)_c - \xi\|_2^2$, where A_c is composed of standard unit row vectors corresponding to the elements of set Υ_c except the last row being $[1, 0, ..., 0]$ and ξ is vector $[1, ..., 1, 0]^T$. Terms λ_S, λ_A, and λ_K are weighting parameters.

With the notations $\tilde{W}_c = [W; \lambda_S S; \lambda_A A_c]$ and $\tilde{y}_c = [(y)_c; \mathbf{0}; \lambda_A \xi]$, Eq. 5.4, as an quadratic optimization problem, can be solved in closed form per dimension:

$$\hat{b}_c = (\tilde{W}_c^T \tilde{W}_c + \lambda_K K)^{-1} \tilde{W}_c^T \tilde{y}_c. \tag{5.5}$$

The matrix K is only constructed once and is used again when constructing the nonuniform lattice. The matrix K depends only on the lattice size and the interpolation function adopted for the smoothness constraint. We select tricubic interpolation for the smoothness constraint to construct K (see [19] for details).

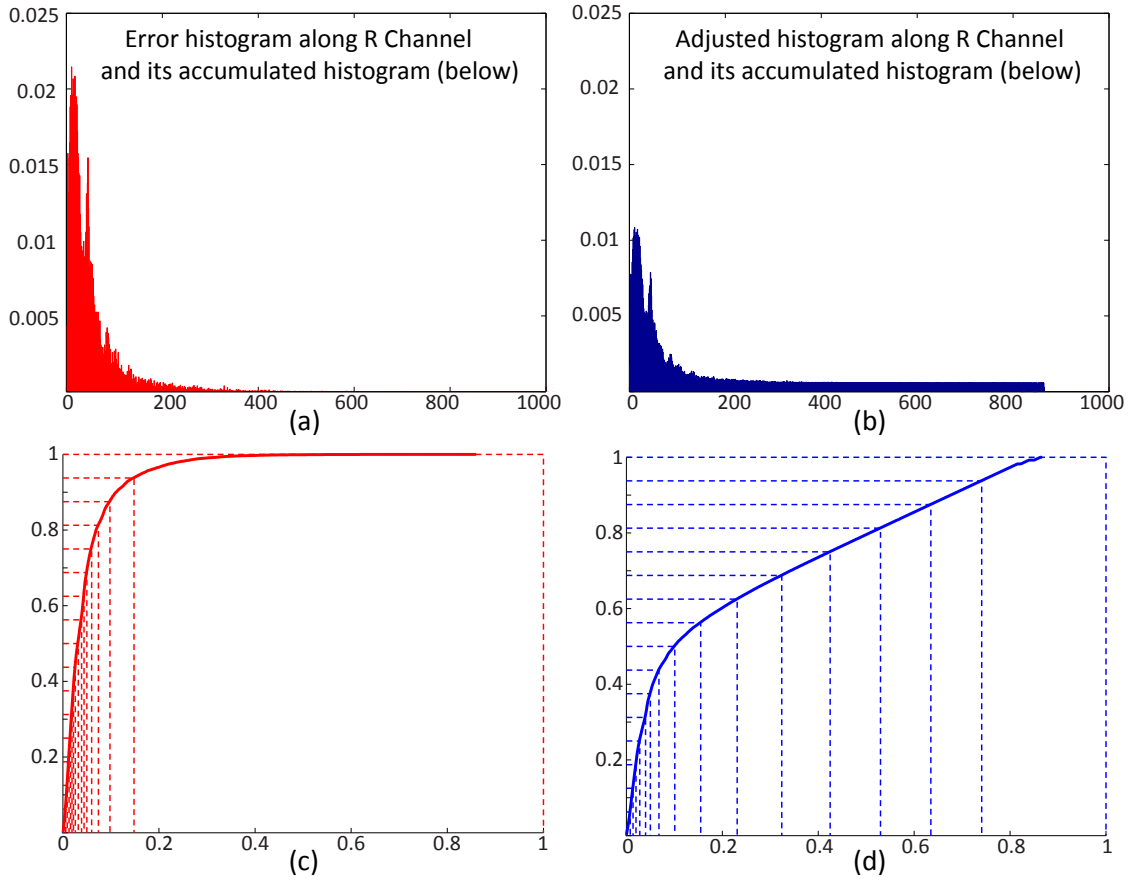


Figure 5.4: Illustration of node level transformation function based on error histogram.

In Eq. 5.5, the weighting scalars $\lambda_S = 0.1$, $\lambda_A = 0.01$, and $\lambda_K = 1e-6$ are fixed in all our computations. We denote this initial uniform lattice as a^o and the corresponding output vertex set as b^o, and W^o is the weighting matrix of x in a^o.

Node Level Transformation Function Formulation

In our formulation, the node level transformation function g_c can be best explained through its inverse function g_c^{-1} that is constructed based on the error histogram computed from the uniform lattice a^o. The idea is that the error histogram indicates the locations where the non-uniformity is most prominent and hence more dense lattice nodes are needed.

From the initial uniform lattice, we can compute each sample error as $e_i^o =$

$\|(W_i^o b^o)^T - y_i\|_2^2$ where W_i^o is the i-th row of W^o. The k-th ($1 \leq k \leq N_{bin}$, where N_{bin} is the number of bins) bin of the normalized error histogram $hist_c$ for channel $c \in \{R, G, B\}$ is computed as:

$$hist_c(k) = \frac{1}{E_c} \cdot \sum_{i \in \Omega_c^k} e_i^o, \tag{5.6}$$

where E_c is the normalization factor such that $\sum_{k=1}^{N_{bin}} hist_c(k) = 1$ and Ω_c^k contains the indexes of all the sample points that are located in the k-th bin for color channel c. In our experiments, we set $N_{bin} = 1000$.

Our approach uses the accumulative error histogram to construct the node level transformation function. We found, however, that using the $hist_c()$ computed above is problematic as many bins are nearly empty (see Fig. 5.4(a)). This results in flat regions in the accumulative error histogram (see Fig. 5.4(c)). To avoid this problem, we use an adjusted version of the error histogram, $hist_c^*$ which is blended with a constant $(1 - \alpha)$ and smoothed by a Gaussian filter (see Eq. 5.7). Fig. 5.4(b) shows the adjusted histogram.

$$hist_c^*(k; \alpha) = \alpha \cdot Gauss(hist_c)(k) + (1 - \alpha)\frac{1}{N_{bin}}. \tag{5.7}$$

For the sake of simplicity, the histogram is treated as a continuous function Θ defined on $[0, 1]$, where $\Theta(t; \alpha) = N_{bin} \cdot hist_c^*(k; \alpha)$ when $t \in [\frac{k}{N_{bin}}, \frac{k+1}{N_{bin}})$. We finally define the inverse of node level transformation function as follows:

$$g_c^{-1}(v; \alpha) = \int_0^v \Theta(t; \alpha) dt, \tag{5.8}$$

where v is a node level. Since Θ incorporates both the error histogram and the uniform term, the derived function g_c^{-1} is able to guide the node level transformation to the erroneous locations while maintaining some nodes in regions of small errors through the control of the uniform term. The monotonically increasing property of the accumulated histogram guarantees the existence of a node level transformation function $g_c(u)$. In our implementation, v can be computed using a binary search over g_c^{-1} for a given u. Fig. 5.4(c) shows the accumulative histogram and the node level transformation functions using the original histogram where α is set to 1. Fig. 5.4(d) shows the accumulative histogram and the node level transformation functions using

the adjusted histogram where α is set to 0.5. We can see the function in (d) has a much slower accumulative error histogram function and the lattice levels are not too skewed by the errors.

Nonuniform Lattice Construction The nonuniform lattice node levels $L_c(\alpha)$ along the dimension $c \in \{R, G, B\}$ can be obtained by transforming uniform node levels using $g_c(u)$. The i-th node level $L_c(i; \alpha)$ is calculated as follows:

$$L_c(i;\alpha) = g_c(\frac{i-1}{N_L - 1}; \alpha), i = 1, 2, ..., N_L. \tag{5.9}$$

Non-uniform lattice $a(\alpha)$ is then constructed as the Cartesian product of $L_R(\alpha)$, $L_G(\alpha)$ and $L_B(\alpha)$, and $b_c(\alpha) = ((\tilde{W}_c^\alpha)^T \tilde{W}_c^\alpha + \lambda K)^{-1} (\tilde{W}_c^\alpha)^T \tilde{y}_c$, where $\tilde{W}_c^\alpha = [W^\alpha; \lambda_S S^\alpha; \lambda_A A_c]$ with W^α representing the weighting matrix of x in $a(\alpha)$ and S^α the new regularization matrix.

While we can set α to a fix value (0.5), we instead let our algorithm choose the best α to blend the nonuniform and uniform arrangements. The value of α can be optimized by solving the following minimization problem using the Trust-Region-Reflective Optimization technique [10, 11]:

$$\hat{\alpha} = \min_\alpha \sum_{i=1}^{n} \|(W_i^\alpha b(\alpha))^T - y_i\|_2^2, \tag{5.10}$$

where W_i^α is the i-th row of W^α. The final lattice is set to be $a(\hat{\alpha})$.

5.4 Experimental results

In this section, we show several comparisons of our nonuniform lattice regression approach in terms of pixel errors and computation time compared to RBFs and uniform lattice regression. All experiments were performed in Matlab and C++ code using a PC with a dual-core 2.3Ghz processor and 3.25GB memory. We used the RAW and sRGB images of MacBeth color charts for different camera models under various settings, which were collected as described in Chapter 3. This calibration data has 164 color samples per image with up to 10 different exposures and a number of white-balance settings per picture style (e.g. landscape, portrait, standard, vivid, etc). Color samples from the RAW images are first multiplied white-balance matrices.

This gives us approximately 10,000 training samples per camera and picture style in the form of $x_i \rightarrow y_i$ as described in Section 5.2.

We apply uniform lattice regression using the method based on Eq. 5.3. Nonuniform lattice regression is computed using the method described in Section 5.3. For each method we compute lattices with resolutions of $13 \times 13 \times 13$, $17 \times 17 \times 17$, and $24 \times 24 \times 24$. The one-off computation time for the uniform lattice regression at the highest resolution is approximately 7s, while our nonuniform lattice regression required approximately 120s. To evaluate the RBFs, we implemented a cached-optimized method that can reuse colors that have already been computed.

Our results are shown in three tables for the following cameras: Nikon D7000 (Table 1), Canon 1Ds Mark III (Table 2), Sony α200 (Table 3). The different techniques are denoted as nonuniform lattice regression (NULR), uniform lattice regression (LR), and RBFs. Each table shows the following normalized pixel error statistics: average error, max error, 25% quartile (Q1) error, 50% quartile (Q2) or median error, and the 75% upper quartile (Q3) error. We also show the running times in seconds (all results computed using C++). The errors are computed on a variety of images with different picture styles, exposure, white-balance settings, and resolutions. Note that Sony α200 has only one picture style available.

We also show qualitative results in Fig. 5.5, which shows the different pixel errors as a hotmap. Our quantitative and qualitative results demonstrate that our nonuniform lattice regression approach provides a performance better than uniform lattice regression and comparable to the results obtained with RBFs. In terms of time complexity, our approach is an order of magnitude faster than using RBFs and comparable to using a uniform lattice with a slight overhead of an indirect lookup table to compensate for the nonuniform layout.

For completeness, we also tested this nonuniform lattice regression method in the inverse process from sRGB to RAW, however, with some modification on the regularization term about boundary constraint in Eq. 5.4. The term λ_A is set to 0, in other words, the boundary constraint is dropped, since the sRGB output may already hit the maximum value before RAW value gets fully saturated. In this comparison, sRGB images from Canon EOS1D MarkIII and Nikon D7000 are transformed back to their RAW images using the different methods. From the error maps, we can see that our nonuniform lattice method produces comparable results as the RBFs method. It is noted that the uniform lattice method also produces good results.

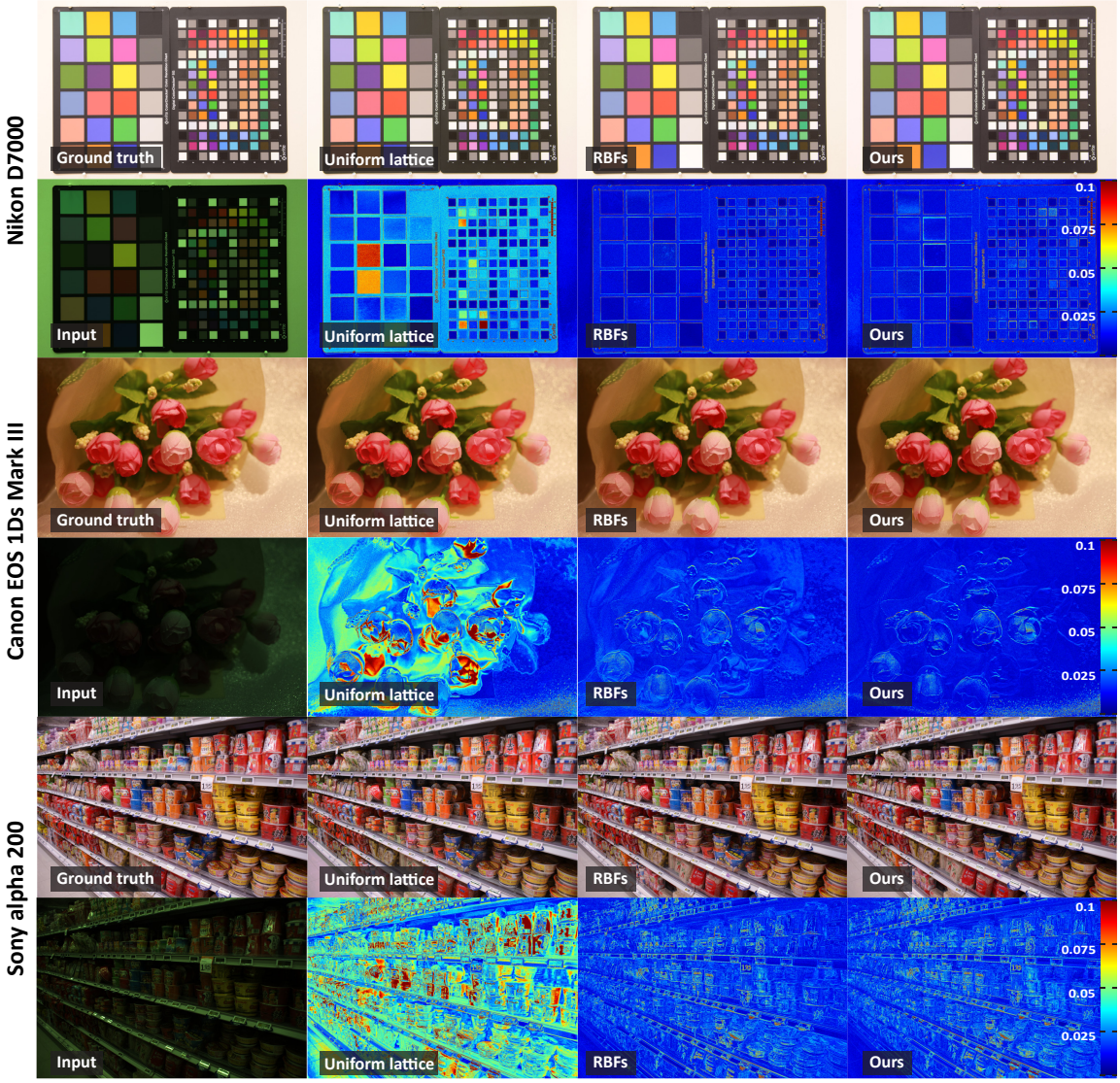


Figure 5.5: Several examples from different camera models showing the error maps between our method, RBFs, and an LUT based on uniform lattice regression [19].

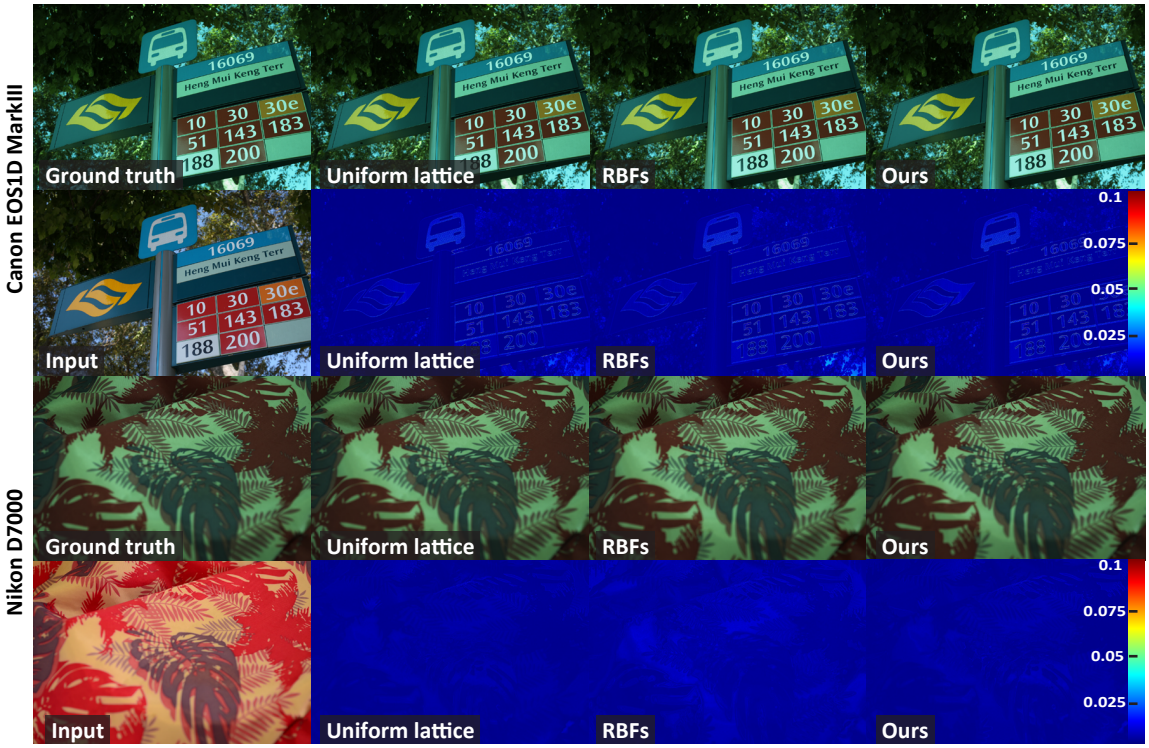


Figure 5.6: Comparing the results of transforming images in sRGB back to their RAW images. The input Canon image is of standard picture style and sunny white balance, while the Nikon input image is of landscape picture style and shade white balance. The error maps between our method, RBFs, and an LUT based on uniform lattice regression [19] are shown, from which we can see that since the input space is not very skewed, the uniform lattice method produces comparable results as our method and RBFs. Noted that the intensities of RAW images are stretched for visualization purpose.

That is because the input sRGB space is not skewed as much as the RAW space. The distinguishable colors are distributed much more evenly in sRGB space than in RAW space. Therefore our nonuniform arrangement of the lattice dose not provide much benefit over a uniform lattice.

NIKON D7000 (size 1632×2464)							
Error (pixel)		Average	Maximum	Q3 75%	Q2 50%	Q1 25%	Time (s)
Method	Lattice Size						
Example 1: Picture Style Normal, Exposure Time 1/15 sec, White Balance Tungsten							
RBFs	-	0.011790	0.060117	0.006792	0.011092	0.015686	24.2028
NULR	$13 \times 13 \times 13$	***0.013215***	***0.061005***	***0.008769***	***0.012401***	***0.017094***	***0.8906***
LR	$13 \times 13 \times 13$	0.058985	0.203733	0.032338	0.052467	0.076546	0.8281
NULR	$17 \times 17 \times 17$	***0.012502***	***0.068935***	***0.007843***	***0.011765***	***0.016169***	***0.9219***
LR	$17 \times 17 \times 17$	0.050673	0.223701	0.025415	0.041130	0.067126	0.8437
NULR	$24 \times 24 \times 24$	***0.012172***	***0.064795***	***0.007843***	***0.011765***	***0.016169***	***0.9844***
LR	$24 \times 24 \times 24$	0.037690	0.154939	0.017971	0.028818	0.047384	0.8437
Example 2: Picture Style Normal, Exposure Time 1/5 sec, White Balance Fluorescent							
RBFs	-	0.009282	0.062868	0.005546	0.008769	0.013006	30.5152
NULR	$13 \times 13 \times 13$	***0.012030***	***0.066667***	***0.006792***	***0.011092***	***0.016169***	***0.8906***
LR	$13 \times 13 \times 13$	0.041590	0.251470	0.019608	0.033735	0.051729	0.8437
NULR	$17 \times 17 \times 17$	***0.011387***	***0.068487***	***0.005546***	***0.009606***	***0.014673***	***0.9062***
LR	$17 \times 17 \times 17$	0.031173	0.211110	0.014673	0.023854	0.039411	0.8437
NULR	$24 \times 24 \times 24$	***0.010408***	***0.086808***	***0.005546***	***0.008769***	***0.013006***	***0.9375***
LR	$24 \times 24 \times 24$	0.022122	0.161405	0.008769	0.016638	0.027730	0.8437
Example 3: Picture Style Landscape, Exposure Time 1/8 sec, White Balance Tungsten							
RBFs	-	0.028220	1.006079	0.016169	0.021118	0.027730	7.3593
NULR	$13 \times 13 \times 13$	***0.023837***	***0.446319***	***0.014673***	***0.020377***	***0.028006***	***0.8750***
LR	$13 \times 13 \times 13$	0.051046	0.460162	0.021479	0.032575	0.077445	0.8281
NULR	$17 \times 17 \times 17$	***0.021774***	***0.456353***	***0.013585***	***0.019212***	***0.025110***	***0.8906***
LR	$17 \times 17 \times 17$	0.041038	0.455088	0.018394	0.028549	0.052320	0.8437
NULR	$24 \times 24 \times 24$	***0.021427***	***0.510241***	***0.013006***	***0.018394***	***0.025110***	***0.9375***
LR	$24 \times 24 \times 24$	0.029896	0.488373	0.016169	0.022866	0.036367	0.8437
Example 4: Picture Style Landscape, Exposure Time 1/15 sec, White Balance Sunny							
RBFs	-	0.021352	1.000653	0.014139	0.019608	0.025415	6.4687
NULR	$13 \times 13 \times 13$	***0.019733***	***0.264437***	***0.012401***	***0.017538***	***0.024174***	***0.8594***
LR	$13 \times 13 \times 13$	0.048376	0.224901	0.022866	0.039992	0.070588	0.8437
NULR	$17 \times 17 \times 17$	***0.018589***	***0.268334***	***0.011765***	***0.016638***	***0.022528***	***0.8906***
LR	$17 \times 17 \times 17$	0.040553	0.215829	0.019212	0.033735	0.057099	0.8281
NULR	$24 \times 24 \times 24$	***0.018108***	***0.283766***	***0.011092***	***0.016169***	***0.021479***	***0.9687***
LR	$24 \times 24 \times 24$	0.029612	0.232434	0.016169	0.025110	0.039411	0.8437

Table 5.1: Normalized pixel errors and evaluation time comparisons of RBFs, uniform lattice regression (LR) and our nonuniform lattice regression (NULR) approach on Nikon examples.

CANON 1Ds MARK III (size 1872×2808)							
Error (pixel)		Average	Maximum	Q3 75%	Q2 50%	Q1 25%	Time (s)
Method	Lattice Size						
Example 1: Picture Style Standard, Exposure Time 1/15 sec, White Balance Cloudy							
RBFs	-	0.009156	0.114332	0.005546	0.008769	0.012401	17.7498
NULR	$13 \times 13 \times 13$	***0.009829***	***0.114332***	***0.005546***	***0.008769***	***0.012401***	***1.0625***
LR	$13 \times 13 \times 13$	0.045459	0.161595	0.026307	0.039411	0.059344	0.9531
NULR	$17 \times 17 \times 17$	***0.009217***	***0.117057***	***0.00554***	***0.008769***	***0.012401***	***1.0937***
LR	$17 \times 17 \times 17$	0.028856	0.126467	0.016169	0.026307	0.038822	0.9687
NULR	$24 \times 24 \times 24$	***0.009011***	***0.114332***	***0.005546***	***0.008769***	***0.012401***	***1.1719***
LR	$24 \times 24 \times 24$	0.018508	0.108819	0.009606	0.016638	0.025110	0.9687
Example 2: Picture Style Standard, Exposure Time 1/10 sec, White Balance Fluorescent							
RBFs	-	0.009908	0.103607	0.005546	0.008769	0.013006	20.2185
NULR	$13 \times 13 \times 13$	***0.010629***	***0.103607***	***0.006792***	***0.009606***	***0.014139***	***1.0625***
LR	$13 \times 13 \times 13$	0.048408	0.199769	0.022866	0.037818	0.059084	0.9531
NULR	$17 \times 17 \times 17$	***0.010329***	***0.105007***	***0.006792***	***0.009606***	***0.014139***	***1.0937***
LR	$17 \times 17 \times 17$	0.035407	0.174146	0.014673	0.026307	0.042418	0.9687
NULR	$24 \times 24 \times 24$	***0.009949***	***0.105007***	***0.005546***	***0.008769***	***0.012401***	***1.1562***
LR	$24 \times 24 \times 24$	0.019140	0.110988	0.011092	0.016638	0.024174	0.9844
Example 3: Picture Style Portrait, Exposure Time 1/15 sec, White Balance Tungsten							
RBFs	-	0.011571	0.175553	0.006792	0.011092	0.014673	22.5049
NULR	$13 \times 13 \times 13$	***0.013200***	***0.170171***	***0.008769***	***0.012401***	***0.017094***	***1.0471***
LR	$13 \times 13 \times 13$	0.060657	0.224422	0.028549	0.045901	0.084017	0.9846
NULR	$17 \times 17 \times 17$	***0.011961***	***0.178421***	***0.007843***	***0.011765***	***0.016169***	***1.1252***
LR	$17 \times 17 \times 17$	0.046053	0.169945	0.023854	0.036367	0.066320	0.9690
NULR	$24 \times 24 \times 24$	***0.011522***	***0.170442***	***0.006792***	***0.011092***	***0.014673***	***1.1721***
LR	$24 \times 24 \times 24$	0.029428	0.137423	0.017094	0.025110	0.034634	0.9690
Example 4: Picture Style Portrait, Exposure Time 1/5 sec, White Balance Fluorescent							
RBFs	-	0.011001	0.215722	0.005546	0.008769	0.014673	29.2521
NULR	$13 \times 13 \times 13$	***0.012381***	***0.152084***	***0.006792***	***0.011765***	***0.016638***	***1.0469***
LR	$13 \times 13 \times 13$	0.037227	0.327774	0.016169	0.027730	0.042599	0.9688
NULR	$17 \times 17 \times 17$	***0.011451***	***0.156224***	***0.005546***	***0.009606***	***0.015686***	***1.0782***
LR	$17 \times 17 \times 17$	0.026838	0.271269	0.009606	0.019608	0.034187	0.9688
NULR	$24 \times 24 \times 24$	***0.011098***	***0.156961***	***0.005546***	***0.009606***	***0.015686***	***1.1407***
LR	$24 \times 24 \times 24$	0.018117	0.168262	0.008769	0.014673	0.023854	0.9844

Table 5.2: Normalized pixel errors and evaluation time comparisons of RBFs, uniform lattice regression (LR) and our nonuniform lattice regression (NULR) approach on Canon examples.

SONY α200 (size 1296×1936)							
Error (pixel)		Average	Maximum	Q3 75%	Q2 50%	Q1 25%	Time (s)
Method	Lattice Size						
Example 1: Picture Style Standard, Exposure Time 1/15 sec, White Balance Sunny							
RBFs	-	0.009144	0.043844	0.005546	0.008769	0.012401	12.9061
NULR	$13 \times 13 \times 13$	***0.009999***	***0.069046***	***0.005546***	***0.008769***	***0.012401***	***0.5156***
LR	$13 \times 13 \times 13$	0.061349	0.165358	0.041687	0.056693	0.077841	0.4531
NULR	$17 \times 17 \times 17$	***0.009889***	***0.080559***	***0.005546***	***0.008769***	***0.012401***	***0.5156***
LR	$17 \times 17 \times 17$	0.049572	0.181497	0.029866	0.043844	0.058298	0.4531
NULR	$24 \times 24 \times 24$	***0.009811***	***0.127013***	***0.005546***	***0.008769***	***0.012401***	***0.5781***
LR	$24 \times 24 \times 24$	0.028771	0.168855	0.016169	0.027730	0.036155	0.4687
Example 2: Picture Style Standard, Exposure Time 1/10 sec, White Balance Fluorescent							
RBFs	-	0.009816	0.063233	0.005546	0.008769	0.013006	13.9998
NULR	$13 \times 13 \times 13$	***0.010954***	***0.065032***	***0.006792***	***0.009606***	***0.014139***	***0.5156***
LR	$13 \times 13 \times 13$	0.053259	0.165498	0.035076	0.050980	0.070806	0.4687
NULR	$17 \times 17 \times 17$	***0.010605***	***0.060117***	***0.006792***	***0.009606***	***0.014139***	***0.5312***
LR	$17 \times 17 \times 17$	0.044052	0.181497	0.022528	0.036996	0.055459	0.4687
NULR	$24 \times 24 \times 24$	***0.010498***	***0.062868***	***0.005546***	***0.009606***	***0.014139***	***0.5625***
LR	$24 \times 24 \times 24$	0.028732	0.144673	0.014139	0.024174	0.035727	0.4844

Table 5.3: Normalized pixel errors and evaluation time comparisons of RBFs, uniform lattice regression (LR) and our nonuniform lattice regression (NULR) approach on Sony examples.

5.5 Conclusion

We have introduced a novel nonuniform lattice regression approach to compute a sparse LUT for use in the camera imaging pipeline. Our approach is based on a regular lattice design but introduces nonuniform spacing of the lattice layout. Our algorithm adaptively adjusts the lattice sampling based on the error histogram to capture the complexity of the 3D transformation functions. Our results demonstrate that this nonuniform lattice provides errors comparable to the scheme that uses RBFs, but with computational efficiency similar to a uniform lattice which is an order of magnitude faster than the optimized RBFs computation. Moreover, the adaptive lattice design allows us to incorporate more steps of the imaging process into the LUT. This advantage comes from the benefit of being a local interpolation method which is more robust in its ability to adapt to the training samples.

Chapter 6

Application: photo refinishing

After the parameters of our in-camera imaging model based on RBFs or nonuniform LUT are calibrated, this model could be used in image editing applications. Through the calibration procedure, we inherently have camera-specific photofinishing information pertaining to different white-balance and picture style settings. In this chapter, we demonstrate how our new imaging pipeline model can be used to develop a system that converts an sRGB input image captured with wrong settings to an sRGB output image that would have been recorded under different and correct camera settings. In essence, our model allows us not only to undo the onboard image processing, but also to refinish an image in a camera-specific manner, producing a result that would appear almost identical to the sRGB output that the camera would have produced with the new settings. For example, given a JPEG image (sRGB) taken with a Canon EOS-1D under a certain white balance and picture style, we can reproduce this photograph as it would appear from the same camera but with a different white balance and picture style. To our knowledge, this is the first system capable of providing such camera-specific refinishing abilities. Moreover, with minor modifications to our approach, we can even allow the user to produce a photograph using another camera's settings.

Fig. 6.1 illustrates the procedure for transferring colors between different settings. Given an input image $\mathbf{I}$ taken under a white balance w_i and a picture style p_i, a new image $\mathbf{I}'$ under a new white balance w_o and a new picture style p_o can be generated

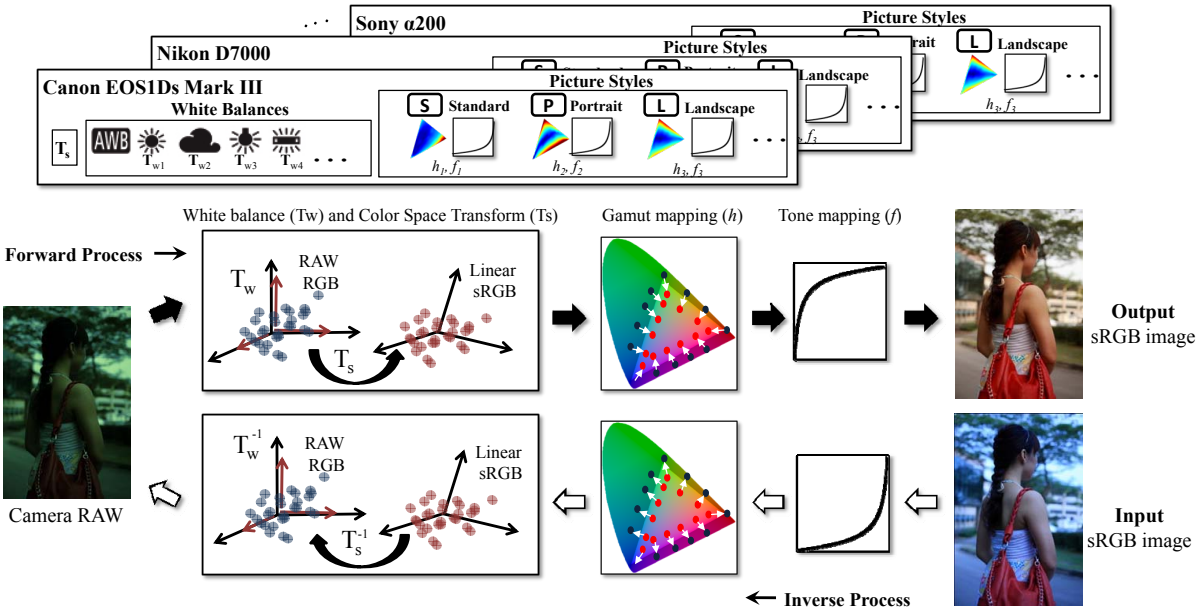


Figure 6.1: Overview of the new imaging model of Eq. 5.1 and its application. The parameters of the imaging process for different cameras and settings including the white balance and the picture style are calibrated using training images. An sRGB image under a certain setting can be transformed to RAW through reverse imaging, and then to another sRGB image under a different setting through forward imaging using the corresponding parameters.

with k times the exposure of the original as follows:

$$\begin{bmatrix} I'_r \\ I'_g \\ I'_b \end{bmatrix} = \begin{bmatrix} f_{r,p_o}(e'_{rx}) \\ f_{g,p_o}(e'_{gx}) \\ f_{b,p_o}(e'_{bx}) \end{bmatrix}, \tag{6.1}$$

$$\text{where} \begin{bmatrix} e'_{rx} \\ e'_{gx} \\ e'_{bx} \end{bmatrix} = h_{p_o} \left(\mathbf{T}_s \mathbf{T}_{w_o} k \mathbf{T}_{w_i}^{-1} \mathbf{T}_s^{-1} h_{p_i}^{-1} \left(\begin{bmatrix} f_{r,p_i}^{-1}(I_{rx}) \\ f_{g,p_i}^{-1}(I_{gx}) \\ f_{b,p_i}^{-1}(I_{bx}) \end{bmatrix} \right) \right).$$

6.1 Manual Mode

Frequently the wrong settings that ruin a photo are manually set by the user, in many cases by mistake. For those images taken under a manual mode, the input settings for white balance (w_i), picture style (p_i), and exposure can all be read from the EXIF data of the input image. The user then only has to select the exposure and the choices for white balance and picture style among the camera presets to correct an image. This correction procedure is intuitive because the user chooses the output settings just as one would do when using a camera.

We also provide a feature which enables the user to change the white balance setting of the output in a continuous manner rather than just selecting from preset options. The white balance parameters (diagonal elements of $\mathbf{T}_w$) are associated with color temperature and thus can be ordered, e.g. tungsten (3200K), fluorescent (4000K), daylight (5200K), and cloudy (6000K). The output white balance parameters ($\mathbf{T}_{w_o}$) in Eq. 6.1 could then be computed by linear interpolation with respect to either a user supplied color temperature or user scrolling between preset white balance settings.

6.2 Auto White Balance Mode

In some cases, one may not like a photograph taken under the camera's auto white balance mode and wish to correct it. The problem with auto-mode images is that it is difficult to recover the specific settings of the camera from the EXIF data. Therefore, we cannot determine which white balance (w_i) and picture style (p_i) to use for Eq. 6.1. For the auto-mode case, we rely on user assistance to convert an image to another

setting. The user can either set the input or the output setting as he wishes and then tune the other settings until he is satisfied with the final output image. The user can choose any of the available picture styles for the camera and change the white balance setting in a continuous manner using interpolation as explained previously.

6.3 Camera-to-Camera Transfer

Thus far, we have described how to transform an image to another image under different settings but from the same camera. We can extend our framework to transfer color between different cameras and their settings. One can imagine such a feature being useful for many applications. For example, it could be used to compare color differences between cameras to guide a person planning to purchase a new camera. It can also be used to align colors of images from different cameras to create seamless mosaics and texture maps.

While the information about sensor spectral sensitivity of the cameras is necessary to accurately compute camera-to-camera color transfers, we approximate the color space transformation between the color spaces of two cameras by a 3×3 matrix $\mathbf{T}_c$. The matrix $\mathbf{T}_c$ is computed using two aligned RAW images ($\mathbf{E}_1$,$\mathbf{E}_2$) of the same scene, one for each camera:

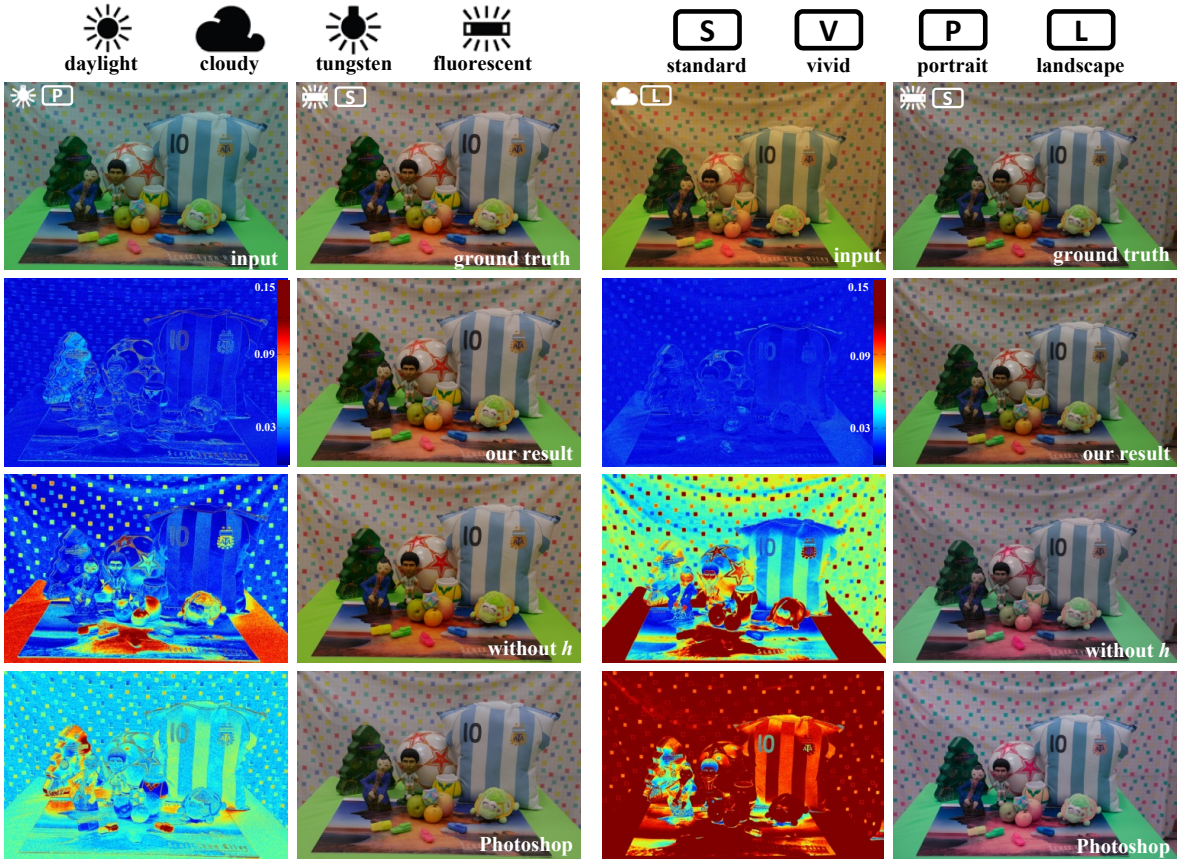
$$\mathbf{T}_c = \underset{\mathbf{T}}{\operatorname{argmin}} \sum_x \|\mathbf{E}_{2x} - \mathbf{T}\mathbf{E}_{1x}\|_2 \tag{6.2}$$

Then the color transfer between cameras is achieved similar to Eq. 6.1:

$$\begin{bmatrix} e'_{rx} \\ e'_{gx} \\ e'_{bx} \end{bmatrix} = h_{p_o}\left(\mathbf{T}_s\mathbf{T}_{w_o}\mathbf{T}_c k \mathbf{T}_{w_i}^{-1}\mathbf{T}_s^{-1}h_{p_i}^{-1}\mathbf{z}\right), \tag{6.3}$$

$$\mathbf{z} = \begin{bmatrix} f_{r,p_i}^{-1}(I_{rx}) \\ f_{g,p_i}^{-1}(I_{gx}) \\ f_{b,p_i}^{-1}(I_{bx}) \end{bmatrix}.$$

The transfer matrix $\mathbf{T}_c$ between two cameras can also be computed via transformations to a reference camera: $\mathbf{T}_{c,1\to3} = \mathbf{T}_{c,1\to2}\mathbf{T}_{c,2\to3}$. Note that $\mathbf{T}_c$ is inherently included in $\mathbf{T}_z$ in Section 4.2.4 and transferring color between cameras that do not support RAW is not a problem.



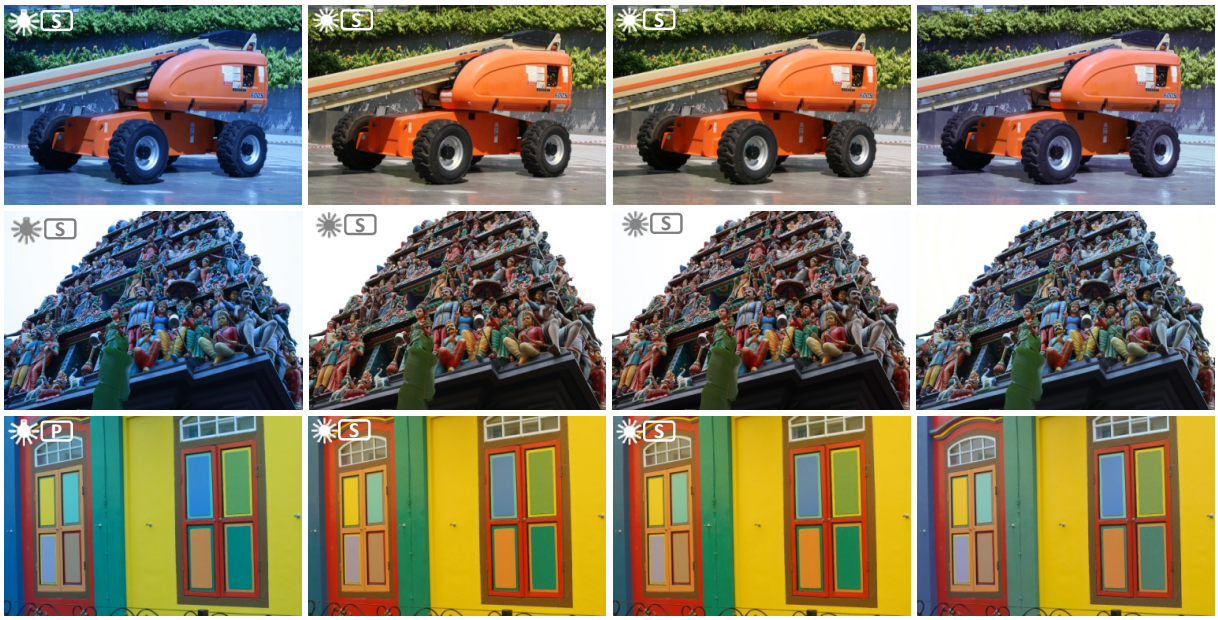
(a) Nikon D7000

(b) Canon EOS 1D

Figure 6.2: Comparisons of different methods for correcting input images taken under inappropriate settings (WB, picture style). Our photo refinishing technique recovers images that are very close to the images from the cameras themselves (ground truth) while the technique without consideration of gamut mapping h and the Photoshop method do not effectively deal with the nonlinearities in the imaging process. The scale for the error maps is the same for all the error maps shown. The RMSE's for the new method, the conventional method, and the Photoshop are (a) (0.02, 0.05, 0.06) and (b) (0.02, 0.1, 0.18).

6.4 Refinishing results

Here we show the ability of our approach to refinish photographs using the extracted parameter settings. For the sake of comparisons, we compare our method based on RBFs with Photoshop and a version of our method based on RBFs without gamut mapping (no h in Eq. 6.1) in Fig. 6.2.



(a) input images (b) our results (c) ground truth (d) Photoshop

Figure 6.3: More examples of our photo refinishing using images from Sony α-200, Canon EOS-1D, and Nikon D200 (from top to bottom). The ground truth images are actual images from the cameras themselves under the proper settings.

For the Photoshop results, we use the Camera RAW utility and choose the best result either from the auto white balancing feature or the semi-auto feature in which we chose a point in the image to be white. As can be seen from the error maps, our photo refinishing technique can transfer colors between different settings accurately, therefore provide a practical method to correct undesired visual errors in photographs taken with the wrong camera settings. Meanwhile, the other two methods have difficulties dealing with the nonlinearities in the imaging process. This is especially visible in Canon's Landscape mode which is shown to have greater nonlinearity in Fig. 4.4. More examples for different cameras are shown in Fig. 6.3, and results

for point-and-shoot cameras that do not support RAW (Section 4.2.4) are shown in Fig. 6.4. Both cameras used for Fig. 6.4 were calibrated using a RAW image from a Canon EOS1-D, and the results closely approximate the ground truth.

We should note that correction using Photoshop can yield visually satisfactory results as can be seen from the second example of Fig. 6.4. However, the quality of Photoshop results is unreliable since it can vary greatly depending on the scene (e.g. distribution of color, especially white and gray colors) and camera settings. Furthermore, we have found that Photoshop almost never reproduces accurate camera specific images as our technique does.

Next, we show examples of transferring colors between cameras. In Fig. 6.5, three images of a scene were taken, each with different cameras, namely Canon EOS-1D, Sony α-200, and Nikon D40. All the cameras were under fluorescent white balance and the standard picture style. Images from these cameras exhibit differences in color, notably the color of the face and the balloon in the middle. The second and the third rows of Fig. 6.5 are the results of transferring camera colors to the Nikon and Sony cameras, respectively. As can be seen, the colors from different cameras can be transferred and matched accurately using our framework (Section 6.3).

As the last example, we show the result of photofinishing an image taken under auto white balance. As mentioned earlier, when a photograph is taken under the auto mode, the input white balance $\mathbf{T}_{wi}$ is unknown and the system relies on user provided information on the unknown parameter. In Fig. 6.6, the user presumes that the image was taken under "daylight" and the system produces images under different white balance settings. In the end, using a slightly warmer color temperature than daylight provides an image more satisfying than the auto white balanced image (Fig. 6.6-d).

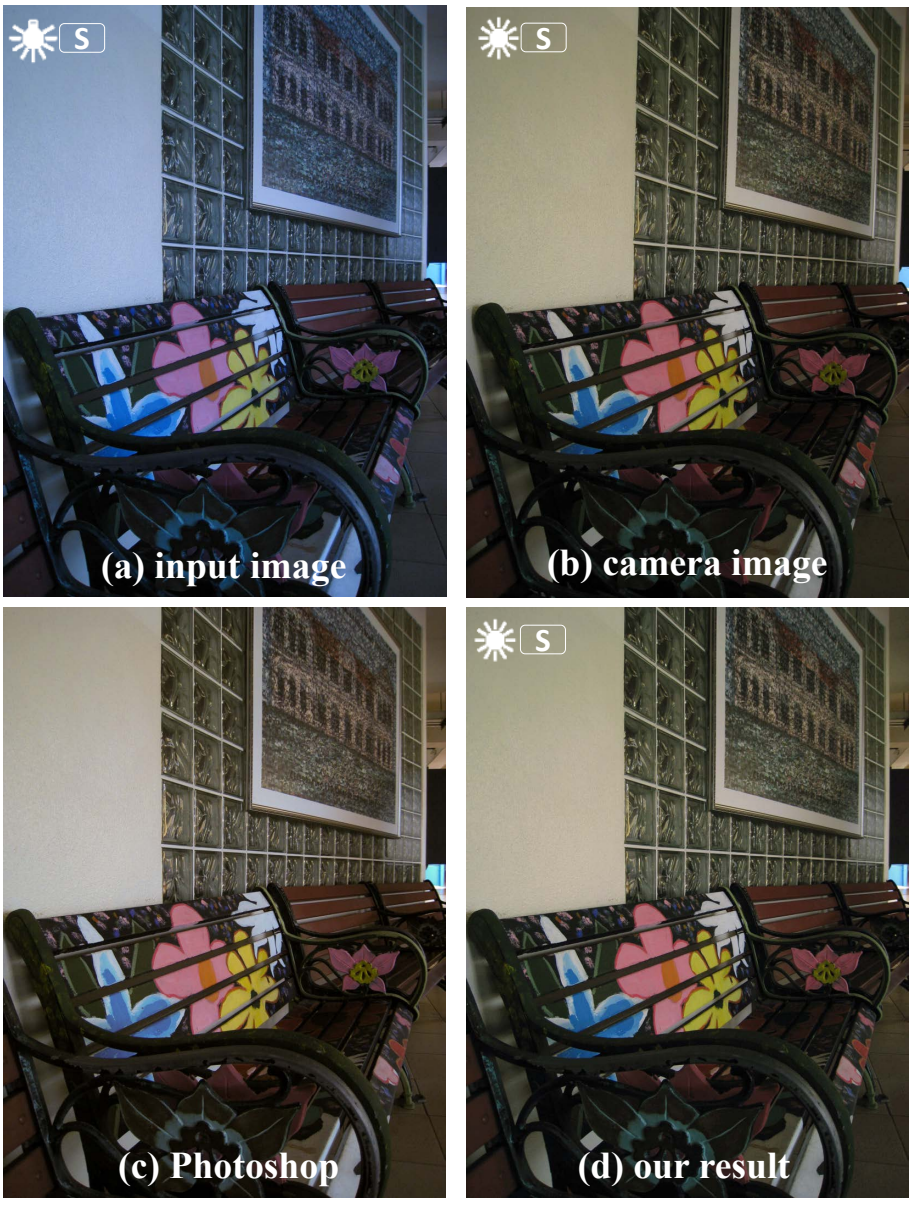
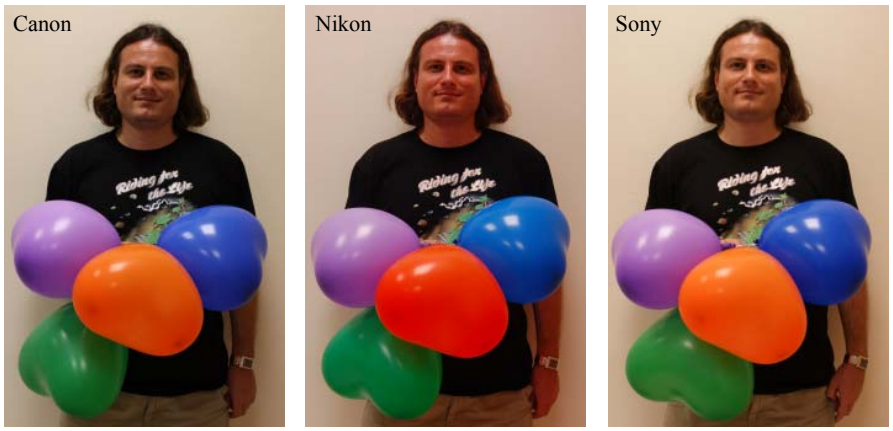


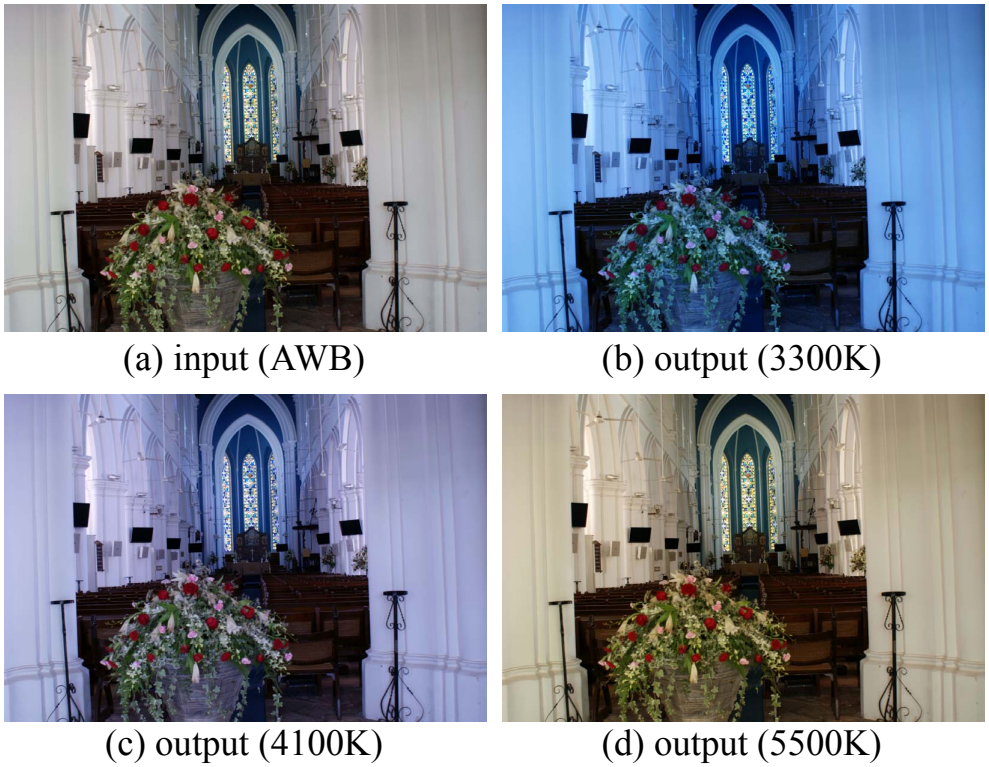
Figure 6.4: Photo refinishing result for a camera (Canon IXUS 860IS) without the RAW support (see Section 4.2.4).



(a) original photographs

(b) color transfers between cameras

Figure 6.5: Transferring colors between cameras. (a) Original photographs from three cameras (Canon EOS-1D, Sony α-200, and Nikon D40) display varying colors. (b) Colors from different cameras can be matched by using the method described in Section 6.3.



(a) input (AWB) (b) output (3300K)

(c) output (4100K) (d) output (5500K)

Figure 6.6: (a) Input image taken under auto white balance. (b)-(d) The user specifies daylight as the input WB and changes the output WB to different color temperatures. Through this process, our system can produce an output image with warmer whites in comparison to the input image.

Chapter 7

Discussions and conclusions

This chapter concludes this thesis by summarizing the work described in previous chapters, including data collection and analysis, our proposed model for in-camera imaging pipeline using RBFs and nonuniform lattice regression, and the photofinishing experiments. This is followed by a short description of possible future directions.

7.1 Summary

The primary objective of this study was to propose a general model of the in-camera imaging pipeline. This proposed model was based on an extensive analysis of a large image database, including thousands of images from more than 30 cameras. Such analysis settled the uncertainty about the response function and led us to realize the significance of the color (gamut) mapping that occurs in the in-camera image processing. Our model formulated the imaging pipeline as steps including white balancing, color space transformation, color mapping and tone mapping. While the other steps are also mentioned in [7], the color mapping step was uniquely proposed in this work. The inclusion of this step covers the limitations present in the conventional imaging model and calibration methods. We also managed to formulate this key step using radial basis functions (RBFs) and non-uniform look up table (LUT). By considering gamut mapping in the imaging process, we can compute not only the radiometric response function more accurately than previous approaches, but can also convert a given sRGB image to RAW using our calibration scheme. Based on the new model, we further introduced a new framework for refinishing photos, which

enables one to correct photographs taken with wrong settings without the associated RAW files. As shown in the result sections of Chapter 4 and 5 and application Chapter 6, the accurate conversion results and the white balance (WB) correction results illustrated the effectiveness and the preciseness of our model.

As one of the most important elements in the in-camera imaging processing model, the response function is the critical yet not well validated part. We conducted a thorough validation over 30 cameras from different camera companies (Chapter 3). From the plotting results, it was shown that the response function is a fixed property for a given camera model in the photographic reproduction mode [27], which was achieved by capturing images in the manual mode and turning off features for scene dependent rendering. However, in the photofinishing mode [27], the intent of which is to produce a pleasing image, the response function varies spatially even within an image, and of course across different images as well. This clearance helps us to define the scope of our study.

In this study, we focused on examining the in-camera imaging pipeline under the photographic reproduction mode. It is understood that any scene dependent tone mapping algorithm [51, 2, 16] could be adopted to achieve spatial varying effects of tone mapping in the photofinishing mode, which brings extra uncertainties and complexities into the imaging pipeline. Focusing on photographic reproduction mode enables us to establish a compact model explaining the core processing of the imaging pipeline.

Since the gamut mapping function is a key element in defining the color characteristics of a camera, every company has its own competing intellectually protected technique. Designing a single parametric model which describes the gamut mapping functions on different camera models is challenging. We first resolved this difficulty by opting for a nonparametric approach and using RBFs to model the mapping function. With this general formulation of the new model, where the gamut mapping function was formulated with RBFs and the response function was formulated based on PCA, we further developed a practical calibration algorithm to learn the parameters of the model. The result section in Chapter 4, where the MSREs of the estimated RAW responses are all below 0.006, showed the accuracy of the model and the effectiveness of the calibration algorithm. Compared to the current calibration techniques, most of which only estimate the response functions channel independently, our inclusion of color mapping function in the imaging pipeline reveals how the cameras process

the RGB colors as a whole. Furthermore, with the color mapping function calibrated for different picture styles and different cameras, researchers are able to examine the differences between the color treatments of different picture styles and cameras. The better understanding through comparison helps to develop more pleasing color processing techniques.

However, RBFs is a global interpolation algorithm, which means the output depends on all of the control points for any input. This results in high computational cost of evaluating the 3D mapping function, since this RBFs approximation contains several thousand control points. In order to reduce the cost, we further proposed a local interpolation method of sparse lookup table that is effective in modeling the 3D mapping function. We also described a novel nonuniform lattice regression method that adapts the LUT lattice structure to better fit the underlying function. Since it is a local interpolation method which is more robust to measurement errors of training samples compared to RBFs, it can be used to model more complex function such as the combination of color mapping and response functions. The results in Chapter 5 shows that our method offers not only a performance speedup by an order of magnitude faster than RBFs, but also a compact mechanism to describe the imaging pipeline.

Although the focus of our nonuniform lattice regression method was the color mapping pipeline in digital cameras, we believe that our approach's ability to simplify complex functions using a nonuniform lattice can be useful in other applications, such as nonlinear image warping or 3D deformable registration. In addition, the structure of the final nonuniform lattice provides insight into the comprehension of the underlying warp, which may be useful for function analysis and/or to help guide better or adaptive calibration methods.

Having the parameters of the proposed model calibrated, we applied our model in the WB correction problem (Chapter 6), which is a practical problem faced by photographers when the photo of the original shooting WB is unsatisfactory. With the flexibilities provided by our model, photographers can adjust the WB of their photos in a more intuitive way. The adjusted results are almost identical to what the camera would have produced while shooting with the selected WBs.

This work, to our best knowledge, is the first to introduce gamut mapping into the imaging pipeline modeling. The proposed model achieves a new level of accuracy in converting sRGB images back to the RAW responses. (These RAW responses

were shown to be linear to the physical scene radiance.) Acting as a fundamental modeling of in-camera imaging pipeline, it has a significant impact on simulating the behavior of a camera and will benefit many computer vision algorithms which rely on the physical scene radiance.

7.2 Future directions

Several future research directions are discussed in the following:

Photofinishing mode investigation While this work targeted modeling the imaging pipeline of photographic reproduction mode, it is worthy to investigate what and how much scene dependent processing is performed in images under the photofinishing mode. The analysis on the photofinishing mode together with the analysis in this work will suggest a direction for the internet color vision research [7, 25, 34, 35] in the future.

Transformation between different RAW spaces In our camera to camera transfer application (see section 6.3), we related two RAW spaces of different cameras with a 3×3 linear transformation. However, this linear transformation is inadequate in compensating the spectral sensitivity differences between the RAW spaces. A more sophisticated model such as piecewise linear model will be preferable. This will be useful in relating images from different cameras.

Identification of source camera from image itself In our work, it is shown that scene colors are rendered differently by different cameras due to their differences in image sensors and color managements. In this way, an image carries some specific properties of its source camera. By differentiating the color rendering results of common scene spectra such as those from blue sky, it may be possible to identify the source camera from the image itself. This is an interesting area for future work.

Bibliography

[1] E. H. Adelson. Checkshadow illusion, 1995. `http://web.mit.edu/persci/people/adelson/checkershadow_illusion.html`; accessed 02-Jul-2012. 2.2.1, 2.3

[2] M. Ashikhmin. A tone mapping algorithm for high contrast images. In *Proc. International Conference Graphics and Interactive Techniques (GRAPHITE)*, 2002. 7.1

[3] G. J. Braun and M. D. Fairchild. Image lightness rescaling using sigmoidal contrast enhancement functions. *Electronic Imaging*, 8(4):380–393, 1999. 2.2.3

[4] M. D. Buhmann. *Radial Basis Functions: Theory and Implementations*. Cambridge University Press, 2003. 1.2, 4.2.3

[5] CANON INC. Canon: picture style, 2011. `http://web.canon.jp/imaging/picturestyle/style/index.html`; accessed 07-Jul-2012. 1.2, 1

[6] J. C. Carr, R. K. Beatson, J. B. Cherrie, T. J. Mitchell, W. R. Fright, and B. C. McCallum. Reconstruction and representation of 3D objects with radial basis functions. In *Proceedings of SIGGRAPH*, pages 67–76, 2001. 1.2, 4.2.3, 4.2.3, 4.2.3

[7] A. Chakrabarti, D. Scharstein, and T. Zickler. An empirical camera model for internet color vision. In *Proc. British Machine Vision Conference*, 2009. 1, 1.1, 2.1, 2.3.1, 2.3.2, 2.3.2, 2.3.2, 2.3.3, 3.1, 4.1, 4.5, 4.3.2, 4.4, 7.1, 7.2

[8] J. Z. Chang, J. P. Allebach, and C. A. Bouman. Sequential linear interpolation of multidimensional functions. *IEEE Transactions on Image Processing*, pages 1231–1245, Sep 1997. 5.2

[9] D. Coffin. Decoding raw digital photos in linux, 1997. `http://www.cybercom.net/~dcoffin/dcraw/`; accessed 13-Jun-2011. 3.1

[10] T. F. Coleman and Y. Li. On the convergence of reflective newton methods for large-scale nonlinear minimization subject to bounds. *Mathematical Programming*, 67(2):189–224, 1994. 5.3

[11] T. F. Coleman and Y. Li. An interior, trust region approach for nonlinear minimization subject to bounds. *SIAM Journal on Optimization*, 6:418–445, 1996. 5.3

[12] P. Debevec and J. Malik. Recovering high dynamic range radiance maps from photographs. In *Proc. ACM SIGGRAPH*, pages 369–378, 1997. 1.1, 2.3.2

[13] J. Duchon. Splines minimizing rotation-invariant semi-norms in sobolev spaces. *Constructive Theory of Functions of Several Variables*, Lecture Notes in Mathematics(571):85–100, 1977. 4.2.3

[14] H. Farid. Blind inverse gamma correction. *IEEE Transactions on Image Processing*, 10(10):1428–1433, 2001. 2.3.2

[15] G. Farin. Triangular bernsteincbézier patches. *Computer Aided Geometric Design*, 3:83–127, 1986. 4.2.3

[16] R. Fattal, D. Lischinski, and M. Werman. Gradient domain high dynamic range compression. *Proc. ACM SIGGRAPH*, pages 249–256, 2002. 7.1

[17] R. Franke and G. M. Nielson. Scattered data interpolation and applications: a tutorial and survey. *Geometric Modelling Methods and Applications*, pages 131–160, 1991. 4.2.3

[18] E. Garcia and M. Gupta. Building accurate and smooth ICC profiles by lattice regression. In *Proc. SID/IS&T Color Imaging Conference*, Nov 2009. 5.2

[19] E. Garcia and M. Gupta. Optimized construction of ICC profiles by lattice regression. In *Proc. SID/IS&T Color Imaging Conference*, Nov 2010. 5.2, 5.3, 5.5, 5.6

[20] E. K. Garcia and M. R. Gupta. Lattice regression. In *Proc. Advances in Neural Information Processing Systems (NIPS)*, pages 1–9, 2009. 5.2

[21] R. C. Gonzales and R. E. Woods, editors. *Digital Image Processing*. Pearson Education, Inc, 2008. 1, 1.1

[22] L. Greengard and V. Rokhlin. A fast algorithm for particle simulations. *J. Comput. Phys.*, 73:325–348, 1987. 4.3.2

[23] M. Grossberg and S. Nayar. Determining the camera response from images: What is knowable? *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(11):1455–1467, 2003. 1.1, 3.2

[24] M. Grossberg and S. Nayar. Modeling the space of camera response functions. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(10):1272–1282, 2004. 1.1, 2.3.1, 2.3.2, 4.2.1, 4.3.1, 4.2, A.3.1

[25] T. Haber, C. Fuchs, P. Bekaert, H.-P. Seidel, M. Goesele, and H. Lensch. Relighting objects from image collections. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8, 2008. 4.4, 7.2

[26] P. D. Hiscocks. Measuring camera shutter speed, 2010. http://www.syscompdesign.com/AppNotes/shutter-cal.pdf. 3.2

[27] J. Holm, I. Tastl, L. Hanlon, and P. Hubel. Color processing for digital photography. In P. Green and L. MacDonald, editors, *Colour Engineering: Achieving Device Independent Colour*, pages 79 – 220. Wiley, 2002. 2.3.3, 3.2, 4.1, 7.1

[28] B. K. P. Horn. Exact reproduction of colored images. *Comput Vis Graph Image Process*, 26:135–167, 1984. 2.2.2

[29] A. Isaac. Scattered data interpolation methods for electronic imaging system: a survey. *Journal of Electronic Imaging*, Lecture Notes in Mathematics(571):85–100, 1977. 4.2.3

[30] ISO 22028-1:2004. Photography and graphic technology - extended colour encodings for digital image storage, manipulation and interchange - Part 1: architecture and requirements, 2004. International Organization for Standardization. 2.3.3, 3.2, 4.1

[31] ITU-R BT.709-5. Parameter values for the HDTV standards for production and international programme exchange, April 2002. International Telecommunication Union. 2.2.2

[32] S. J. Kim, J. M. Frahm, and M. Pollefeys. Radiometric calibration with illumination change for outdoor scene analysis. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8, 2008. 1.1

[33] S. J. Kim and M. Pollefeys. Robust radiometric calibration and vignetting correction. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(4):562–576, 2008. 1.1, 2.3.2, 3.2

[34] S. Kuthirummal, A. Agarwala, D. Goldman, and S. Nayar. Priors for large photo collections and what they reveal about cameras. In *Proc. European Conference on Computer Vision*, pages 74–86, 2008. 1.1, 2.3.2, 4.4, 7.2

[35] J.-F. Lalonde, A. A. Efros, and S. G. Narasimhan. Webcam clip art: Appearance and illuminant transfer from time-lapse sequences. *ACM Transactions on Graphics (Proceedings of SIGGRAPH Asia)*, 28(5):131:1–131:10, 2009. 4.4, 7.2

[36] S. Lin, J. Gu, S. Yamazaki, and H. Shum. Radiometric calibration from a single image. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, pages 938–945, 2004. 1.1, 2.3.2

[37] S. Lin and L. Zhang. Determining the radiometric response function from a single grayscale image. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, pages 66–73, 2005. 1.1, 2.3.1, 2.3.2

[38] R. Luther. Aus dem gebiel der farbreizmetrik. *Z Tech Phys*, 8:540–558, 1927. 2.2.2

[39] C. Manders, C. Aimone, and S. Mann. Camera response function recovery from different illuminations of identical subject matter. In *Proc. International Conference on Image Processing*, pages 24–27, 2004. 1.1

[40] S. Mann and R. Picard. On being 'undigital' with digital cameras: Extending dynamic range by combining differently exposed pictures. In *Proc. IS&T 46th annual conference*, pages 422–428, 1995. 1.1, 2.3.2

[41] T. Mitsunaga and S. Nayar. Radiometric self-calibration. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, pages 374–380, 1999. 2.3.2

[42] V. Monga and R. Bala. Sort-select-damp: An efficient strategy for color look-up-table lattice design. In *Proc. SID/IS&T Color Imaging Conference*, Nov 2008. 5.2

[43] V. Monga and R. Bala. Algorithms for color look-up table design via joint optimization of node locations and ouptput values. In *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing*, Mar 2010. 5.2

[44] J. Morovič. To develop a universal gamut mapping algorithm, 1998. Ph.D. thesis, University of Derby. 2.2.3

[45] J. Morovič and M. R. Luo. The fundamentals of gamut mapping: A survey. *Journal of Imaging Science and Technology*, 45(3):283–290, 2001. 3.2

[46] G. M. Nielson and R. Franke. Surface construction based upon triangulations. *Surfaces in CAGD*, pages 163–177, 1983. 4.2.3

[47] G. M. Nielson, H. Hagen, and H. Müller. *Scientific Visualisation*. IEEE, New York, 1997. 4.2.3

[48] O. Noboru and A. R. Robertson. 3.9: Standard and Supplementary Illuminants. In *Colorimetry*, chapter 3, pages 92–96. Wiley, 2005. 2.2.2

[49] C. Pal, R. Szeliski, M. Uyttendaele, and N. Jojic. Probability models for high dynamic range imaging. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, pages 173–180, 2004. 2.3.2

[50] Panasonic Corporation. Share photos and HD movies with family and friends, 2012. `http://panasonic.net/avc/lumix/compact/zs8_tz18/others.html`; accessed 07-Jul-2012. 1.4

[51] Q. Shan, J.-Y. Jia, and M. S. Brown. Globally optimized linear windowed tone-mapping. *IEEE Transactions on Visulization and Computer Graphics*, 16(4), July-Aug 2010. 7.1

[52] G. Sharma, editor. *Digital Color Imaging Handbook*. CRC Press, 2003. 2.2, 2.2.1, 2.4

[53] R. Sibson. A brief description of natural neighbour interpolation. *Interpreting Multivariable Data*, pages 21–36, 1981. 4.2.3

[54] I. Tastl, J. Recker, Y. Zhang, and G. Beretta. An efficient high quality color transformation. In *Proc. SID/IS&T Color Imaging Conference*, Nov 2009. 5.2

[55] A. J. Worsey and G. Farin. An n-dimensional Clough-Tocher interpolant. *Constructive approximation*, 3:99–110, 1987. 4.2.3

[56] W. D. Wright. A re-determination of the trichromatic coefficients of the spectral colours. *Transactions of the Optical Society*, 30(4):141–164, 1928. 2.2.2

Appendix A

Calibration Interface

A.1 Scope

In this appendix, the calibration interface developed for calibrating color imaging model based on different methods is introduced. This appendix is neither the explanation of the underlying code nor the detailed operation manual. (Interested readers are suggested to go to our project page `http://www.comp.nus.edu.sg/~brown/radiometric_calibration/` for more information.) However, it describes the main functions of our interface.

A.2 User Interface

This section briefly describes the main window of our calibration interface. The main functions of each part of the interface are introduced here. Moreover, the input and output for this software are briefly specified.

A.2.1 Main Window

Fig A.1 shows the main window of our software. This interface is designed for data analysis and manipulation. Therefore, various data visualization modes and convenient interaction ways are available. The layout of this interface can be further introduced in parts as illustrated in Fig. A.2. As the labels overlaid on the top indicate, this main window is composed of several parts including: 1. initialization

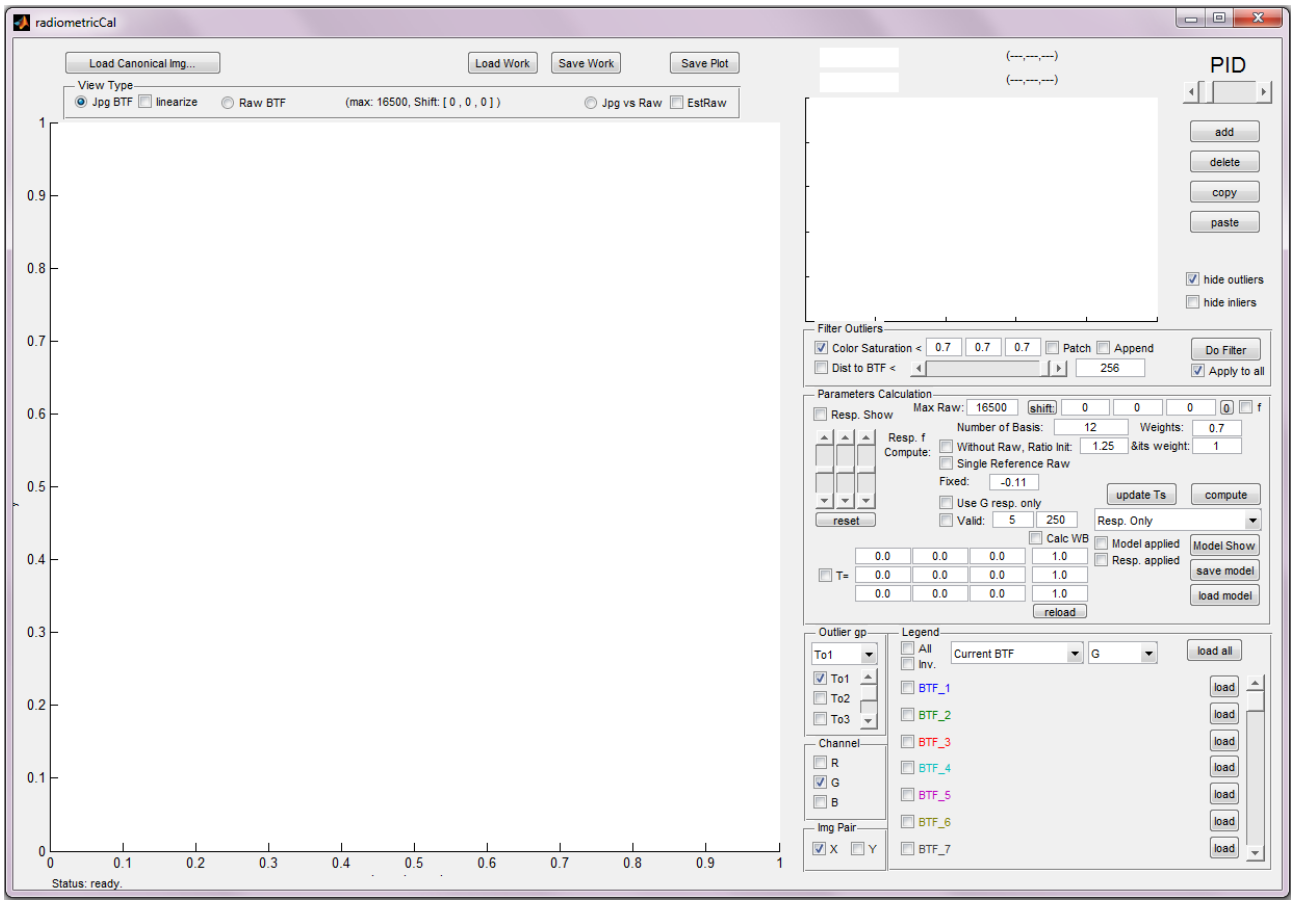


Figure A.1: Main operation window of the interface.

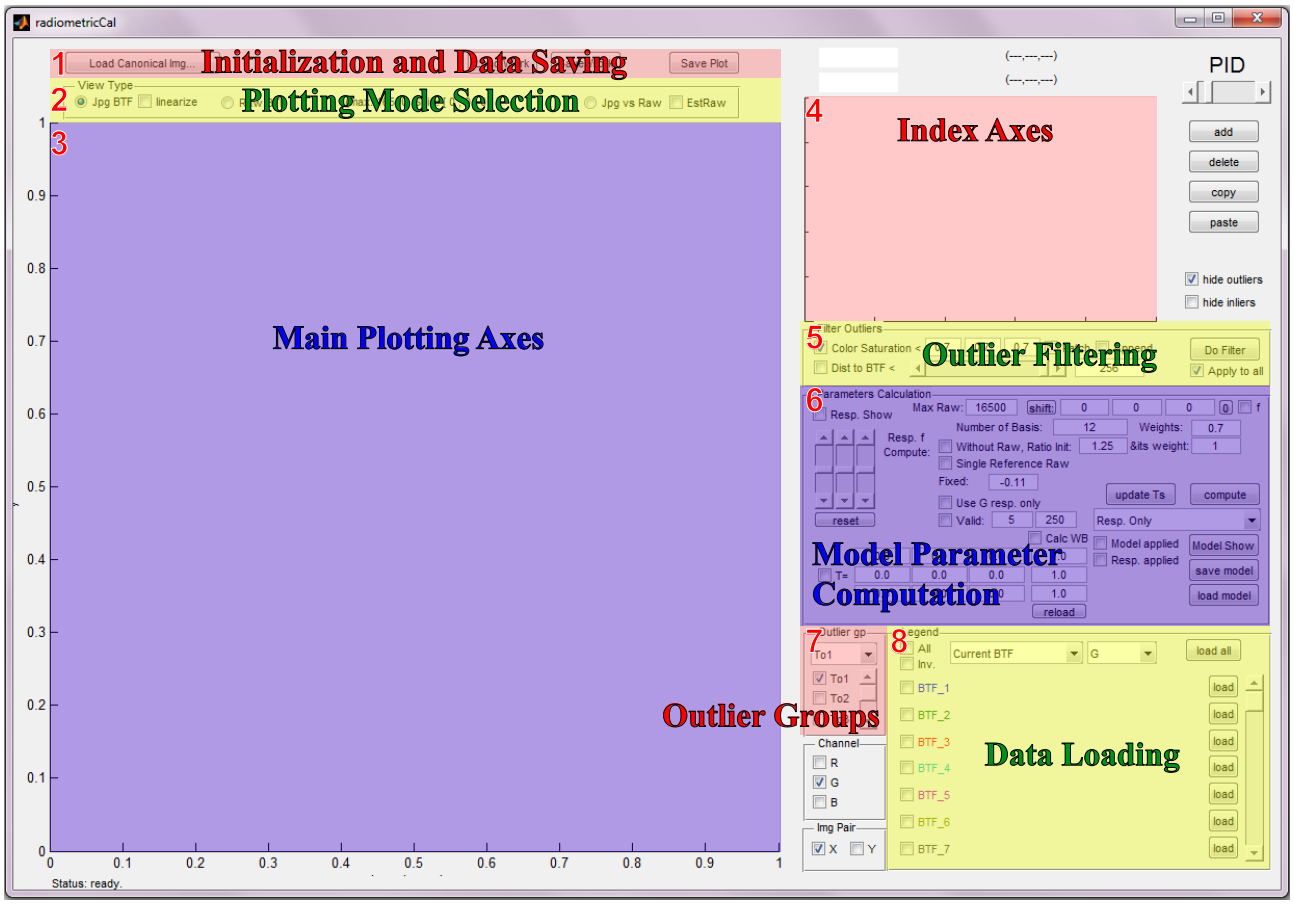


Figure A.2: Labeling of the different parts of the main interface window.

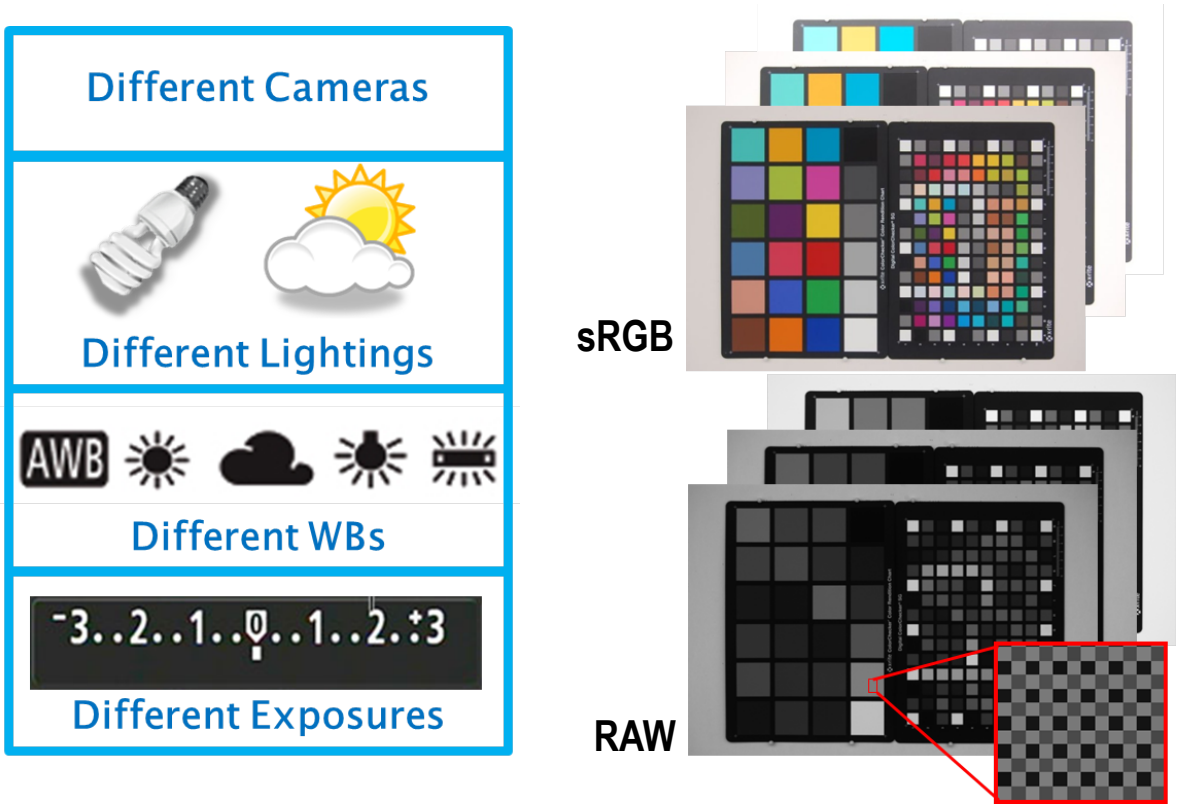
and data saving region, for preparing the initial data input into the software and saving/loading the whole workspace; 2. plotting mode selection region, for changing the plotting mode of the main plotting axes; 3. main plotting axes region, for showing/selecting the data points plotted according to the selected mode and the computation model; 4. index axes region, for showing the index of the input data point or selecting a data point of certain index; 5. outlier filtering region, for filtering the outliers when estimating the response functions; 6. model parameter computation region, for estimating the optimal parameters of different models including RBFs based model and non-uniform lattice regression model; 7. outlier groups region, for selecting the preferred group among different outlier groups gathered by different strategies (e.g. different saturation levels); 8. data loading region; for loading the data into the software for post processing and analysis.

A.2.2 The input and output of the interface

As described in Chapter 3, we collected images of MacBeth color charts using many cameras ranging from DSLR cameras to point-and-shoot cameras. (While our interface supports recovering response functions for those cameras only having sRGB images as their outputs, in this appendix, we focus on the cameras that support RAW outputs, mainly the DSLR cameras only.) Images were taken in manual mode under different settings including white balance, aperture, shutter speed, picture styles, and also under different lighting conditions: indoor lighting and/or outdoor cloudy condition. Both sRGB and RAW images are recorded when possible. Fig A.3 illustrates the data collection settings.

The input data to the software is the sRGB and RAW colors extracted from each sRGB and RAW image pairs. Each patch of the color chart gives one color point after averaging over the patch region. Two sets of sRGB/RAW image pairs with different exposures but same other settings are grouped as one brightness transfer pair with corresponding exposure ratio. This brightness transfer pair is the data unit which will be loaded in to the software.

This interface supports saving the whole workspace as a matlab data file which could be loaded again next time. It is also able to save any plotting in the main plotting axes as a figure. More importantly, the model parameters computed could be saved in text files for further usage, for example, generating the results of real



(a) Capture settings (b) sRGB and RAW image pairs

Figure A.3: Input image data collection. Noted that both sRGB and RAW images are color images, while in RAW format, color channels are arranged according to a certain pattern such as Bayer pattern.

images shown in previous chapters.

A.3 Calibration Procedure

In this section, corresponding to our proposed in-camera imaging pipeline, the main calibration procedure using our interface is described, including response function recovery, white balance and space transformation estimation, different gamut mapping function calibrations based on RBFs and non-uniform lattice regression.

Having loaded the data sets into the interface, (the data sets are from one pic-

Appendix A. Calibration Interface

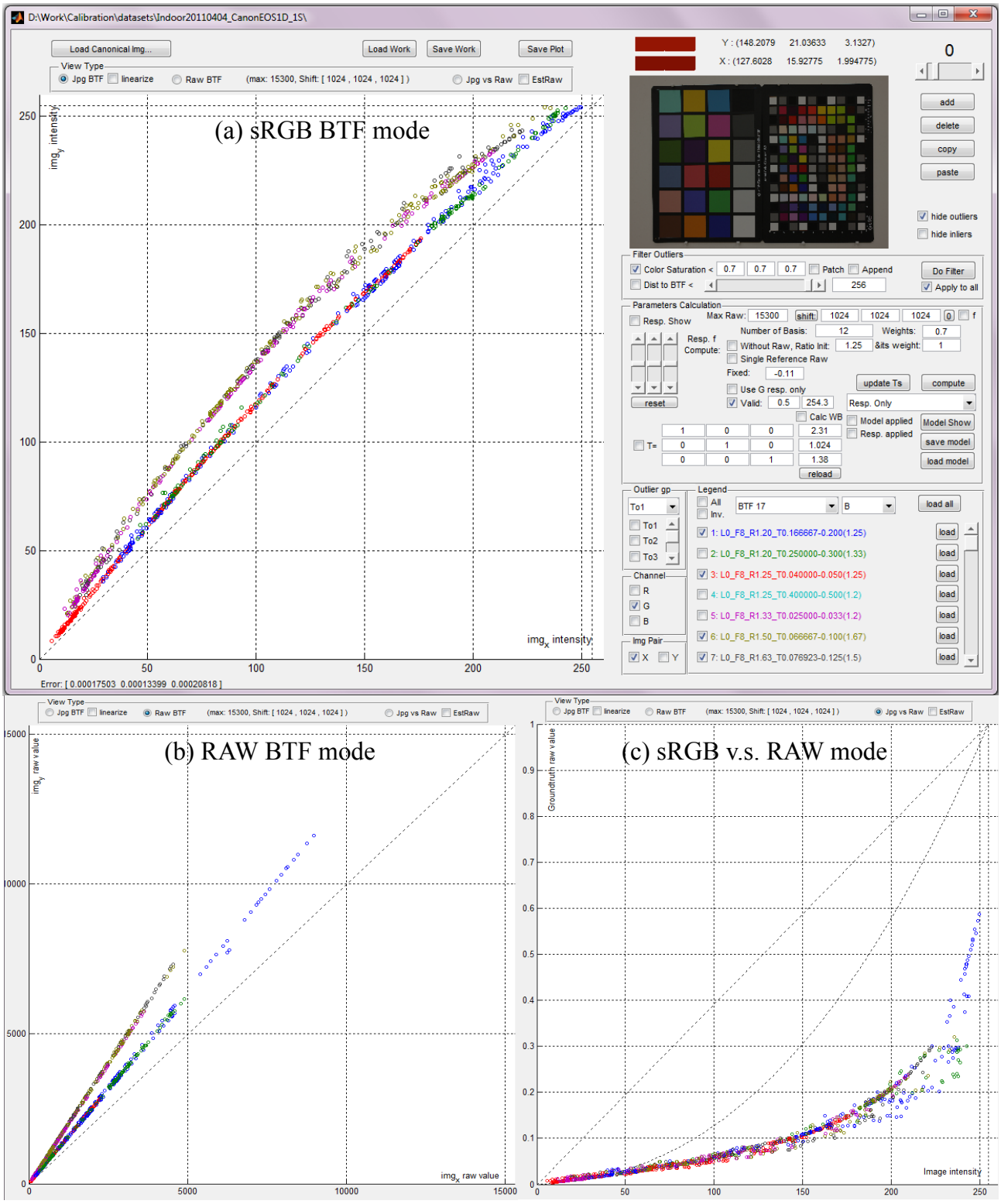


Figure A.4: Snapshots of different plotting modes at data loading step.

ture style but of different white balances in order to calibrate the model parameters for that particular picture style,) we could examine the data in different plotting modes as shown in Fig. A.4. The color of the data point markers corresponds to the color of their corresponding brightness transfer image pair listed in the data loading region. Denoting the i-th brightness transfer sRGB and RAW image pair as $\{I_{ix}, I_{iy}, J_{ix}, J_{iy}\}$, where I, J indicate sRGB image and RAW image respectively, and x indicates the image with shorter exposure and y with longer exposure, the sRGB BTF mode (Fig. A.4(a)) shows I_{ix} v.s. I_{iy}; the RAW BTF mode (Fig. A.4(b)) shows J_{ix} v.s. J_{iy}; and the sRGB v.s. RAW mode (Fig. A.4(c)) shows I_{ix} v.s. J_{ix} and/or I_{iy} v.s. J_{iy} at the data loading step. After model parameters getting estimated, the meanings of the plotting modes might be changed according to the applications of the models.

A.3.1 Response Function Recovery

As described in Chapter 4, the response function recovery method is based on the work [24] which adopted PCA model to capture the space of camera response functions. The parameters such as number of bases and data weighting could be tuned in the model parameter computation region. Before fitting the coefficients of the PCA model, we need to specify a suitable set of image pairs and remove the outliers for response function recovery.

Image pair selection

Not all loaded data points are used in response function recovery, since skewed distribution of data points within the image intensity range will bias the fitting result. What we want is a set of roughly evenly distributed data points. This could be done through selecting several brightness transfer image pairs.

Outlier filtering and refinement

As mentioned in Section 4.2.1, in order to accurately recover the response functions, we use only the points that do not get altered by the gamut mapping function, which are the color points of low color saturation level. The affected color points are considered as outliers in the response function estimation step. From the plot of

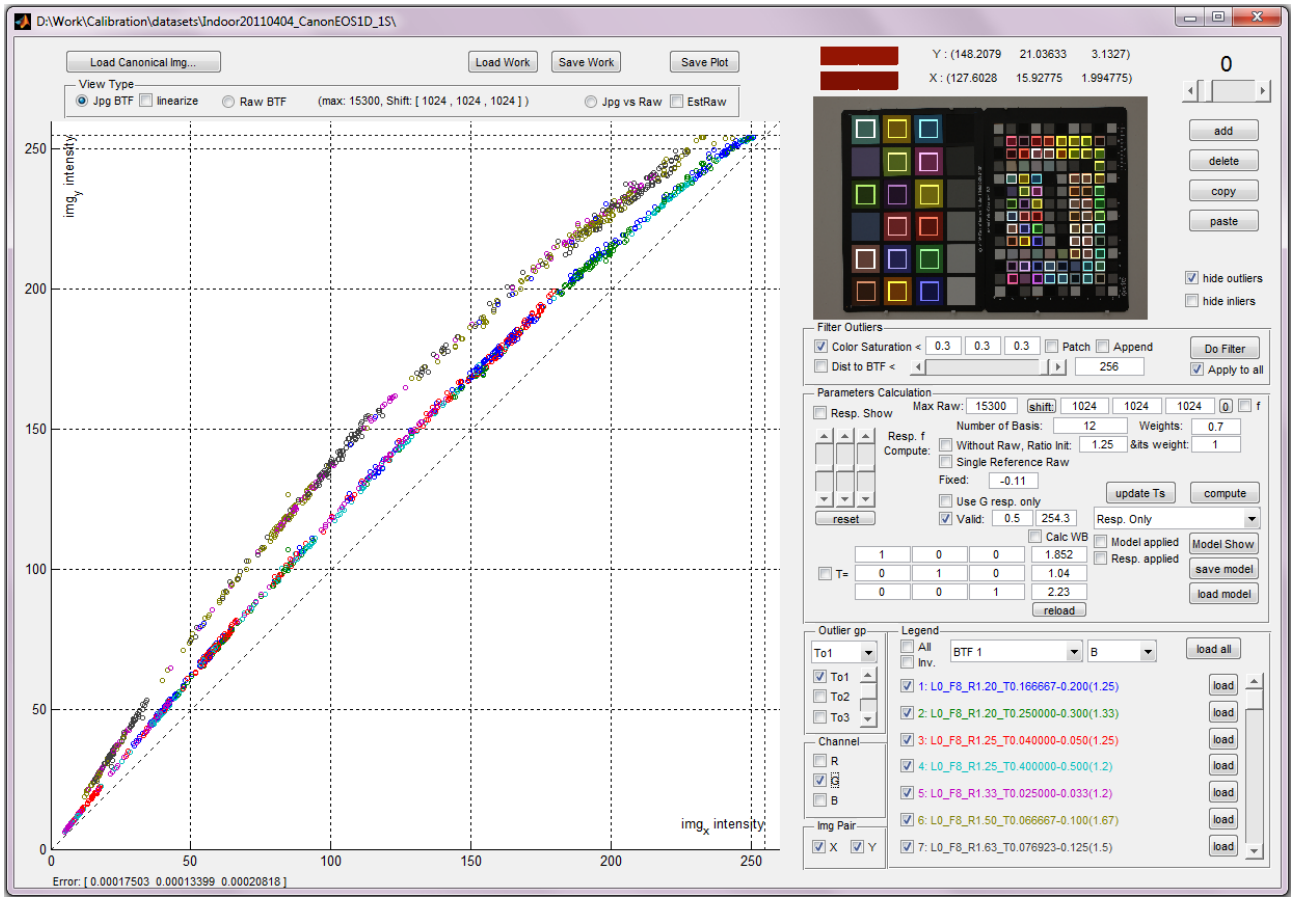


Figure A.5: Outlier filtering for response function estimation.

sRGB BTF mode, those outliers probably fall off their brightness transfer function (BTFs) curves. The majority of the outliers should be those of high saturation level. With the help of the saturation level based filtering tool, we could get much cleaner data as shown in Fig. A.5. The indices of outliers are shown in the index axes, with corresponding patches marked by rectangles.

Further refinement could be done by either manually adding selected points in the main plotting axes through clicking at the corresponding color point and "add" button, or by removing the points certain distance far away from the initial brightness transfer functions computed from response functions which are estimated using the pre-refinement data points.

The interface also provides other views to show the inliers and outliers for analysis

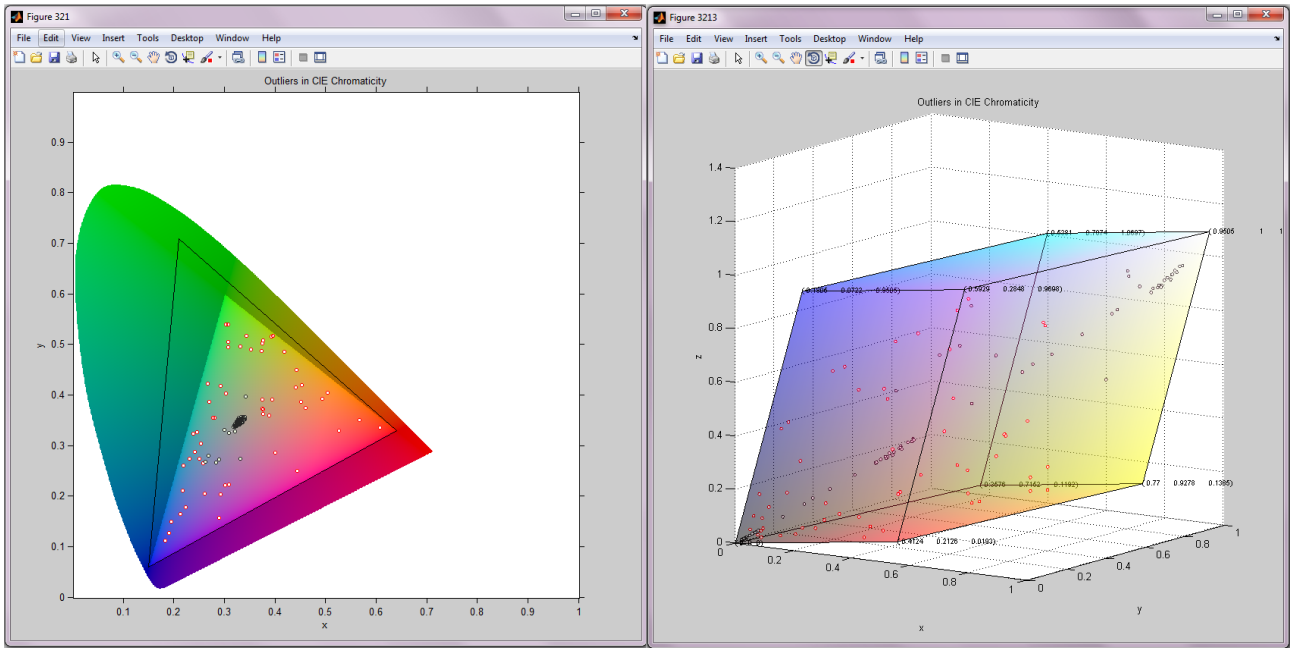
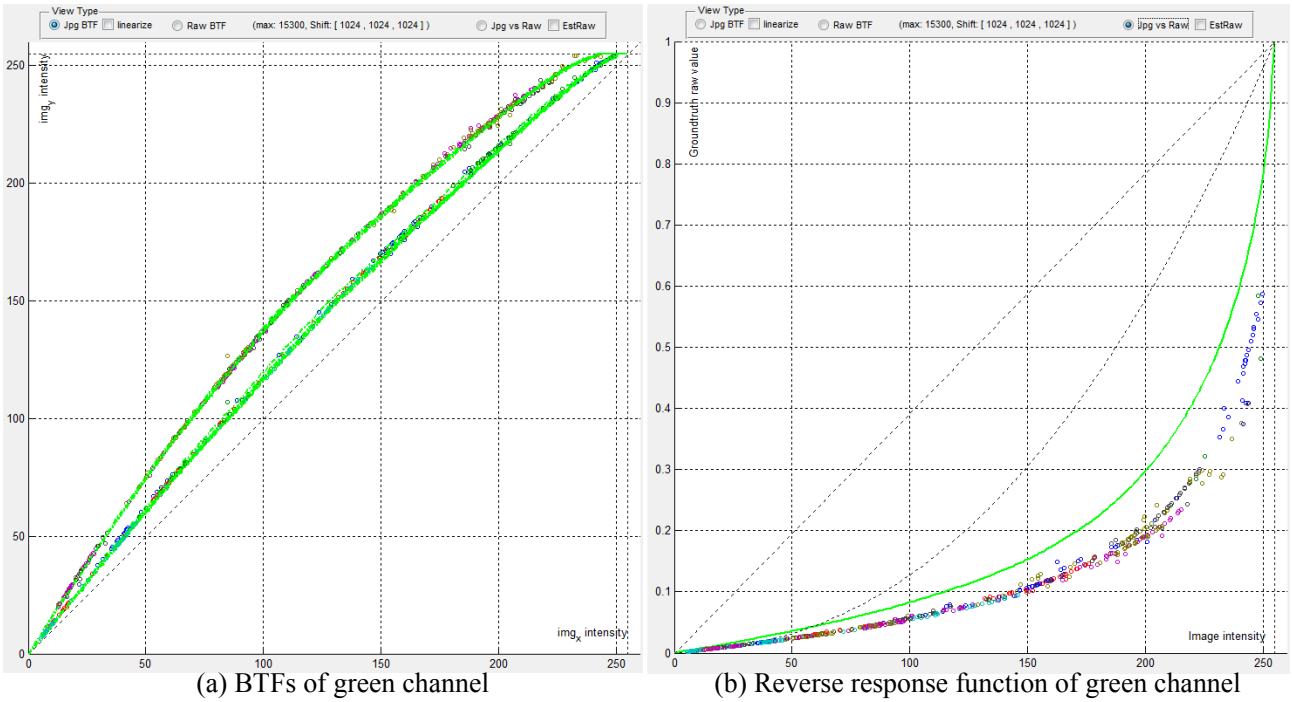


Figure A.6: Inliers and outliers shown in 2D CIE XYZ chromaticity diagram and 3D CIE XYZ color space. The red dots are outliers and black dots are inliers.

purpose. Fig. A.6 shows the windows showing both the inliers and outliers (sRGB color points) in 2D CIE XYZ chromaticity diagram and 3D CIE XYZ color space, both with sRGB color space specified inside. When the space transformation from RAW to sRGB is calibrated, the transformed RAW points could also be shown in those figures, which is informative and helpful during data analysis period.

Response function estimation

With the current model set to "Resp. only", by pressing the compute button, the coefficients of PCA model will be computed by optimization. Fig. A.7 (a) shows the BTFs in the sRGB BTF mode and Fig. A.7 (b) shows the reverse response function in the sRGB v.s. RAW mode, both take the green channel as an example. The linearization result only using response function could be plotted as shown in Fig. A.8.



(a) BTFs of green channel (b) Reverse response function of green channel

Figure A.7: Examples of BTFs and reverse response function.

A.3.2 White Balance and Space Transformation Estimation

When having the response functions calibrated, the white balance scales for each white balance setting and the space transformation from RAW space to linearized sRGB space are estimated from the data point correspondents: RAW color points and linearized sRGB color points. Some DSLR cameras record white balance scales in EXIF and those scales can be directly used. For others with unkown white balance scales, our algorithm computes the optimal solution simultaneously for the white balance scales and the space transformation matrix.

With the white balance and space transformation applied, the plotted points (transformed RAW value of one color channel v.s. the corresponding sRGB value of the same color channel) should fall on the response function of its corresponding color channel, if the gamut mapping function dose not affect the color points involved in this plotting. Fig. A.9 shows an example, where (a) plots only inliers and (b) plots both inliers and outliers.

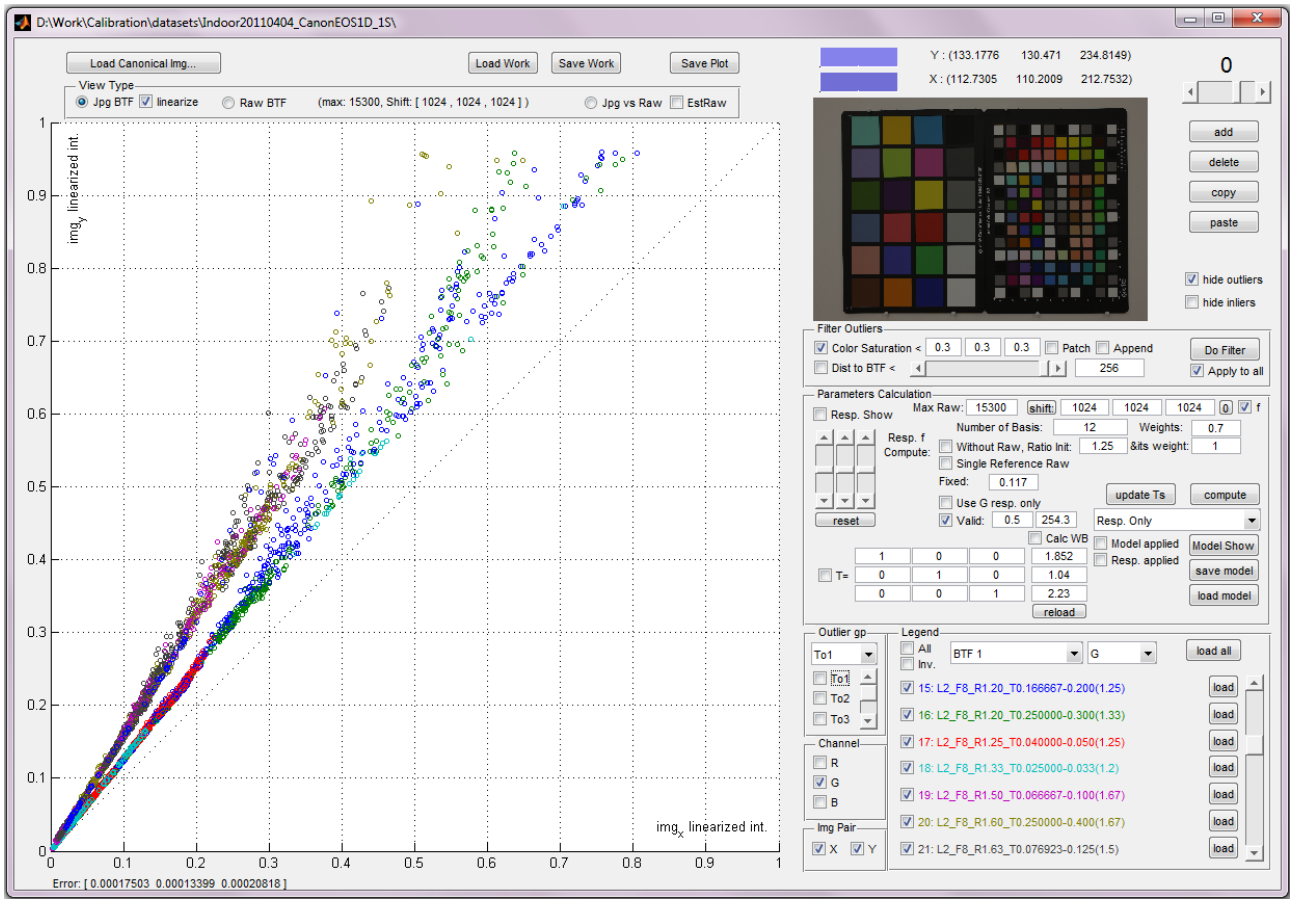
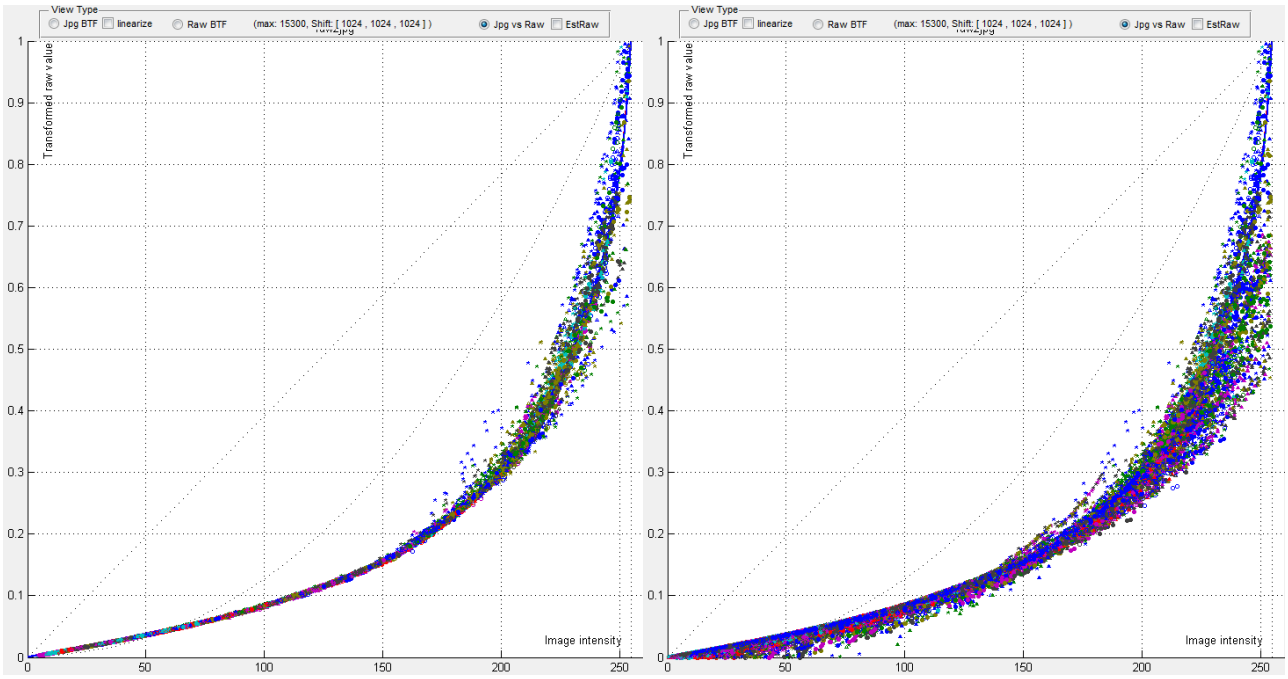
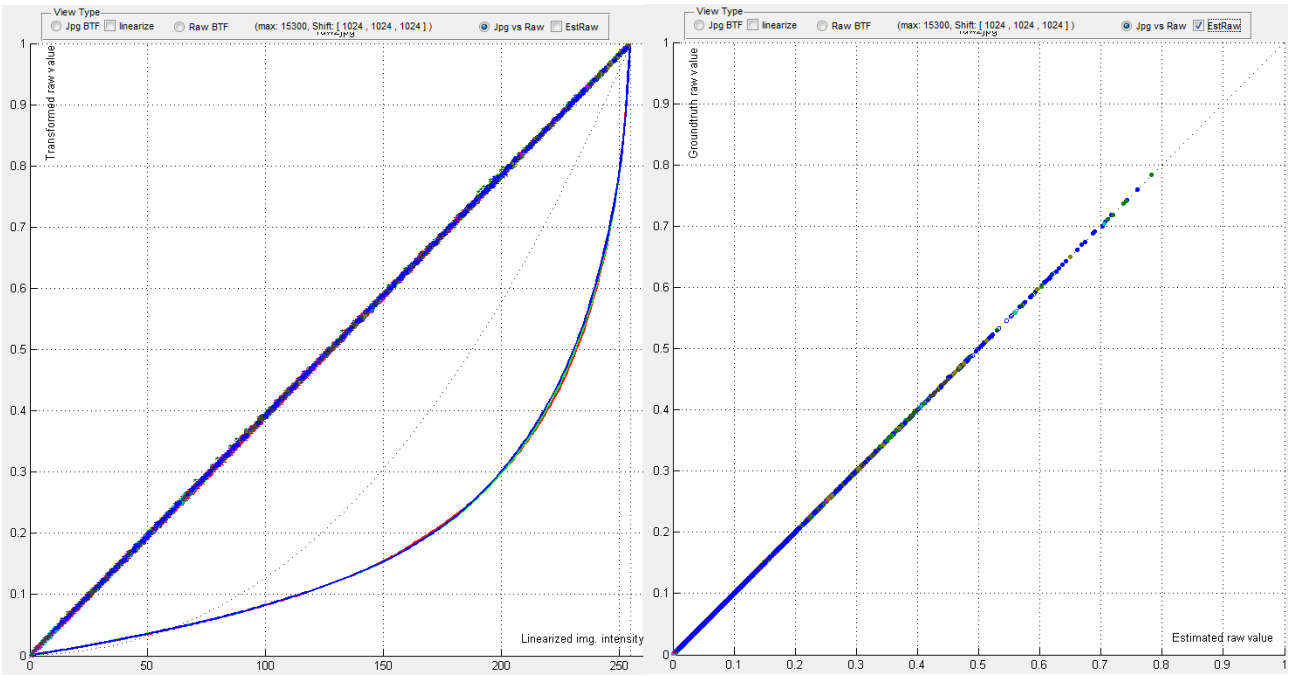


Figure A.8: Linearization result of green channel using response function only. Noted that those filtered outliers are also included in this plotting.



(a) Inliers only (b) Inliers and outliers

Figure A.9: Transformed RAW v.s. sRGB intensity.



(a) RAW to sRGB (b) sRGB to RAW

Figure A.10: Transformed RAW v.s. linearized sRGB calibrated using RBFs method.

A.3.3 Gamut Mapping Function Calibration

In this subsection, the gamut mapping function calibration based on two proposed methods is introduced. We follow the settings of the previous chapters, where RBFs based method only models the gamut mapping function, while the non-uniform lattice regression method models the whole pipeline except the white balance scales. Our software could be easily extended to deal with the modeling of the whole pipeline except the white balance scales based on RBFs method or model only the gamut mapping function based on lattice regression method.

RBFs based Method

As a global interpolation method, RBFs method works better in modeling a simpler underlying function. Here, only the gamut mapping function is modeled using RBFs. All loaded data points are used to calculate the RBFs parameters, i.e. no outliers any more in this step (the fully saturated color points which contain 255 in any one

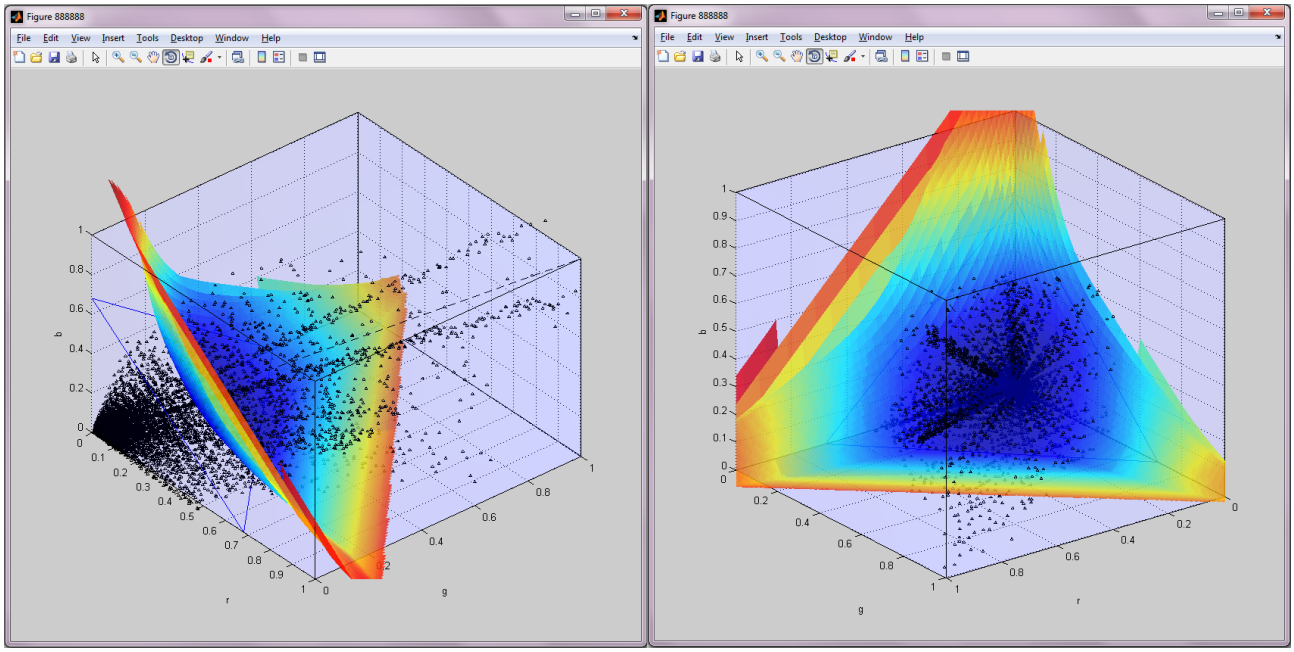


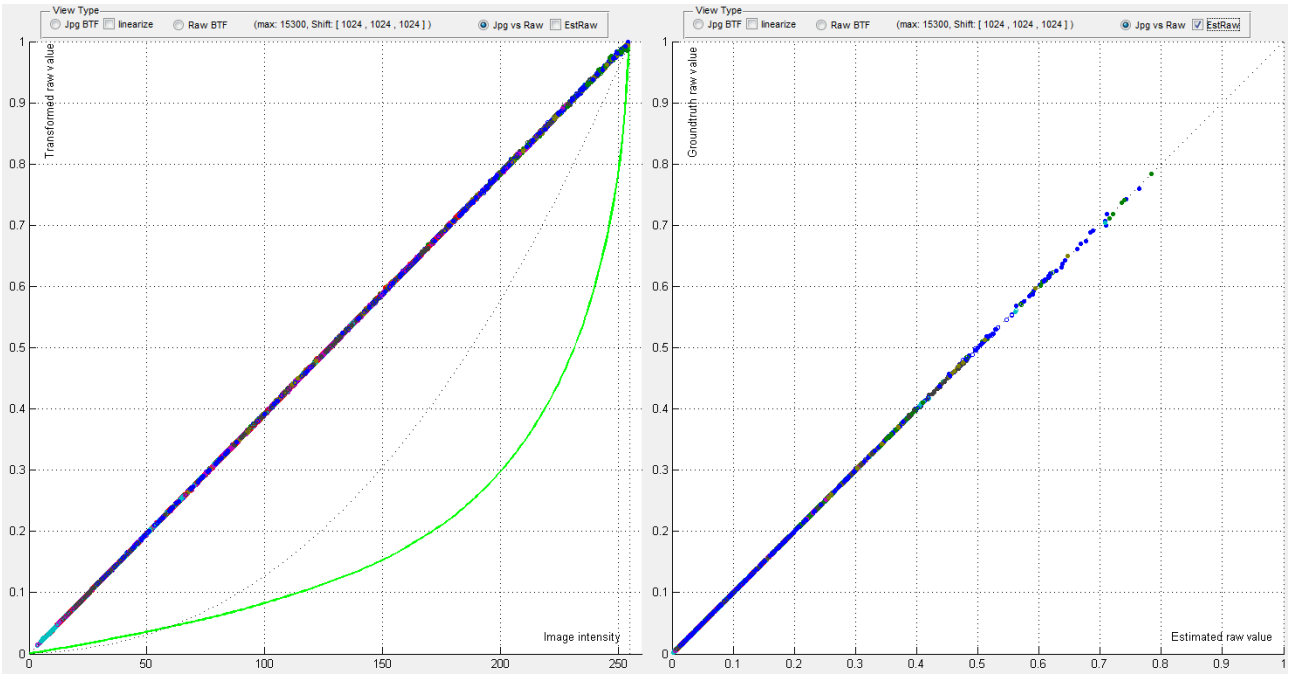
Figure A.11: Different views of the same gamut mapping function slice, from RAW to sRGB.

of RGB channels are still excluded in this step). Set the current model to "warp model" and press compute button, the algorithm automatically collects the set of control points and computes the model coefficients. Fig. A.10 shows the result of the calibration of the whole pipeline in both direction: from RAW to sRGB and from sRGB back to RAW, with gamut mapping function included. From this figure, we can see that the transformed RAW value almost equals to its corresponding linearized sRGB values in both cases.

The gamut mapping function is illustrated in Fig. A.11. The slice indicated by the blue triangle is mapped to the curved surface in the RAW to sRGB direction. The black dots are the control points of the RBFs.

Non-uniform Lattice Regression Method

Non-uniform lattice regression method treats the whole pipeline except the white balance scales as one function. In this case, the response function and the space transformation are not required to be computed in advance. Set the current model

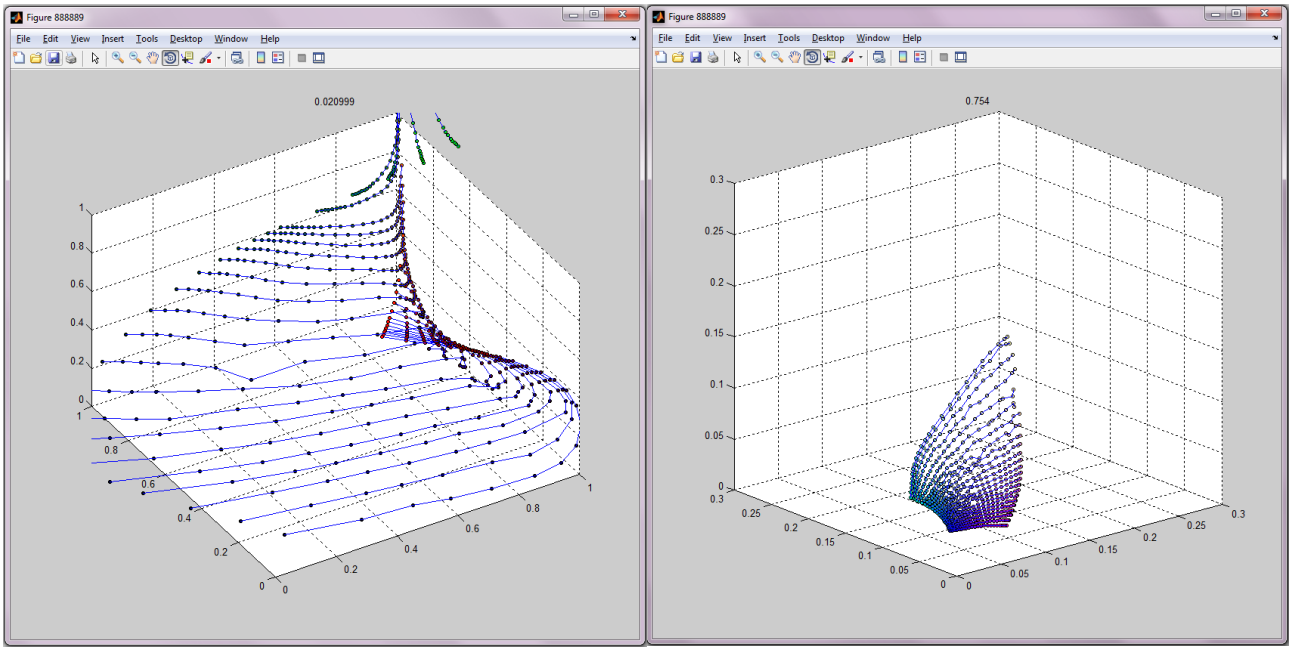


(a) RAW to sRGB (b) sRGB to RAW

Figure A.12: Transformed RAW v.s. linearized sRGB calibrated using non-uniform lattice regression method.

to "lattice" and press compute button, the algorithm automatically computes the model parameters for a lattice of a pre-specified size. Fig. A.12 shows the result of the calibration of the whole pipeline in both direction: from RAW to sRGB and from sRGB back to RAW, with gamut mapping function included.

The mapping function captured by the lattice model is illustrated in Fig. A.13. As an example, the mapping destinations (from RAW to sRGB color space) of the lattice vertices located on the plane whose blue channel equals to 0.021 are plotted in Fig. A.13 (a). The mapping destinations (from sRGB to RAW color space) of the lattice vertices located on the plane whose blue channel equals to 0.754 are plotted in Fig. A.13 (b).



(a) RAW to sRGB (b) sRGB to RAW

Figure A.13: The illustration of mapping destinations assigned to the lattice vertices on certain plane whose blue channel equals to the numbers indicated by the titles.

A.4 Summary

This appendix describes the main functions our interface provides. This calibration interface is a helpful tool for data analysis and an efficient instrument for calibrating the in-camera imaging pipeline. It is also a general framework extendable to incorporate more models for the calibration. We have also added several other models which are not described here, such as polynominal model and uniform lattice regression method. The source code can be downloaded from our project page. Interested users can develop their work further based on the current version.