2017

Data-Driven Radiometric Photo-Linearization

Han Li
College of William and Mary, zimushuihan@gmail.com

Follow this and additional works at: https://scholarworks.wm.edu/etd

Part of the Computer Sciences Commons

Recommended Citation
Li, Han, "Data-Driven Radiometric Photo-Linearization" (2017). Dissertations, Theses, and Masters Projects. Paper 1516639668.
http://dx.doi.org/doi:10.21220/S25Q2M

This Dissertation is brought to you for free and open access by the Theses, Dissertations, & Master Projects at W&M ScholarWorks. It has been accepted for inclusion in Dissertations, Theses, and Masters Projects by an authorized administrator of W&M ScholarWorks. For more information, please contact scholarworks@wm.edu.
Data-Driven Radiometric Photo-Linearization

Han Li

Xinyang, Henan, China

Bachelor of Engineering, Wuhan University, China, 2011

A Dissertation presented to the Graduate Faculty
of The College of William & Mary in Candidacy for the Degree of
Doctor of Philosophy

Department of Computer Science

College of William & Mary
July 2017
This Dissertation is submitted in partial fulfillment of the requirements for the degree of

Doctor of Philosophy

Han Li

Approved by the Committee, July 2017

Committee Chair
Associate Professor Pieter Peers, Computer Science
College of William & Mary

Weizhen Mao

Professor Weizhen Mao, Computer Science
College of William & Mary

Associate Professor Peter Kemper, Computer Science
College of William & Mary

Robert Michael Lewis

Associate Professor Robert Michael Lewis, Computer Science
College of William & Mary

Dr. Xing Tong,
Microsoft Research Asia
ABSTRACT

In computer vision and computer graphics, a photograph is often considered a photometric representation of a scene. However, for most camera models, the relation between recorded pixel value and the amount of light received on the sensor is not linear. This non-linear relationship is modeled by the camera response function which maps the scene radiance to the image brightness. This non-linear transformation is unknown, and it can only be recovered via a rigorous radiometric calibration process. Classic radiometric calibration methods typically estimate a camera response function from an exposure stack (i.e., an image sequence captured with different exposures from the same viewpoint and time). However, for photographs in large image collections for which we do not have control over the capture process, traditional radiometric calibration methods cannot be applied. This thesis details two novel data-driven radiometric photo-linearization methods suitable for photographs captured with unknown camera settings and under uncontrolled conditions.

First, a novel example-based radiometric linearization method is proposed, that takes as input a radiometrically linear photograph of a scene (i.e., exemplar), and a standard (radiometrically uncalibrated) image of the same scene potentially from a different viewpoint and/or under different lighting, and which produces a radiometrically linear version of the latter. Key to this method is the observation that for many patches, their change in appearance (from different viewpoints and lighting) forms a 1D linear subspace. This observation allows the problem to be reformulated in a form similar to classic radiometric calibration from an exposure stack. In addition, practical solutions are proposed to automatically select and align the best matching patches/correspondences between the two photographs, and to robustly reject outliers/unreliable matches.

Second, CRF-net (or Camera Response Function net), a robust single image radiometric calibration method based on convolutional neural networks (CNNs) is presented. The proposed network takes as input a single photograph, and outputs an estimate of the camera response function in the form of the 11 PCA coefficients for the EMoR camera response model. CRF-net is able to accurately recover the camera response function from a single photograph under a wide range of conditions.
# TABLE OF CONTENTS

Acknowledgments iii  
Dedication iv  
List of Tables v  
List of Figures vi  

1 Introduction  
1.1 Contributions ........................................... 3  
1.2 Dissertation Organization ................................... 4  

2 Background and Related Work  
2.1 Background ............................................. 5  
2.1.1 Terminology ............................................ 5  
2.1.2 Image Formation Process ................................. 5  
2.2 Related Work ............................................. 7  
2.2.1 Classic Methods of Radiometric Calibration .............. 8  
2.2.2 Radiometric Calibration in Large Photo Collections ...... 11  
2.2.3 Single Image Radiometric Calibration ..................... 11  
2.2.4 Empirical Model of Response (EMoR) .................... 12  

3 Example-based Radiometric Linearization of Photographs 14  
3.1 Introduction ............................................ 14  
3.2 Background – SIFT ........................................ 15  
3.2.1 SIFT ................................................... 15
ACKNOWLEDGMENTS

This dissertation is written with the support and help from many individuals. I would like to thank all of them.

First and foremost, I would like to express my deepest appreciation to my advisor, Dr. Pieter Peers. Without his guidance in my research, encouragement in my life, and confidence in my abilities, this dissertation would not have been possible. I would also like to thank my dissertation committee, Dr. Weizhen Mao, Dr. Peter Kemper, Dr. Robert Michael Lewis, and Dr. Xing Tong, for serving on my Ph.D committee as well as their insightful comments.

My sincere thanks also go to all members of the graphics group, Dr. Bo Dong, Dr. Kathleen Moore, Weiyi Zhang, Victoria L. Cooper, Zhaoliang Duan, and James C. Bieron for the stimulating discussions, constructive suggestions, and generous assistance.

Furthermore, I would like to thank the faculty and staff at the Computer Science Department of the College of William & Mary. Special thanks to Vanessa Godwin, Jacquelyn Johnson, and Dale Hayes for their considerate and effective assistance.

Last but not the least, I would like to thank my family, especially my parents, whose unwavering love and support has made me who I am today.

This dissertation was supported in part by the U.S. National Science Foundation under grants: IIS-1217765 and IIS-1350323, and a gift from Google.
This dissertation is dedicated to my beloved parents for their endless and selfless love and support.
LIST OF TABLES

<table>
<thead>
<tr>
<th>Table</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.1</td>
<td>Summary of the absolute and relative RMSE of the results in Figure 3.10</td>
<td>26</td>
</tr>
<tr>
<td>4.1</td>
<td>Summary of the DeepDetect’s ResNet-18 architecture used for CRF-net</td>
<td>51</td>
</tr>
<tr>
<td>4.2</td>
<td>Estimation and linearization errors over the validation dataset for different exposures scaled relatively with respect to the ’ideal’ auto-exposure</td>
<td>56</td>
</tr>
<tr>
<td>4.3</td>
<td>Estimation and linearization error over the validation dataset for different amounts of normal distributed camera noise</td>
<td>56</td>
</tr>
<tr>
<td>4.4</td>
<td>Estimation and linearization error over the validation dataset for different amounts of blur</td>
<td>59</td>
</tr>
</tbody>
</table>
LIST OF FIGURES

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.1</td>
<td>Flow diagram showing two mappings: the mapping from scene radiance to image irradiance is linear, and the mapping from image irradiance to image brightness is non-linear.</td>
<td>6</td>
</tr>
<tr>
<td>2.2</td>
<td>Example of different camera response curves from DoRF: DoRF contains 201 camera response curves for common brands of film, as well as video and digital cameras (Image from [2]).</td>
<td>7</td>
</tr>
<tr>
<td>2.3</td>
<td>Macbeth color chart with 24 color patches with known reflectances.</td>
<td>8</td>
</tr>
<tr>
<td>2.4</td>
<td>EMoR: (a) The mean of the camera responses in DoRF. (b) First four basis of the DoRF. (c) The cumulative energy occupied by the first 10 basis. The subspace corresponding to the three largest eigenvalues (an EMoR model with three parameters) captures more than 99.5 percent of the energy (Image from [2]).</td>
<td>13</td>
</tr>
<tr>
<td>2.5</td>
<td>EMoR in log space: (a) The mean of the camera responses in DoRF in log-space. (b) Four principal components in log-space. (c) A plot showing the percentages of the energies in log-space. This shows a three-dimensional subspace captures more than 99.6 percent of the energy in log-space (Image from [2]).</td>
<td>13</td>
</tr>
<tr>
<td>3.1</td>
<td>Example of feature points: First row: Two photographs capturing snow mountains from different viewpoints and under different illuminations. Second row: Feature points extracted from photographs. (Image from [3]).</td>
<td>16</td>
</tr>
</tbody>
</table>
3.2 The Process of Scale Space Extrema Detection: In each scale, the input image is repeatedly convolved with a Gaussian kernel to produce a set of Gaussian images shown on the left. Difference-of-Gaussian images are generated by subtracting adjacent Gaussian images, as shown on the right. Different scales are created by downsampling by a factor of 2. (Image from [4].)

3.3 Keypoint Selection: (a) Input image. (b) Initial feature keypoint candidates from scale space. (c) Keypoint candidates after removing points with low contrast. (d) Remaining candidates after removing points near edges. (Image from [4].)

3.4 Examples of SIFT Descriptor Generation: Left: 8×8 neighborhood. Right: Computed 2×2 descriptor. (Image from [4].)

3.5 Top Row: two radiometrically linear photographs captured under different lighting conditions (office lighting and LCD-panel illumination respectively). Bottom Left: ratio κ(x) of the input images. Bottom Right: False color plot of the error on a 1D linear subspace approximation (κ(x) ≈ κx) for a 33×33 window around each pixel location x.

3.6 Outlier Rejection. Recovered camera response function (red) with corresponding reference camera response function (black) from the Desk example shown in Figure 3.5 before and after outlier rejection. The false color cloud represents the recovered pixel radiance for each pixel in the patches (times the corresponding patch scale factor).

3.7 Stability Validation. False color difference image between the radiometric transfer results from two different source exemplar images with significantly different lighting and view conditions.
3.8 Transfer between Different Camera Models. Robustness validation of radiometric transfer between three different camera models: Nikon D700, Canon 600D, and Canon 60D.

3.9 Robustness validation of radiometric transfer under different lighting conditions. Four selected results from transfers between simulated captures from the WILD database [5] and a random camera response from the DoRF database [2]. The first three columns show successful transfer results under vastly different lighting conditions between the source and target image. The last column shows a failure case where insufficient SIFT features were found.

3.10 Radiometric Transfer Results. Left to right: radiometrically calibrated source image, radiance transfer result (pixels outside the recoverable range are marked in white – also highlighted in the inset), reference linearized target image obtained by applying the (inverse) ground truth camera, false color difference, and camera response curve (recovered in red, ground truth in black). In addition, the recovered pixel radiance (times scale factor) for each pixel value is included in the right plot. Ideally this should form a tight ‘cloud’ around the ground truth camera response curve.

4.1 Example of an artificial neuron: this artificial neuron takes two inputs \( x_1 \) and \( x_2 \), and produces one output.

4.2 Example of an artificial neural network: this neural network contains three layers – one input layer containing two input neurons, one hidden layer containing three hidden neurons, one output layer containing one output neurons.
4.3 Example of a receptive field: each $5 \times 5$ region of input neurons is connected to one hidden neuron. All $5 \times 5$ regions for all input neurons share the same weight and bias when connecting with the hidden neurons.

4.4 Example of stride 1: for each move, the step size is 1.

4.5 Example of stride 2: for each move, the step size is 2.

4.6 Example of zero-padding: the original size is $32 \times 32 \times 3$, and after padding 2 zeros on the boarder, the input size is $36 \times 36 \times 3$. Then, applying a convolutional layer of $5 \times 5 \times 3$ with stride 1, we can still get an output size of $32 \times 32$.

4.7 Example of a Max Pooling layer: a $2 \times 2$ Max Pooling Layer takes the maximum number over $2 \times 2$ region as output.

4.8 Example of a Average Pooling layer: a $2 \times 2$ Average Pooling Layer takes the average value of $2 \times 2$ region as output.

4.9 Example of a Fully-connected layer

4.10 LeNet: Architecture of LeNet-5, a Convolutional Neural Network for digits recognition. Each plane is a feature map. (Image from [5]).

4.11 AlexNet: Architecture of AlexNet, a Convolutional Neural Network containing 5 convolutional layers, max-pooling layers, dropout layers, and 3 fully-connected layers. Two streams show that the training process is splitted onto 2 GPUs. (Image from [7]).

4.12 VGGNet: Architecture of VGGNet, a deep Convolutional Neural Network made up with small filters. The depth of the configurations increases from the left (A) to the right (E), as more layers are added (the added layers are shown in bold). (Image from [8]).

4.14 Three examples of photographs and corresponding ground truth (purple) and estimated (green) camera response functions. The estimation errors are: 0.326, 0.267, 0.491, and the linearization errors are $(\times 10^{-5})$: 0.902, 0.423, 2.293.

4.15 Examples of suboptimal radiometric calibration. The left image exhibits many oversaturated pixels, whereas the right exhibits a very high contrast. In both cases, it is difficult to find good windows that sufficiently (and uniformly) cover the full pixel range. The respective estimation (and linearization $(\times 10^{-5})$) errors are: 2.367(5.701) and 3.927(10.12).

4.16 Estimated camera response curves from a single window: a randomly selected one (red) and one selected with the proposed selection criterion.

4.17 Radiometric calibration of colored versus grayscale images. Grayscale images exhibit less variation in intensity distributions, and are therefore less robust to calibrate. The top row shows a successful calibration for both color and grayscale; the bottom row shows an example where radiometric calibration on a colored image succeeds, but fails on the same grayscale image.

4.18 Comparison to between Lin et al.’s single image radiometric calibration method [10] and CRF-net on an image for which the former works well.
Results from CRF-net applied to captured (tone-mapped) photographs obtained with different camera models (Canon 600D, Nikon D700, Nikon D750), and a single well-exposed photograph from an exposure stack [11]. The respective estimation (and linearization ($\times 10^{-5}$)) errors are: 2.1758 (0.0332), 0.4221 (0.0239), 0.7517 (0.0606), and 1.7897 (0.3951).
Data-Driven Radiometric Photo-Linearization
Chapter 1

Introduction

In computer vision and computer graphics, a photograph is often considered a photometric representation of a scene. However, directly using such a recorded photograph in computations that rely on accurate radiometric estimates is incorrect due to the inherent non-linear transformation of physical radiance into perceptual brightness values preferred by human viewers. This non-linear transformation is modeled by a camera response function that can differ between camera models, and which can only be recovered via a rigorous radiometric calibration process.

Classic radiometric calibration methods typically estimate a camera response function from an exposure stack (i.e., an image sequence captured with different exposures from the same viewpoint and time). Image irradiance in an exposure stack is constant for each pixel, thus the ratio of captured sensor exposures for the same locations in an image pair is the same as the exposure ratio. With known exposure ratios and corresponding pixel values in an exposure stack, it is possible to estimate the camera response function. Given the camera response function, we can then undo the non-linear transformation (i.e., radiometric linearization). However, for photographs in large image collections for which we do not have control over the capture process, traditional radiometric calibration methods cannot be applied since we have no information on the exposures. Given a large community image collection, we want to leverage the information embedded in this large image collection, to radiometrically calibrate and linearize all the photographs in the
image collection which are possibly captured by different camera models.

1.1 Contributions

This dissertation proposes two solutions towards radiometric calibration and linearization of photographs acquired under uncontrolled conditions:

1. Example-based Radiometric Linearization of Photographs

We propose a novel example-based radiometric linearization method that takes as input a radiometrically linear photograph of a scene (i.e., exemplar), and a standard (radiometrically uncalibrated) image of the same scene potentially from a different viewpoint and/or under different lighting, and which produces a radiometrically linear version of the latter. In detail,

- We present a novel method to estimate the camera response function for an input photograph based on correspondences between this input photograph and another radiometrically linearized photograph captured from totally different viewpoints and under different lightings for the same scene.
- We propose a method to select best matching patches/correspondences between two photographs and align those patches to get accurate results.
- We introduce a robust outlier rejection method to remove matched patches which cannot be well represented by our model.
- We prove that under modest assumptions, the change in appearance of a small local pixel neighborhood in a photograph resides in a 1D linear subspace.

2. CRF-net: Single Image Radiometric Calibration using CNNs

We propose a robust single image radiometric calibration method based on convolutional neural networks (CNN), named CRF-net (or Camera Response Function net). The proposed network takes as input a single photograph, and outputs an
estimate of the camera response function in the form of the 11 PCA coefficients for the EMoR camera response model [2].

- We train a CRF-net, which can estimate the camera response function of any single input photograph.
- We build a sufficiently large training database of well-exposed radiometrically linear “RAW” photographs, captured with different camera models from a variety of scenes, and captured under a variety of conditions.
- We propose a method to select 10 windows that best cover the intensity range from the input photograph and aggregate the estimated camera response functions from the 10 well-chosen windows, by removing the outliers and averaging the PCA coefficients of the remaining estimated camera response functions.

1.2 Dissertation Organization

The rest of this dissertation is structured as follows. In Chapter 2, we discuss background and related work of radiometric calibration. In Chapter 3, we describe our novel example-based radiometric linearization method. In Chapter 4, we propose a single image radiometric calibration method using convolutional neural networks. Finally, we conclude the dissertation in Chapter 5.
Chapter 2

Background and Related Work

This chapter reviews background and related work on radiometric calibration.

2.1 Background

2.1.1 Terminology

In radiometry, radiance is defined as the power passing through or emitted from a surface per differential area per differential solid angle. In geometry, a solid angle is defined as the surface area of a unit sphere covered by the surface’s projection onto the sphere. It is expressed as a dimensionless unit called steradian (sr); it is the 3D analogue of radians on circle. The unit for radiance is watts per square meter per steradian ($W \times m^{-2} \times sr^{-1}$).

Scene radiance ($L$) is the amount of light reflected or emitted from a visible object and going into the direction of the camera, and image irradiance ($E$) is the amount of light that falls onto a specified area of camera sensor. The unit for irradiance is watts per square meter ($W \times m^{-2}$).

2.1.2 Image Formation Process

As shown in Figure 2.1 when light passes through a camera lens, scene radiance $L$ is attenuated by the lens aperture (a hole through which light travels), resulting in an irradiance on the sensor linearly proportional to radiance:
Figure 2.1: Flow diagram showing two mappings: the mapping from scene radiance to image irradiance is linear, and the mapping from image irradiance to image brightness is non-linear.

\[ E = L \times n, \]  
(2.1)

where, \( n \) is a linear factor that depends on the camera parameters and lighting directions.

Sensor irradiance is integrated at the sensor for a user-set exposure time \( k \), resulting in an exposure:

\[ X = k \times E, \]  
(2.2)

which is then non-linearly converted to pixel brightness via the camera response function \( f \).

The relationship between image irradiance \( E \) and image brightness \( I \) can be formulated as follows:

\[ I = f(k \times E). \]  
(2.3)

According to the described image formation process, when a photograph is captured using a digital camera, the final digitized pixel values are not exact measurements of scene radiances, but rather of non-linear brightness values. For example, if the scene radiance at point A is twice that of the scene radiance at point B, then the pixel value of captured point A in the photograph will not be twice the pixel value of the captured point B. Instead, the pixel value and the actual scene radiance share an unknown nonlinear relationship between each other. The effects of the different nonlinear mappings are aggregated into a single function called the camera response function or camera response curve. The process of undoing this nonlinear mapping to recover image irradiance is called radiometric linearization.
Different camera models can have vastly different camera response functions. The exact form of the camera response function plays a major role in the quality and “feel” of the image quality of a camera, and therefore camera response functions are considered a trade secret and not publicly shared by camera manufacturers. Figure 2.2 shows a selection of camera response curves from the DoRF database. DoRF (Database of Response Functions) is a database collected by Grossberg and Nayar in 2004 which contains 201 camera response curves for common brands of film, as well as video and digital cameras.

2.2 Related Work

Radiometric calibration is an essential step for any computer vision or computer graphics method that relies on extracting radiometric cues from photographs. Because the exact form of the camera response functions are not published by camera manufacturers, researchers are forced to reverse engineer these camera response functions via an inverse process. There exists a large body of prior work on various strategies to recover camera response functions.

2.2.1 Classic Methods of Radiometric Calibration

One way of radiometric calibration is to capture a photograph with a chart in the scene with known reflectances, such as the Macbeth color checker chart (shown in Figure 2.3). The Macbeth chart includes 24 patches with known reflectances of different colors, which provides a mapping from known reflectances to image intensities. Typically, one can apply standard interpolation based on this mapping to obtain a continuous camera response function. The reason for doing interpolation is that the Macbeth chart only contains a limited number of patches which do not cover all possible pixel intensities. However, it is not convenient to do radiometric calibration using a color chart, since this method requires a color chart to be presented in a photograph.

The most common approach to radiometric calibration estimates the camera response
function from an exposure stack (i.e., an image sequence captured under varying exposure from the same viewpoint at the same period of time) of a static scene (captured with a static camera).

Assuming that the scene is static and the capturing process is completed in a short time so that the illumination is constant, the irradiance values $E_x$ for each pixel $x$ are constant. If we denote pixel values by $P_x$ for each pixel $x$, then the camera response function is determined by:

$$P_x = f(kE_x),$$  \hspace{1cm} (2.4)

where $k$ is the exposure value. Assuming $f$ is monotonic, we can take the inverse of both sides:

$$f^{-1}(P_x) = kE_x.$$  \hspace{1cm} (2.5)

By taking the natural logarithm of both sides, we get:

$$\ln f^{-1}(P_x) = \ln k + \ln E_x.$$  \hspace{1cm} (2.6)

Denote an exposure stack as a set of photographs $\{P_{x,i}\}$ with corresponding exposures $\{k_i\}$, then for two images taken under exposure values $k_i$ and $k_j$, we obtain the following two equations:

$$\ln f^{-1}(P_{x,i}) = \ln k_i + \ln E_{x,i},$$  \hspace{1cm} (2.7)

$$\ln f^{-1}(P_{x,j}) = \ln k_j + \ln E_{x,j}.$$  \hspace{1cm} (2.8)

Note that $E_x$ is constant for each equation, thus $E_{x,i} = E_{x,j}$. By subtracting Equation 2.7 from Equation 2.8 and canceling out $E_x$, we can get the following equation:

$$\ln f^{-1}(P_{x,i}) - \ln f^{-1}(P_{x,j}) = \ln k_i - \ln k_j.$$  \hspace{1cm} (2.9)

To simplify notations, we define function $g = \ln f^{-1}$, and assume $k_{ij} = k_i/k_j$, and rewrite
Equation 2.9 as:
\[ g(P_{xi}) - g(P_{xj}) = \ln k_{ij}. \] (2.10)

In Equation 2.10, given known pixel values \( P_{xi} \) and \( P_{xj} \), and known exposure ratio \( k_{ij} \), we can recover the function \( g \) subject to a priori form of the camera response function. Several models for camera response functions have been proposed. Debevec and Malik [11] estimate a non-parametric camera response model regularized by a smoothness constraint. Mitsunaga and Nayar [12] use a flexible polynomial model and only require approximate estimates of the exposures ratios. Grossberg and Nayar [2] propose a data-driven model based on a large database of measured camera response functions, using principal component analysis (PCA), called the empirical model of response functions (EMoR). However, methods that rely on an exposure stack are limited to static cameras and scenes. If images are captured by non-static cameras or scenes, such as videos or scenes captured under different lighting conditions, the relationship between ratios of sensor exposures and exposure time does not hold anymore, since the assumption that irradiance values \( E_x \) for each pixel \( x \) are constant is violated.

Several strategies have been proposed to address this non-static camera problem. Grossberg and Nayar [13] use intensity histograms and derive a brightness transfer function between two photographs captured with different exposures. Kim and Pollefeys [14] exploit epipolar geometry and stereo matching to find corresponding points. In follow up work, Kim et al. [15] integrate tracking and camera response function recovery to further improve the quality of both.

In concurrent work to our second contribution, Kalantari and Ramamoorthi [16] use a convolutional neural network (which will be explained in section 4.2) as a learning model for producing a high dynamic range image from a set of images with different exposures. Kalantari and Ramamoorthi focus on composing an HDR image, requiring multiple input images captured by one camera for one scene.

However, these methods are not suited for large photo collections, because photos from online collections are usually captured by different camera models and under vastly
2.2.2 Radiometric Calibration in Large Photo Collections

The recent interest in mining visual information from online photo collections necessitates radiometric calibration methods that can work not only for non-static cameras, but also for non-static scenes and in particular for uncontrolled and unknown lighting. Shaque and Shah [17] recover the camera response function, modeled by a gamma curve, for fixed-viewpoint photographs under different uncontrolled lighting. Kim and Pollefeys [18] perform radiometric calibration for static viewpoint images of outdoor scenes under changing illumination by grouping pixels with similar behavior with respect to changes in illumination. Diaz and Strum [19, 20] recover the camera response functions for images in large photo collections assuming Lambertian surface reflectance and directional or low frequency lighting respectively. They use an inverse rendering approach and leverage the geometry obtained from multiview stereo [21]. Kuthirummal et al. [22] establish prior statistics (i.e., joint histograms of irradiances at neighboring pixels are very similar for different camera models) from large photo collections to recover radiometric camera properties by minimizing the difference between joint histograms of irradiances.

However, these methods are mainly designed to recover only a single camera response function for one camera model, not to radiometrically linearize each photograph in the photo collection, each captured with a potentially different camera, and thus a different camera response function.

2.2.3 Single Image Radiometric Calibration

A different strategy for radiometric calibration is to rely on single image methods which exploit statistical properties of photographs. For a local edge region in a photograph, measured color values across edges should follow linear distributions because of linear color blending at edge pixels. But after applying a non-linear camera response function, the edge colors form a non-linear distribution. Based on the mapping from the non-linear
color distribution at the edge regions into a linear distribution, Lin et al. [23] estimate a camera response function from only a single input image. Using a similar idea, Lin and Zhang [24] calculate the camera response function of a single grayscale image by using the histograms of edge regions. Matsushite and Lin [25] exploit the expected symmetry of camera noise distributions. While, the noise distribution should be symmetric for image irradiances, the observed pixel noise distributions are skewed because of the non-linear camera response functions). Lin and Zhang use the asymmetric profiles of measured noise distributions to do radiometric calibration. Takamatsu et al. [26] infer camera response functions by looking at the non-affinity relationship between image intensities and noise variances. These methods all rely on specific cues which are not always sufficiently present in random photographs, such as high levels of image noise or the assumption on the symmetric noise distribution. Hence, these methods are not robust enough to process large photo collections.

2.2.4 Empirical Model of Response (EMoR)

In this section, we will further discuss the Empirical Model of Response functions (EMoR) in detail, as we will use this model for modeling the camera response functions in this dissertation. Grossberg and Nayar [2] collect a Database of Response Functions (DoRF) of a variety of imaging systems, which include a total of 201 real-world response curves, and then formulate a new model for camera response functions called the Empirical Model of Response functions (EMoR), which is a low-parameter approximation model. The EMoR is obtained by applying PCA to the training curves from DoRF:

$$f(E) = f_0(E) + \sum_{n=1}^{M} c_n h_n(E),$$  \hspace{1cm} (2.11)

where $M$ is the total number of basis elements, $h_1, h_2, ..., h_M$ are the basis functions, $f_0$ is the mean function, and $f(E)$ is the camera response function of irradiance $E$.

Figure 2.4 shows that EMoR can represent the space of camera response functions accurately. Retaining only the 3 bases with the largest eigenvalues can already cover
Figure 2.4: EMoR: (a) The mean of the camera responses in DoRF. (b) First four basis of the DoRF. (c) The cumulative energy occupied by the first 10 basis. The subspace corresponding to the three largest eigenvalues (an EMoR model with three parameters) captures more than 99.5 percent of the energy (Image from [2]).

Figure 2.5: EMoR in log space: (a) The mean of the camera responses in DoRF in log-space. (b) Four principal components in log-space. (c) A plot showing the percentages of the energies in log-space. This shows a three-dimensional subspace captures more than 99.6 percent of the energy in log-space (Image from [2]).

99.5% of the energy. 11 basises can recover the original curves at almost 100% accuracy.

When operating with exposure ratios, it is more convenient to work with the log variant of the EMoR model. In log space,

\[ g(E) = g_0(E) + \sum_{n=1}^{M} c_n h_{\text{log},n}(E), \]  

(2.12)

where \( g(E) = \ln f^{-1}(E) \) is the camera response function in log space, \( g_0 \) is the mean function, \( h_{\text{log},n} \) are the basis functions, both found by applying PCA to the log space of training curves in DoRF. Figure 2.5 shows that EMoR model in log space can also accurately characterize most camera response functions.
Chapter 3

Example-based Radiometric Linearization of Photographs

3.1 Introduction

Most classic radiometric calibration methods focus on recovering the camera response function from an exposure stack of a static scene recorded from a fixed viewpoint and under static lighting conditions. However, such calibration methods are not suited for consumer cameras that offer limited control on exposure or that employ an image-content dependent camera response function. Similarly, such methods are also not suited when direct access to the camera is not possible (e.g., internet photographs). However, one usually has access to (or the opportunity to record) a radiometrically linearized image of the same scene, albeit recorded with a different camera and/or under different lighting or viewing conditions. This potentially rich source of information has not been considered for radiometric calibration, and it raises the question of whether it is possible to transfer radiometric information from a radiometric linear image to an uncalibrated photograph of the same scene.

In this chapter, we propose a novel example-based radiometric linearization method that takes as input a radiometrically linear photograph of a scene (i.e., exemplar), and a standard (radiometrically uncalibrated) image of the same scene potentially from a dif-
ferent viewpoint and/or under different lighting, and which produces a radiometrically linear version of the latter. Key to our method is the observation that under modest assumptions, the change in appearance of a small local pixel neighborhood in a photograph resides in a 1D linear subspace. This allows us to formulate a fast and lightweight solution that resembles the classic solution for radiometric calibration from exposure stacks.

We demonstrate the qualitative accuracy of our method on a variety of different scenes, and quantitatively validate the robustness and accuracy of various components of our system.

3.2 Background – SIFT

In order to relate the different observations of surface points between photographs, we need to establish exact corresponding pixel pairs of the same surface point in two images captured under different viewpoints. For this we will rely on SIFT features \[4\]. SIFT or Scale-Invariant Feature Transform is the most popular technique for feature extraction and matching in computer vision. In this section, we review the key idea of SIFT.

3.2.1 SIFT

Given an image pair that contains the same object but captured from different viewpoints and under different illumination conditions (as is commonly the case in large photo collections), it is easy for a human observer to identify corresponding points in both images. However, it is difficult for computers to identify the correspondences. Algorithms in computer vision rely on image features to compare two images. Image features are interesting points that identify an object, such as uniquely shaped patches of snow or peaks of a mountain as shown in Figure 3.1. To get good matching results, reliable features are required that are stable under changes in image rotation, scale, and illumination.

SIFT \[4\] is a widely used method to detect features in images, that are invariant to
scale, rotation and illumination changes. SIFT features can be extracted in 4 steps: (1) Scale-Space Extrema Detection, (2) Keypoint Localization, (3) Orientation Assignment and (4) Descriptor Construction

1. Scale-Space Extrema Detection

To detect features from image pairs under different scales, we cannot only consider pixels in fixed-size pixel neighborhoods, due to changes in feature size. Instead, we need to search all possible scales for all locations in images to find reliable features. The exploration of the different scales can be sped up by working in “scale space”. The scale space of an image is computed by (1) first repeatedly convolving an input image with a Gaussian kernel, and then subtracting the adjacent Gaussian images to produce “Difference of Gaussian” images followed by (2) a downsampling by a factor of 2 and repeating step (1) (Figure 3.2). Local minima and maxima are then searched in the scale space by comparing with every neighbor at the current and adjacent scales.

2. Keypoint Selection

Since the result from a Difference-of-Gaussian function is very sensitive to noise and local edges, feature candidates selected from the first stage (Scale-Space Extrema Detection) have to be refined to remove unreliable feature keypoints that are near
local edges or have low contrast (and thus are sensitive to noise) to get more reliable feature points. To remove keypoints with low contrast, a Taylor expansion of each keypoint in the Difference-of-Gaussian scale space is evaluated to find an extremum. If the Taylor function evaluation at the extremum is less than a threshold, then this keypoint is rejected. To eliminate strong edge responses, a $2 \times 2$ Hessian matrix (a square matrix of second order partial derivatives of a function) at each keypoint is computed to detect local edges. The ratio ($r$) of the largest two eigenvalues of the Hessian matrix for a candidate keypoint is used to calculate a ratio $R$:

$$R = \frac{(r + 1)^2}{r}.$$  \hspace{1cm} (3.1)

If $R$ is larger than a threshold, this keypoint is detected as local edges and hence rejected. All keypoints left are considered as reliable feature keypoints. Figure 3.3 shows an example of keypoint selection.

3. Orientation Assignment
Figure 3.3: Keypoint Selection: (a) Input image. (b) Initial feature keypoint candidates from scale space. (c) Keypoint candidates after removing points with low contrast. (d) Remaining candidates after removing points near edges. (Image from [4].)

Feature candidates from the second stage (Keypoint Selection) are only invariant to image scale. To achieve invariance to image rotation, an orientation parameter is assigned to each feature keypoint. This orientation parameter is calculated from histogram information created by the gradient magnitude and direction in a neighborhood around each feature keypoint.

4. Descriptor Construction

A keypoint descriptor is computed to uniquely identify each keypoint and improve invariance to illumination changes. First, a $16 \times 16$ neighborhood centered around a keypoint is divided into $4 \times 4$ subregions. Orientation histograms are then computed from the gradient magnitudes and orientations at each image point on a $4 \times 4$ neighborhood, weighted by a Gaussian window. A keypoint descriptor is a vector of all values from orientation histograms in a $16 \times 16$ neighborhood for each keypoint, normalized to unit length to reduce the effects of illumination changes. [Figure 3.4]
Figure 3.4: Examples of SIFT Descriptor Generation: Left: 8×8 neighborhood. Right: Computed 2×2 descriptor. (Image from [3].)

shows an example of a 2×2 descriptor.

To find matching points from image pairs A and B, the Euclidean distance between every descriptor from each keypoint needs to be computed. For a keypoint $p_1$ in A, two keypoints $p_2$ and $p_3$ that have the first and second closest Euclidean distances with $p_1$ are used to calculate the ratio between the Euclidean distance of $p_1$ with $p_2$ and $p_1$ with $p_3$. If the calculated ratio is smaller than a certain threshold, then the keypoint pair of $p_1$ and $p_2$ is accepted as a matching pair. Since SIFT descriptors are invariant to scale, rotation and illumination changes (to some degree), matching results from SIFT are also robust to these changes. The robustness and computational efficiency of SIFT feature matching has made it one of the cornerstones of many large photo-collection algorithms.

3.2.2 SIFT Flow

SIFT only provides matching points between image pairs. To obtain dense pixel-to-pixel correspondences between two images or patches, we use SIFT flow [27], proposed by Ce Liu et al. in 2008. First, a SIFT descriptor is extracted at each pixel to characterize local image structures. Then, SIFT flow is formulated similarly as optical flow with the exception of matching SIFT descriptors instead of pixel values. Optical flow is the pattern of apparent motion of image objects caused by the relative motion between objects and a camera. Optical flow works on the assumption that the pixel intensities of an image object do not change between two frames, so optical flow cannot provide
meaningful correspondences if two images are from different scene categories. The SIFT flow algorithm allows densely robust matching SIFT features between two images, and preserves spatial discontinuities at the same time.

3.3 Radiometric Transfer

Problem Statement The image formation process in a digital camera can be abstracted as: \( M = f(E) \), where \( M \) is the resulting image and \( E \) is an image that is linearly proportional to the time-average of the incident radiance on the camera’s sensor. The exact scale between incident radiance and \( E \) depends on various camera characteristics such as exposure time, aperture, light efficiency of the sensor and lenses, etc. The camera response function \( f \), a non-linear mapping between \( E \) and the image \( M \), is designed to remap and compress the dynamic range in order to produce visually pleasing images. As noted before, the goal of radiometric calibration is to undo the effects of \( f \) and to recover a radiometrically linear image that is proportional (up to an unknown scale factor) to the time-average image irradiance \( E \).

Radiometric transfer takes as input two images; a radiometrically linear source image \( E_s \), and a regular target image \( M_t = f(E_t) \). Both images depict the same subject, but viewed from different viewpoints and under different lighting conditions. The goal of radiometric transfer is to recover \( E_t = f^{-1}(M_t) \) by exploiting the knowledge that \( E_s \) depicts the same subject.

Fixed Viewpoint We will first consider the case where the viewpoint is the same for both images, but the lighting conditions can differ. In this case, the relation between the known source irradiance \( E_s \) and the unknown target irradiance \( E_t \) can be expressed by their ratio: \( E_t(x) = \kappa(x)E_s(x) \). The ratio \( \kappa \) can potentially vary with pixel position \( x \) due to changes in the underlying surface normal, material properties, and/or angular variation in lighting (Figure 3.5, bottom-left). This makes it difficult to directly estimate \( \kappa \) for every pixel from \( E_s \) and \( M_t \) only. However, we observe that \( \kappa \) is locally slowly varying: \( E_t(x') \approx \kappa_x E_s(x') \) for \( x' \in N(x) \), a small neighborhood around \( x \). Hence, we
Figure 3.5: Top Row: two radiometrically linear photographs captured under different lighting conditions (office lighting and LCD-panel illumination respectively). Bottom Left: ratio $\kappa(x)$ of the input images. Bottom Right: False color plot of the error on a 1D linear subspace approximation ($\kappa(x) \approx \kappa_x$) for a $33 \times 33$ window around each pixel location $x$.

can approximate the ratio $\kappa(x)$ by a constant ratio $\kappa_x$ for a small neighborhood $N(x)$ around $x$:

$$f^{-1}(M_t(x')) \approx E_s(x')\kappa_x, \quad x' \in N(x).$$  \hspace{1cm} (3.2)

In other words, the appearance space of a small neighborhood of pixels, can be well approximated by a 1D linear subspace, as illustrated in Figure 3.5, bottom-right. Equation 3.2 is similar in form to that of classic radiometric calibration from multiple exposures \cite{11, 28}, except that in our case the “exposures” ($\kappa_x$) are unknown instead of the irradiance image ($E_s$). Similar as in prior work, we reformulate this expression in the log domain:

$$g(M_t(x')) \approx \log E_s(x') + \log \kappa_x,$$  \hspace{1cm} (3.3)

where $g = \log(f^{-1})$. We characterize the log-inverse of the camera response function
using the log-PCA model of Kim and Pollefeys \cite{28}:

$$h'_0(M_t(x')) + \sum_{i=1}^{n} c_i h'_i(I_t(x')) = \log E_s(x') + \log \kappa_x,$$

(3.4)

where \( n = 25 \) is the number of log-PCA terms, \( h'_i \) represents the \( i \)-th PCA term of the space spanned by the log-inverse camera response functions contained in the DoRF database \cite{2}, and \( h'_0 \) is the mean log-inverse camera response function. While \( n = 3 \) log-PCA terms already explain 99.6\% of the energy \cite{28}, we use 25 terms to better model irregular and uncommon camera response functions. Each \( x' \in \mathcal{N}(x) \) in Equation 3.4 provides a linear equation in terms of the unknowns \( \log \kappa_x \) and the \( n = 25 \) log-PCA coefficients \( c_i \). Combining the linear equations in a single system, and assuming sufficient variety in pixel values in each patch, allows us to solve for the \( n + 1 \) unknowns using a linear least squares solver. To improve the stability and to ensure sufficient coverage of the available pixel range, we consider the neighborhoods around \( m \) different pixel location \( x_k, k \in \{0..m-1\} \), and solve for the \( n + m \) unknowns: the \( n = 25 \) coefficients and the \( m \) different \( \log \kappa_{x_k} \) scale factors (one for each patch).

The final camera response function can then be computed by exponentiating and inverting the obtained function: \( f = (\exp g)^{-1} \). However, this is only a partial camera response function since the full range of pixel values might not be covered in the the \( m \) patches (or even in the target image \( M \)). Furthermore, there exists an ambiguity between the partial camera response function and the scale factors \( \kappa \): \((g + \gamma) - (\log \kappa + \gamma) = \log E_s\) for any \( \gamma \). Hence, the partial camera response function is only determined up to an unknown scale factor. To expand the range of the recovered partial camera response function, we linearly extrapolate the camera response function below the recovered lower limit of the range to the origin. However, due to the unknown scale factor, we cannot extrapolate beyond the upper limit, and simply cut off the response function at the upper limit of the range.

In some sense, our solution can be seen as inferring the camera response function and "exposures" (i.e., scale factors) from a set of \( m \) tiny "image-pairs" (i.e., patches).
**Different Viewpoint**  The above algorithm easily extends to the case where the viewpoint between the source $E_s$ and target image $M_t$ (and thus $E_t$) differ by introducing an additional function $\phi_{t \to s}$ that maps pixels in a target patch to the corresponding pixels in the source patch. We can then reformulate the local approximation: $E_t(x') \approx \kappa_x E_s(\phi_{t \to s}(x'))$ for $x' \in N_t(x)$, a small neighborhood around $x$ in the target image, which can be solved using a similar strategy as for the fixed viewpoint.

The direction of the mapping (from target coordinates to source coordinates) is critical, since such a mapping will remap integer pixel coordinates to fractional coordinates, requiring an interpolation to obtain the corresponding pixel intensity. The target image pixels are radiometrically non-linear, so the interpolation performed on target image pixels does not make sense. Consequently, such a warping operation will only be correct when executed on radiometrically linear pixel intensities (i.e., the source image).

**Patch Selection and Mapping**  A critical component in the above algorithm is the selection of the patches and the corresponding mapping functions. We desire patches with a rich variation in pixel values that can be reliably corresponded between the source and target images. We propose to use $33 \times 33$ pixel-neighborhoods (approximately 1% of the image resolution) around the 200 best matching SIFT correspondences [4]. Such correspondences are naturally selected in areas of rich texture, ensuring a rich variety in pixel values in the selected patches. We found that a $33 \times 33$ window offers a good balance between providing sufficient linear equations (Equation 3.4), providing sufficient overlap between the different patches (in terms of pixel values) to tie the different scale factors together, and minimizing the error introduced by the 1D linear subspace approximation (i.e., larger patches exhibit larger approximation errors). Furthermore, we allow different $33 \times 33$ windows to overlap, and include each in the linear system (Equation 3.4) with their respective $\log \kappa_{x_k}$ factor.

To obtain a subpixel accurate alignment and to compensate for any non-linear mapping between the source and target patch, we compute SIFT flow [27] between double sized (i.e., $65 \times 65$) windows in the source and target image, and warp the source patch.
Before Outlier Rejection

After Outlier Rejection

\textbf{Figure 3.6: Outlier Rejection.} Recovered camera response function (red) with corresponding reference camera response function (black) from the Desk example shown in Figure 3.5 before and after outlier rejection. The false color cloud represents the recovered pixel radiance for each pixel in the patches (times the corresponding patch scale factor).

Outlier Rejection

The above algorithm assumes that the change in pixel values in a small patch can be explained by a 1D linear subspace. However, this is not always the case (see section 4.5). Patches that cannot be well represented by the proposed model can adversely affect the quality of the recovered partial camera response function. To remove such outliers, we employ the following two-step strategy. Initially, we compute a candidate camera response function using all patches. We then compute for each patch the fitness of the proposed camera response function:

\[
\epsilon^2(x) = \sum_{x' \in N_t(x)} \left( h_0'(M_t(x')) + \sum_{i=1}^n c_i h'_i(M_t(x')) - \log E_s(\phi_{t\rightarrow s}(x')) - \log \kappa_x \right)^2.
\] (3.5)

to the target patch. We reject cases for which the inner $33 \times 33$ neighborhood contains pixels for which SIFT flow failed to find a corresponding source pixel. We employ a larger $65 \times 65$ window for SIFT flow instead of a $33 \times 33$ window to support shifting (to compensate for misalignments) and scaling (due to differences in camera distance) of pixel values from outside the targeted $33 \times 33$ window.
Next, we reject any patches for which its corresponding fitness \( \epsilon^2(x) \) exceeds \( v \) times the variance, where \( v \) ranges from 2 to 3 depending on how conservative we want outlier rejection to be. Finally, we recompute the camera response function using only the inliers. The key assumption is that the inliers outnumber the outliers, and thus that the initial camera response function can serve as an indicator whether a patch follows the model. Figure 3.6 shows an example of a recovered camera response function (red) compared to a reference camera response function (black) before and after outlier rejection for the Desk example shown in Figure 3.5. Because we can only recover the partial camera response function up to an unknown scale factor, we apply a global scale factor that minimizes the difference between both. Furthermore, we also plot the recovered relative radiance values of each pixel in each patch (times the scale factor \( \kappa_{x_k} \)). Ideally, the recovered relative radiance should fall on the reference camera response function, but instead it forms a “cloud” around the reference camera response function due to the 1D subspace approximation, alignment errors, and camera noise. The (horizontal) extent of the “cloud” depends on the pixel values present in the patches, and thus this scale depends on the scene and camera settings (e.g., exposure).

### 3.4 Validation & Results

**Results** We demonstrate our method on a variety of scenes (Figure 3.10). For each example in Figure 3.10, we show (from left to right): the radiometrically calibrated source image, the resulting radiometric transfer result, a ground truth linearized image, a false color difference image of the former two, and a comparison of the recovered partial camera response function (red) to the ground truth camera response function (black). Pixel values that fall outside the range of the partial camera response function are set to white; we also highlight the discarded pixels in the inset. The reference linearized image is computed by applying the ground truth camera response function to the target uncalibrated image. We compute the ground truth camera response function from an exposure stack using the method of Kim and Pollefeys [28]. For all of our results, we
assume all three color channels share the same camera response function, allowing us
to triple the number of equations per patch. To provide a meaningful qualitative and
quantitative comparison, we optimize for the optimal global scale factor that minimizes
the differences between the reference and recovered camera response function; we also
apply the same global scale factor to the radiometric transfer result.

Both the source and target images for the examples in Figure 3.10 are captured with
a Nikon 700D camera. We briefly summarize the different scenes (from top to bottom; see also Table 3.1):

1. The Wren Building scene exhibits significantly different lighting and viewpoint
   between source (sunrise with strong shadows on the building) and the target image
   (noon on a clear day). The RMSE and relative error of the radiometric transfer
   are 0.000961 and 1.6% respectively.

2. The White House photographs are captured with a greater difference in viewpoint
   than the Wren Building, and under different lighting conditions: overcast sky
   (source) and clear sky (target). The RMSE and relative error are: 0.000708 and
   1.3%.

3. The Store example is captured at approximately the same time of the day, but from
   a different viewpoint. The target image contains a significant portion not visible
   in the source image, which is correctly radiometrically linearized. The RMSE is
   0.002698, and the relative error is 6.2%.

4. The Church is captured at different times of the year (Fall versus Spring). Note

<table>
<thead>
<tr>
<th>Name</th>
<th>Absolute</th>
<th>Relative</th>
</tr>
</thead>
<tbody>
<tr>
<td>Wren Building</td>
<td>0.000961</td>
<td>1.6%</td>
</tr>
<tr>
<td>White House</td>
<td>0.000708</td>
<td>1.3%</td>
</tr>
<tr>
<td>Store</td>
<td>0.002698</td>
<td>6.2%</td>
</tr>
<tr>
<td>Church</td>
<td>0.001638</td>
<td>2.6%</td>
</tr>
<tr>
<td>Desk</td>
<td>0.001827</td>
<td>11.1%</td>
</tr>
<tr>
<td>Magazines</td>
<td>0.002920</td>
<td>6.5%</td>
</tr>
<tr>
<td>Magazines II</td>
<td>0.002214</td>
<td>7.0%</td>
</tr>
<tr>
<td>Flash</td>
<td>0.000585</td>
<td>6.0%</td>
</tr>
</tbody>
</table>
that the target image also contains a parked car not present in the source image. With the exception of the pixels outside the recovered range, this new object’s radiance is recovered correctly. Additionally, this demonstrates that our method can handle significant changes between the input and target scene. The RMSE of the radiometric transfer is 0.001638, and the relative error is 2.6%.

5. The Desk source image is illuminated only by the LCD panel in the background, and the target is illuminated by office lighting. Note that besides differences in lighting and view, the directly visible pixels on the LCD panel are also different. This demonstrates that our method is robust to some degree in change to the scene. The RMSE and relative error of the radiometric transfer result are: 0.001827 and 11.1%. Note, the large relative error is mainly due to the many dark pixels.

6. The Magazines are captured under identical lighting conditions, but from different viewpoints. This result shows that the proposed method also works for non-diffuse materials (i.e., glossy magazine covers). The RMSE of the radiometric transfer is 0.002920, and the relative error is 6.5%.

7. The Magazines II example is captured under similar conditions as the Magazines, except that the order of the magazines is different between the source and target images. The RMSE of the result is 0.002214, and the relative error is 7.0%.

8. The Flash example demonstrates a radiometric transfer from a scene under ambient lighting to a scene illuminated by the camera flash only, illustrating the robustness of our method to drastic changes in lighting. The RMSE and relative error are: 0.000585 and 6.0%.

**Stability** To validate the stability of radiometric transfer, we compute the radiometric linear version of the target image of the Wren Building from two different source images with vastly different view, distance, and lighting conditions. The resulting radiometric linear images are compared in Figure 3.7. We rely on SIFT flow to compensate for the differences in distance and thus global scale of the image features. As can be seen, the recovered transfer from both source images are visually similar, indicating that our
method is robust to different inputs.

Robustness to Camera Model  We validate the robustness of the proposed method with respect to different camera models by simulating the acquisition of the target image for each of the camera response functions in the DoRF database [2] on the Wren Building and Magazines scenes which are shown in Figure 3.10. We then use the proposed radiometric transfer method to linearize the simulated target images, and compare the resulting images with the ground truth reference. The mean and variance of the RMS errors over the different camera response functions are 0.0022 and $4.19 \times 10^{-6}$ for the Wren Building example, and 0.0036 and $8.66 \times 10^{-6}$ for the Magazines example. This shows that the proposed method is robust to a wide variety of camera response functions.

Figure 3.7: Stability Validation. False color difference image between the radiometric transfer results from two different source exemplar images with significantly different lighting and view conditions.
Figure 3.8: Transfer between Different Camera Models. Robustness validation of radiometric transfer between three different camera models: Nikon D700, Canon 600D, and Canon 60D.

Figure 3.8 shows a cross-validation of captured radiometrically linear/non-linear photographs obtained with three different camera models (Nikon D700, Canon 600D, and Canon 60D) of the Wren Building for a wide variety of viewpoints and lighting conditions. As can be seen, our method is able to accurately recover the camera response functions and linearize the target photographs for various combinations of camera-pairs; the relative error is below 3% on all examples.

Robustness to Lighting We validate the robustness of our method with respect to varying lighting conditions using the WILD database \cite{5} which contains a large selection of radiometrically linearized images of an urban scene under a wide variety of weather (and thus lighting) conditions. We select 70 random image pairs from the clear weather subset and simulate capture of the target image with a randomly selected camera response function from the DoRF database. Next, we recover the partial camera response function and radiometrically linearize the target image using the other image as the source. We show four selected pairs and the corresponding recovered camera response functions in Figure 3.9. Of the 70 image pairs, 50 yielded a successful transfer result (Figure 3.9, first three columns). 20 image pairs did not result in a successful transfer; the last column in Figure 3.9 shows such a case. The majority of the failure cases is due
Figure 3.9: Robustness validation of radiometric transfer under different lighting conditions. Four selected results from transfers between simulated captures from the WILD database [5] and a random camera response from the DoRF database [2]. The first three columns show successful transfer results under vastly different lighting conditions between the source and target image. The last column shows a failure case where insufficient SIFT features were found.

to an insufficient number of reliable SIFT matches, mainly caused by large dark regions (and thus little texture) due to shadows. The average and variance of the RMS errors are 0.0082 and $2.79 \times 10^{-4}$ for the full 70 cases, and 0.0053 and $7.55 \times 10^{-6}$ for the 50 successful transfers.

3.5 Discussion

1D Linear Subspace Model  A key assumption in our method is that the change in appearance of a small image patch (viewed from different viewpoints and under different lighting conditions), spans a 1D linear subspace (after unwarping to correct for geometrical distortions). To better understand under which conditions this assumption holds, we express the appearance of a patch’s pixel $E(x, \omega_o)$ in terms of the underlying surface normal $n(x)$, material properties $f_r(x, \omega_o, \omega_i)$, and incident lighting $I(x, \omega_i)$ at
the surface location \(x\) viewed from a direction \(\omega_o\):

\[
E(x, \omega_o) = \int_{\Omega} f_r(x, \omega_o, \omega_i) I(x, \omega_i) \max(n(x) \cdot \omega_i, 0) d\omega_i. \tag{3.6}
\]

Equation 3.6 comes from the rendering equation introduced by James Kajiya in 1986 [29]:

\[
L_o(x, \omega_o) = L_e(x, \omega_o) + \int_{\Omega} f_r(x, \omega_o, \omega_i) I(x, \omega_i) (n(x) \cdot \omega_i) d\omega_i, \tag{3.7}
\]

where, \(L_o(x, \omega_o)\) is the total radiance outward along direction \(\omega_o\) from a particular position \(x\) in space, \(L_e(x, \omega_o)\) is emitted radiance, \(\Omega\) is the unit hemisphere centered around \(n\) containing all possible values for \(\omega_i\), \(f_r(x, \omega_o, \omega_i)\) is the bidirectional reflectance distribution function (the proportion of light reflected from \(\omega_i\) to \(\omega_o\) at position \(x\)), \(\omega_i\) is the negative direction of the incoming light, \(I(x, \omega_i)\) is the radiance coming toward \(x\) from direction \(\omega_i\), and \(n(x)\) is the surface normal (i.e., perpendicular to the surface) at \(x\).

The 1D linear subspace assumption essentially factors the patch’s appearance in a spatially dependent component \(P(x)\) and a position-independent component \(\kappa = \int K(\omega_i, \omega_o) d\omega_i:\)

\[
E(x, \omega_o) \approx P(x) \int_{\Omega} K(\omega_i, \omega_o) d\omega_i. \tag{3.8}
\]

To derive the exact form of the terms \(\kappa\) and \(P(x)\), and to better understand the conditions under which this approximation is valid, we consider each of the three components in Equation 3.6 separately:

- **Material properties** \(f_r(x, \omega_o, \omega_i)\): Assuming that the outgoing direction is constant within a patch, we can factor the material properties in a position dependent albedo function \(\alpha(x)\) and a normalized bidirectional reflectance distribution function \(f(\omega_i, \omega_o)\). While such a factorization is exact for any monochromatic material model such as the common Lambertian surface reflectance model (e.g., with spatially varying albedo), it is only valid for a restricted form of the more general dichromatic surface reflectance model (i.e., a linear combination of diffuse and specular surface reflectance). In particular, this factorization is only valid if relative
ratio of diffuse and specular albedo remains constant over the patch: \( r \equiv \alpha_s/\alpha_d \),
then \( \alpha(x) = \alpha_d(x) \), and \( f(\omega_i, \omega_o) = f_d(\omega_i, \omega_o) + rf_s(\omega_i, \omega_o) \).

- **Lighting** \( I(x, \omega_i) \) can be made position independent by assuming distant lighting:
\( I(x, \omega_i) \approx I(\omega_i) \). Hence, the lighting is the same for all points in a patch. This excludes situations where a shadow edge crosses the patch or other strong position-dependent changes in the incident lighting. However, this does not imply identical incident lighting over all patches.

- **Geometric Term** \( \max(n(x) \cdot \omega_i) \). Except for the case where the surface normal is constant over the patch (i.e., \( n = n(x) \)), the geometric term has both angular as well as positional (surface normal) dependencies. Consequently, the 1D subspace assumption only holds when the surface normals are constant within a patch.

Based on the above analysis, we can refine the terms \( K \) and \( P \) in Equation 3.8 as:

\[
P(x) = \alpha(x),
\]

\[
K(\omega_i, \omega_o) = f(\omega_i, \omega_o)I(\omega_i) \max(n \cdot \omega_i).
\]

While theoretically only valid under the above assumptions, we found that in practice, small deviations from these assumptions can still be resolved in a least squares sense over many patches. Figure 3.5 (bottom-right) shows the approximation error for an office scene under different lighting conditions: for each pixel in the image, we compute the RMS error for the optimal scale factor \( \kappa \) in a \( 33 \times 33 \) pixel-window. The majority of large errors occur in areas with strong deviations from the assumptions such as geometric discontinuities (e.g., edge of blanket) and across shadow boundaries (e.g., monitor stand).

**Limitations** The proposed method relies on a sufficient number of reliable SIFT correspondences that cover the majority of the pixel-intensity range in the target image, and SIFT-flow for fine-scale alignment. For scenes with little texture or for scenes recorded from radically different views or under extremely different lighting conditions, radiometric transfer can fail due to an insufficient number of reliable correspondences or
inaccurate fine-scale alignment.

While our method is theoretically only valid when the surface normal does not vary within a patch, in practice it is robust to minor variations in surface normal (and thus depth) due to the least squares fitting of the partial camera response function. However, our method can fail for scenes that consist mostly of fine details with significant depth discontinuities such as photographs of flower beds.

Additionally, the recovered camera response function is limited to the pixel-intensity range contained in the patches. Clearly, oversaturated and undersaturated pixels cannot be linearized; we exclude pixel-intensities outside the [0.04, 0.96] range for both partial camera response curve recovery as well as linearization. Furthermore, the obtained camera response function might only cover a small portion of the intensity domain. However, we found that our algorithm typically finds a valid response function for most of the pixel range in the target image, and hence even with a partial camera response function, we can still obtain a good radiometric linearization.

Finally, we assume that the camera response function is invariant over the image, and that it is the only non-linear transformation applied to the target image. Other non-linear transformations introduced by chromatic aberrations, edge sharpening or adaptive demosaicing can bias the recovery of the camera response function. We currently, rely on statistical averaging over the various patches to mitigate the impact of such additional non-linear enhancements. However, this is not guaranteed and a skewed error distribution could adversely affect the accuracy of the recovered response function.

### 3.6 Conclusions

In this chapter, we presented a lightweight method for radiometrically linearizing an uncalibrated target image based on an exemplar calibrated photograph of the same scene recorded from a different viewpoint and under different lighting conditions. Key to our method is the observation that for many patches, their change in appearance (from different viewpoints and lighting) forms a 1D linear subspace. This allows us
to reformulate the problem in a form similar to classic radiometric calibration from an exposure stack.
**Figure 3.10:** Radiometric Transfer Results. Left to right: radiometrically calibrated source image, radiance transfer result (pixels outside the recoverable range are marked in white – also highlighted in the inset), reference linearized target image obtained by applying the (inverse) ground truth camera, false color difference, and camera response curve (recovered in red, ground truth in black). In addition, the recovered pixel radiance (times scale factor) for each pixel value is included in the right plot. Ideally this should form a tight ‘cloud’ around the ground truth camera response curve.
Chapter 4

CRF-net: Single Image Radiometric Calibration using CNNs

4.1 Introduction

Many existing radiometric calibration methods require direct control over the capture process (e.g., adding a calibration target \[30\] or capturing an exposure stack \([11, 31, 12]\)). Once the camera response function is known, radiometric linearization (i.e., undoing the non-linear transformation of the camera response function) is trivial. However, with the proliferation of computer vision and computer graphics methods that rely on community datasets without knowledge of the camera or control on the capture process, robust direct single image radiometric calibration has become indispensable.

In the past decade several ingenious single image radiometric calibration methods have been proposed that rely on carefully selected cues such as color mixtures at edges \([32, 10, 33, 34]\), the asymmetry of noise distributions \([25, 35]\), and temporal changes during a single exposure \([36]\). However, these cues are not universally present in all images, and therefore these methods are not practical for large scale image-databases mined from community repositories. Current practice for such large image sets is to gamma correct...
during preprocessing instead of a full radiometric calibration. Gamma correction is a nonlinear image operation (i.e., $f(x) = x^\gamma$) that resembles a camera response curve, and which is used to encode and decode luminance in image display systems. For most computer display systems, images are encoded with a gamma of 0.45 and decoded with the reciprocal gamma of 2.2. While gamma correction is better than directly using tone-mapped pixel intensities, it is still a poor approximation for most camera response functions [34].

In this chapter, we propose a more robust single image radiometric calibration method based on convolutional neural networks (CNN), named CRF-net (or Camera Response Function net). The proposed network takes as input a single photograph, and outputs an estimate of the camera response function in the form of the 11 PCA coefficients for the EMoR camera response model [2]. For training CRF-net, we rely on the DoRF database [2] of 201 measured camera response functions to synthesize a large set of tone-mapped images from a much smaller set of radiometrically linear images. Moreover, we introduce a simple oracle for predicting which image windows are likely to produce good results. We experimentally validate the accuracy and robustness of the proposed CRF-net.

4.2 Background – Convolutional Neural Networks

As noted above, the proposed single image photometric calibration method builds on convolutional neural networks. Convolutional neural networks are a very general and flexible machine learning technique that has recently become a popular tool in computer graphics and computer vision. We therefore first review convolutional neural networks, before detailing our novel single image photometric calibration method.

4.2.1 Neural Network

A Neural Network is currently one of the most popular machine learning techniques; originally designed to simulate the human brain and later applied to implement artificial
Neural networks are composed of a large number of artificial neurons. An artificial neuron is a node containing an activation function, which takes multiple inputs and produces one output. In Figure 4.1, we show an example of an artificial neuron. This artificial neuron takes two inputs \( x_1 \) and \( x_2 \), and produces one output, based on an activation function:

\[
\text{output} = \begin{cases} 
0 & \text{if } \sum_{i=1}^{2} w_i x_i + b < 0 \\
1 & \text{if } \sum_{i=1}^{2} w_i x_i + b \geq 0 
\end{cases}
\]

\( w \) is the weight for each input, \( b \) is the bias for the activation function.

According to this activation function, the neuron’s output is determined by two inputs and \( w, b \). Given certain inputs, the output can be different by varying \( w \) and \( b \). During training process, the neuron can be trained to learn what value \( w \), and \( b \) should be, so that it can generate the desired output. During testing process, \( w \) and \( b \) are fixed, so the output is calculated only based on the two inputs.

Above is a very simple activation function. There are many other different types activation functions, such as Identity function, Sigmoid function, TanH function, ReLU function, Maxout function and etc:

1. Identity function: \( f(x) = x \),

2. Sigmoid function: \( f(x) = \frac{1}{1+e^{-x}} \),
Figure 4.2: Example of an artificial neural network: this neural network contains three layers – one input layer containing two input neurons, one hidden layer containing three hidden neurons, one output layer containing one output neurons.

3. TanH function: $f(x) = \frac{2}{1+e^{-2x}} - 1$,

4. ReLU function: $f(x) = max(0, x)$,

5. Maxout function: $f(x) = max(w_i x + b_i)$.

A simple neural network (Figure 4.2) contains the input layer (the leftmost layer), hidden layer (the middle layer), and output layer (the rightmost layer). Each layer consists of one or more neuron nodes. Information flow is transferred through the connections from one layer to another.

For a normal neural network, the hidden layers can contain one, two or many layers, depending on how large the architecture should be based on the complexity of the problem. When the size and number of layers increases, the space of functions the neural network can represent also grows. So larger neural networks can express more complex functions. However, if a large neural network is used, we have to make sure enough training examples are collected, otherwise, overfitting can happen for a large number of parameters fitting a small set of training data.

\[^1\text{http://cs231n.github.io/neural-networks-1/}\]
4.2.2 Convolutional Neural Network

A Convolutional Neural Network (CNN) is a special neural network designed mainly to process image data. It has become a very successful and popular type of neural network in practical image-based computer vision and computer graphics problems since Krizhevsky et al. won the ImageNet Large Scale Visual Recognition Competition (ILSVRC).\footnote{www.image-net.org/challenges/LSVRC/}

Convolutional Neural Networks and regular Neural Networks are both composed of neurons with learnable weights and biases. The key difference is that Convolutional Neural Networks share weights and biases for neurons in the same layer.

To illustrate why a convolutional neural network is essential when dealing with image data, consider the following problem: if we take an image of reasonable size as example, such as $256 \times 256 \times 3$, the total number of weights for just the first layer will be $256 \times 256 \times 3 = 196,608$. This huge number of parameters will lead to overfitting. Also, for a natural image, neighboring neurons are likely to perform operations in the same way, so it makes sense to share the neuron structure and parameters for small neighborhoods.

A simple convolutional neural network is a sequence of different types of layers. We will introduce three main types of layers to build a convolutional neural network: Convolutional layer, Pooling Layer, and Fully-Connected Layer.

1. Convolutional Layer

- **receptive field** In a normal Neural Network, each input neuron is connected with each hidden or output neuron. In a convolutional neural network, each hidden neuron of a layer is connected to a local small region of the input neurons which represent pixels in a small neighborhood. This small region is called a receptive field. As shown in Figure 4.3, each $5 \times 5$ region of input neurons is connected to one hidden neuron. All $5 \times 5$ regions for all input neurons share the same weight and bias when connecting with the hidden neurons. This $5 \times 5$ window is also called filter or kernel in CNN terminology.
Figure 4.3: Example of a receptive field: each $5 \times 5$ region of input neurons is connected to one hidden neuron. All $5 \times 5$ regions for all input neurons share the same weight and bias when connecting with the hidden neurons.

- **spatial arrangement** When we move the receptive field to the right in steps of one over the entire input neuron range Figure 4.4, and suppose the width of input neurons is $N$ and the width of receptive field is $W$, then the final width of hidden layer is $(N - W) + 1$. If for each move, we take two steps instead of one Figure 4.5, then the final width of hidden layer will be $(N - W)/2 + 1$. This step size is also called stride.

If input width is 32, the width of receptive field $W$ is 5, stride is 1, the output width will be 28. Consequently the spatial dimensions decrease. If we want to keep the same size, we can pad the input neurons with zeros at the ends, so that the receptive field can be applied up to and including the edge element of the input neurons, as shown in Figure 4.6. This method is called zero-
padding. Suppose the size of zero-padding is $P$, then the final width of output size will be $(N - W + 2P)/\text{stride} + 1$.

Since the input is always an image, suppose the input image size is $N \times N \times 3$ (3 is the channel number), and the number of filters is $F$, the output will be $((N - W + 2P)/\text{stride} + 1) \times ((N - W + 2P)/\text{stride} + 1) \times F$. The number of filters is called depth in CNN.

- **parameter sharing** Since we are using receptive fields in CNNs, all neurons in the hidden/output layer detect exactly the same feature for different locations in the input layer. Therefore the mapping from one layer to another is called a feature map. The weights for one feature map are called shared weights. The bias for a feature map is called shared bias. This parameter-sharing approach reduces the number of parameters for a convolutional neural network greatly,
Figure 4.5: Example of stride 2: for each move, the step size is 2.

thus avoids the problem of overfitting.

2. Pooling Layer

Pooling layers are often inserted right after convolutional layers. Its main purpose is to reduce the size of the output from the convolutional layer. The most common two types of pooling layers are

Max Pooling  A $2 \times 2$ Max Pooling Layer (Figure 4.7) takes the maximum number over $2 \times 2$ region as output.

Average Pooling  A $2 \times 2$ Average Pooling Layer (Figure 4.8) takes the average value of $2 \times 2$ region as output.

3. Fully-Connected Layer
Figure 4.6: Example of zero-padding: the original size is $32 \times 32 \times 3$, and after padding 2 zeros on the border, the input size is $36 \times 36 \times 3$. Then, applying a convolutional layer of $5 \times 5 \times 3$ with stride 1, we can still get an output size of $32 \times 32$.

The last layer in a CNN is often a fully-connected layer. It connects each input neuron with each output neuron, similar to a regular Neural Network. The output can be computed as a matrix multiplication of the input activation with weights followed by the corresponding bias. Figure 4.9 shows an example of a fully-connected layer.

4.2.3 Related Work

There exist many different architectures of Convolutional Neural Networks. We will discuss the most popular and common image processing networks in computer graphics and computer vision.
1. **LeNet**

LeNet, introduced by Yann LeCun in 1994, is one of the first convolutional neural networks designed to read zip codes or digits. It uses 3 types of layers: convolutional layer, average pooling layer and fully-connected layer, as shown in Figure 4.10.

2. **AlexNet**

AlexNet designed by Alex Krizhevsky in 2012 is one of the most successful convolutional neural networks which lead to a breakthrough of accuracy in image classification.

AlexNet extends LeNet to a deeper and larger convolutional neural network which can recognize larger and more complex images other than just zip codes or digits. It adds: a ReLU layer to better handle non-linearities, a Max Pooling layer instead of Average Pooling layer, and a dropout method to avoid overfitting (Dropout is a technique to selectively ignore several neurons during training to combat the problem of overfitting to the training data.)

3. **VGGNet**
VGGNet was proposed by Karen Simonyan and Andrew Zisserman in 2014, which won the ILSVRC 2014. It uses $3 \times 3$ convolution layers instead of larger convolutional layers such as $5 \times 5$ or $7 \times 7$. On the other hand, it also increases the total number of layers to 16 or 19. We can think of the change from AlexNet to VGGNet as changing a single $7 \times 7$ layer to multiple $3 \times 3$ layers. By doing this, we gain a more discriminative decision function since we incorporate multiple non-linear rectification layers instead of a single one. Also, the number of parameters are decreased: a single $7 \times 7$ convolutional layer would require $49C^2$ parameters (assuming that the channel size for input and output is $C$); three $3 \times 3$ convolutional layers for the same channel size require only $3 \times (3 \times 3 \times C^2) = 27C^2$ parameters.

4. ResNet

ResNet was designed by Kaiming He et al. in 2015, which won the ILSVRC 2015. It builds on a very simple but effective idea: feeding input neurons to next layers by skipping current connections.

Network depth is of great importance, however, deep models lead to the problem
of vanishing/exploding gradients. To resolve this problem, Kaiming He et al. add identity mapping layers to construct a deep residual learning framework. Adding identity mapping layers can effectively avoid gradients going to zero, thus solving vanishing gradients problem.

ResNet is a much deeper neural network [Figure 4.13], compared with all previous popular Convolutional Neural Networks. But it is still easy to optimize and gains accuracy from considerably increased depth.
Figure 4.11: AlexNet: Architecture of AlexNet, a Convolutional Neural Network containing 5 convolutional layers, max-pooling layers, dropout layers, and 3 fully-connected layers. Two streams show that the training process is split onto 2 GPUs. (Image from [7]).

4.3 CRF-net

4.3.1 CRF-net Architecture

CRF-net follows the ResNet-18 architecture [9] from the DeepDetect library [1] implemented in Caffe [37]. This architecture differs from the 18-layer ResNet introduced by He et al. [9]; DeepDetect’s ResNet-18 architecture is a cut-out of He et al.’s 50-layer ResNet-50. We opt for this architecture because we expect only local pixel relations (e.g., edge information) to inform radiometric calibration. Hence, a shallow network with small filters should suffice. Table 4.1 summarizes the architecture. We also experimented with other network architectures such as VGGnet [8] and AlexNet [7], but these architectures did not produce good results. We add a fully connected layer on top of ResNet-18 that outputs the 11 PCA coefficients of the EMoR model. While Grossberg and Nayar [2] report that 3 PCA coefficients already cover 99.5% of the energy, we opt to use 11 PCA coefficient as this produces nearly perfect matches on the most challenging camera response functions in the DoRF database ([2], Fig. 7).

CNNs are often restricted to input images of limited resolution. Likewise, the proposed CRF-net also only operates on $227 \times 227$ pixel windows. However, radiometric calibration typically deals with much larger images. We therefore select 10 well-chosen $227 \times 227$ windows from the input image, and aggregate the corresponding es-
Figure 4.12: VGGNet: Architecture of VGGNet, a deep Convolutional Neural Network made up with small filters. The depth of the configurations increases from the left (A) to the right (E), as more layers are added (the added layers are shown in bold). (Image from [8]).

Table 2: Number of parameters (in millions).

<table>
<thead>
<tr>
<th>Network</th>
<th>A,A-LRN</th>
<th>B</th>
<th>C</th>
<th>D</th>
<th>E</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of parameters</td>
<td>133</td>
<td>133</td>
<td>134</td>
<td>138</td>
<td>144</td>
</tr>
</tbody>
</table>

Note that we cannot simply scale the input images to a smaller resolution because the non-linearity of the camera response function would destroy the relation be-
Figure 4.13: ResNet: Example network architectures for ImageNet. Left: the VGG-19 model. Middle: a plain network with 34 parameter layers. Right: a residual network with 34 parameter layers. (Image from [9]).
<table>
<thead>
<tr>
<th>Output Size</th>
<th>Configuration</th>
<th>Short-cut</th>
</tr>
</thead>
<tbody>
<tr>
<td>114 × 114</td>
<td>[7 × 7 × 64], stride 2</td>
<td></td>
</tr>
<tr>
<td>57 × 57</td>
<td>max pool 3 × 3, stride 2</td>
<td></td>
</tr>
</tbody>
</table>
| 57 × 57    | 1 × 1 × 64
3 × 3 × 64
1 × 1 × 256 | ×1 [1 × 1 × 256] |
| 57 × 57    | 1 × 1 × 64
3 × 3 × 64
1 × 1 × 256 | ×2 identity |
| 29 × 29    | 1 × 1 × 128
3 × 3 × 128
1 × 1 × 512 | ×1 [1 × 1 × 512] |
| 29 × 29    | 1 × 1 × 128
3 × 3 × 128
1 × 1 × 512 | identity |
| 23 × 23    | average pool 7 × 7, stride 1 |           |
| 11         | fully connected       |           |


Between the (averaged) pixel intensities and the corresponding (averaged) scene irradiance: $I_1 + I_2 = g(M_1) + g(M_2) \neq g(M_1 + M_2)$. However, an ill-chosen $227 \times 227$ window (e.g., covering only the sky) will also not produce good results. We posit that a “good” window should cover a large range of pixel intensities. We therefore repeatedly select and test random candidate windows, until we have found 10 windows whose intensity histograms (256 bins, mixing red, green, and blue intensities) contain at least 220 non-empty bins each. If after a certain number of attempts no such windows are found, then we select the 10 windows that best covered the intensity range. However, in such a case we expect a suboptimal radiometric calibration. Finally, we aggregate the estimated camera response functions from the 10 well-chosen windows, by removing the outliers and averaging the the PCA coefficients of the remaining estimated camera response functions.

4.3.2 Training

Radiometric calibration is significantly different from other problems, such as object recognition, intrinsic decomposition, etc., on which CNNs have successfully been applied. Therefore, we cannot refine an existing network. Consequently, we are forced to train
CRF-net from scratch, and thus we require an extensive training dataset. Obtaining a large dataset of photographs for a large variety of scenes and capture conditions, together with corresponding ground truth camera response functions from a diverse set of camera models, is time-consuming and difficult. Instead we follow the recent trend of using synthetic training data.

We have collected a set of 595 well-exposed radiometrically linear “RAW” photographs, captured with 3 different camera models (i.e., Canon EOS 600D, Nikon D800, and Nikon D300S), from a variety of scenes (approximately 60% indoor scenes and 40% outdoor scenes) captured under a variety of conditions (e.g., clear sky, overcast sky, night time, etc.). From this set of radiometrically linear images, we generate corresponding tone-mapped photographs for each of the 201 camera models in the DoRF database [2].

To reduce storage requirements and minimize disk overhead during training, we scale the radiometrically linear image first by an integer factor such that the smallest dimension is just larger than 227. We deliberately only apply an integer scale factor such that each image pixel is only assigned to a single tone-mapped image pixel. Furthermore, since CRF-net requires $227 \times 227$ pixel windows (and we cannot scale tone-mapped images), we select 10 well-chosen pixel windows using the same intensity criterion as detailed in subsection 4.3.1. Furthermore, we desire to train CRF-net for reasonably exposed images, such as those produced by using the auto-exposure function on a consumer camera; severely underexposed or overexposed image are unlikely to contain sufficient information to retrieve the camera response function and/or to extract any meaningful image information. We therefore precompute for each camera response curve in the DoRF database a scale factor ‘s’ that generally produces well-exposed images, roughly approximating the effect of ‘auto-exposure’. To further avoid biasing CRF-net to relate overall brightness and the camera response function, we produce 5 slightly different exposed versions by randomly sampling an effective exposure in the range: $[s - 0.4, s + 0.4]$. In total, our training dataset consists of $595 \times 201 \times 10 \times 5 = 5,979,750$ image windows with corresponding camera response functions.
In addition to the training dataset, we also generate a validation dataset, but using a different set of 20 radiometrically linear RAW images captured with 3 different camera models, of which one is shared with the training dataset (i.e., Canon EOS 600D), and two are new camera models (i.e., Nikon D700 and Canon EOS 60D). We generate synthetic photographs from this set of 20 images using again all 201 camera models from the DoRF database and with 5 different exposures selected in a similar fashion as for the training dataset.

As noted before, we train CRF-net from scratch. However, we found that directly training CRF-net is difficult. We therefore first train a slightly different variant that instead of outputting the PCA coefficients, outputs a likelihood that a photograph was generated by each of the camera response functions in the DoRF database (i.e., a classification network where the fully connected layer outputs 201 likelihoods instead of 11 PCA coefficients). Due to the similarity of many camera response functions in the DoRF database, the accuracy of this classification network is poor (only 26% of the photographs are correctly classified). However, it serves as an excellent starting point to refine the full CRF-net. We train the classification network using the following hyperparameters: learning policy “step”, base learning rate of 0.01, a step size of 500,000, 2,000,000 maximum number of iterations, momentum 0.9, and a weight decay of 0.0005. After convergence, we replace the fully connected layer of the classification network, and copy the trained CNN parameters. We use the same training images and hyperparameters to refine CRF-net from the classification network, except for: base learning rate 0.0001, step size 20,000, and maximum number of iterations 25,000.

4.4 Results

We will employ two kinds of error metrics to gauge the accuracy of the recovered camera response functions. The “estimation error” is defined as the L2 distance between two curves. As we represent the camera response functions using the EMoR PCA model, we simply use the L2 distance between the corresponding PCA coefficients. While the
Figure 4.14: Three examples of photographs and corresponding ground truth (purple) and estimated (green) camera response functions. The estimation errors are: 0.326, 0.267, 0.491, and the linearization errors are ($\times 10^{-5}$): 0.902, 0.423, 2.293.

estimation error indicates how similar both camera response curves are, it does not take in account whether the whole range is meaningful with respect to the target image. For example, the error outside the range of pixel values present in the image has little influence on the accuracy at which the image can be radiometrically linearized. We therefore also consider the “linearization error” that is defined as the RMSE between the images linearized by the ground truth and estimated camera response functions.

Figure 4.14 shows 3 images generated by applying a camera response function from the DoRF database to a radiometrically calibrated image not part of the training dataset. In addition we show the ground truth and recovered camera response functions which are a close match. When the image contains many oversaturated pixels or a large contrast, it becomes difficult to find many good windows (Figure 4.15), resulting in a less accurate radiometric calibration. Depending on the application, the resulting camera response functions and/or radiometrically linearized images might still be of sufficient quality. Over the full validation dataset, the average estimation and linearization error are 1.607 and $2.544 \times 10^{-5}$ respectively.
Figure 4.15: Examples of suboptimal radiometric calibration. The left image exhibits many oversaturated pixels, whereas the right exhibits a very high contrast. In both cases, it is difficult to find good windows that sufficiently (and uniformly) cover the full pixel range. The respective estimation (and linearization ($\times 10^{-5}$)) errors are: 2.367(5.701) and 3.927(10.12).

4.5 Discussion

CRF-net only operates on a small $227 \times 227$ window, and the content of a window greatly affects the quality of the radiometric calibration. While we aggregate the estimates from 10 windows, it is still interesting to know what kind of windows provide good estimates, and how effective our selection criterion works in practice. Figure 4.16 compares the camera response curves estimated from a randomly selected window (marked in red) and a window that matches our selection criterion (marked in green). As expected, the random window that exhibits little pixel variations does not provide sufficient cues to estimate an accurate camera response function.

To better understand the limitations of CRF-net, we furthermore validate its robustness against the following factors: variations in exposure, image/feature scale, color
Figure 4.16: Estimated camera response curves from a single window: a randomly selected one (red) and one selected with the proposed selection criterion.

vs. grayscale, measurement noise, sharpness/blur, and camera models and scenes that significantly differ from those in the training database.

<table>
<thead>
<tr>
<th>Exposure</th>
<th>0.6</th>
<th>0.8</th>
<th>1.0</th>
<th>1.2</th>
<th>1.4</th>
</tr>
</thead>
<tbody>
<tr>
<td>Estimation</td>
<td>1.621</td>
<td>1.611</td>
<td>1.607</td>
<td>1.647</td>
<td>1.676</td>
</tr>
<tr>
<td>Linear. ($\times 10^{-5}$)</td>
<td>3.099</td>
<td>2.836</td>
<td>2.544</td>
<td>3.386</td>
<td>3.631</td>
</tr>
</tbody>
</table>

Table 4.2: Estimation and linearization errors over the validation dataset for different exposures scaled relatively with respect to the 'ideal' auto-exposure.

<table>
<thead>
<tr>
<th>Noise $\sigma^2$</th>
<th>0</th>
<th>0.5</th>
<th>1</th>
<th>2</th>
<th>4</th>
</tr>
</thead>
<tbody>
<tr>
<td>Estimation</td>
<td>1.607</td>
<td>2.302</td>
<td>3.036</td>
<td>4.726</td>
<td>6.069</td>
</tr>
<tr>
<td>Linear. ($\times 10^{-5}$)</td>
<td>2.544</td>
<td>4.069</td>
<td>5.017</td>
<td>11.4</td>
<td>59.99</td>
</tr>
</tbody>
</table>

Table 4.3: Estimation and linearization error over the validation dataset for different amounts of normal distributed camera noise.

Exposure: We scale the radiometrically linear input images of our validation set by $[0.6, 0.8, 1.0, 1.2, 1.4]$ and compute the evolution of the estimation and linearization errors for different exposure scales (Table 4.2). From this we conclude that CRF-net is robust for moderate deviations from the optimal exposure, as long as there are windows that cover a sufficiently large range of pixel intensities uniformly. Unless severe, oversaturation only affects local regions and thus we can still find good windows for recovering
the camera response function. Undersaturation, on the other hand, typically affects the overall brightness of the whole image, making it difficult to find good windows. Consequently, CRF-net is more sensitive to undersaturation. Camera response functions that tend to overly boost the contrast of the image (e.g., Figure 4.15, right) suffer from a similar problem as undersaturation. Unlike undersaturation, there exist windows that fulfill our selection criteria. However, the histograms for these windows exhibit a severely skewed distribution, and thus provide insufficient information for certain regions of the intensity range to reliably estimate the camera response function.

**Scale:** To ensure CRF-net is not overtrained for a specific image-feature size, we compute the estimation and linearization error on the validation dataset, upscaled by a factor: 2 and 4 (applied before tone-mapping). Note, that the original captured images are downscaled by at least a factor 8, and hence we can, without loss of image information, generate higher resolution versions. The average estimation (and linearization) errors \(1.864 \times 10^{-5}\) and \(2.144 \times 10^{-5}\) for 2\(\times\) and 4\(\times\) respectively) are similar to the unscaled errors \(1.607 \times 10^{-5}\). Note that the linearization error is not resolution independent. From this experiment, we observe that scaling slightly impacts the accuracy.

**Grayscale:** Inspired by Lin et al. [33], we also validate whether CRF-net requires colored input. By removing the color information, we also remove a significant amount of the intensity variation in the image. As a result, over and undersaturation effects (including those of contrast enhancing camera response functions) are amplified, and affect the accuracy of the calibration more severely. Figure 4.17 shows a comparison between two different response functions applied to the color and grayscale version of two images. In both cases, a successful calibration is achieved for the color images. However, the calibration on the grayscale versions of the same images with the same camera response function is bimodal: succeeding in one case without loss of accuracy, and failing on the second case. The average estimation (linearization) error on the
Figure 4.17: Radiometric calibration of colored versus grayscale images. Grayscale images exhibit less variation in intensity distributions, and are therefore less robust to calibrate. The top row shows a successful calibration for both color and grayscale; the bottom row shows an example where radiometric calibration on a colored image succeeds, but fails on the same grayscale image.

validation set are $1.607 \times 10^{-5}$ for the color input, and $3.546 \times 10^{-5}$ for the corresponding grayscale versions.

Measurement Noise: Inspired by prior work that exploits the symmetry of noise distributions [25, 35], we also validate the robustness of CRF-net with respect to noise. For each image in the validation dataset, we add normal distributed noise before applying the camera response curve. Table 4.3 shows the respective errors for increasing noise variances. These results show a gradual degradation of the calibration accuracy for increasing magnitude of camera noise. In general we observe that when the noiseless calibration is very accurate, camera noise impacts the radiometric calibration to a lesser
degree than for cases where the noiseless calibration is less accurate.

<table>
<thead>
<tr>
<th>Blur $\sigma^2$</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>4</th>
<th>8</th>
</tr>
</thead>
<tbody>
<tr>
<td>Estimation</td>
<td>1.607</td>
<td>1.728</td>
<td>2.052</td>
<td>2.219</td>
<td>2.335</td>
</tr>
<tr>
<td>Linear. ($\times 10^{-5}$)</td>
<td>2.544</td>
<td>3.422</td>
<td>4.116</td>
<td>4.325</td>
<td>4.259</td>
</tr>
</tbody>
</table>

**Table 4.4**: Estimation and linearization error over the validation dataset for different amounts of blur.

**Image blur**: Depending of the aperture setting, or motion in the scene, certain parts of the image might be blurred. To validate the robustness against blur, we apply differently sized blur filters to the radiometrically linear validation images before applying the camera response function (Table 4.4). From this experiment we can conclude that our method is not sensitive to moderate amounts of blur, and robust to strong blurring. This seems to suggest that CRF-net only weakly relies on edge information (in contrast to [32, 10, 33]).

**Camera Model & Post-processing**: All our training data is synthetically generated from radiometrically linear photographs captured using 3 different cameras. This raises the question whether CRF-net is overtrained to the characteristics (e.g., noise) of the image sensors in these cameras. Furthermore, all our synthetically generated images lack the typical post-processing steps camera manufacturers apply to make the photograph “look good”. This also raises the question whether CRF-net is robust to such post-processing steps. We validate its robustness to these issues by demonstrating the recovery of the camera response functions from 4 well-exposed tone-mapped photographs (i.e., not synthesized from a radiometrically linear input image) with known camera response functions. Since we directly use the camera-produced tone-mapped photographs as an input, unknown post-processing is included. Furthermore, none of the camera models is present in the DoRF database. As demonstrated in Figure 4.19, CRF-net exhibits a similar performance on post-processed non-synthetic photographs as on the synthesized images in the training and validation datasets.
Figure 4.18: Comparison to between Lin et al.’s single image radiometric calibration method [10] and CRF-net on an image for which the former works well.

Our experiments show that CRF-net can robustly estimate the camera response function under a wide range of conditions. A fair comparison to prior work is difficult as it is easy to find examples on which prior single image radiometric calibration methods fail. Nevertheless, even a partial comparison is still instructive to better understand the advantages and limitations of CRF-net. **Figure 4.18** compares the estimated camera response function using CRF-net with that obtained using the method of Lin et al. [10] on a carefully selected photograph for which the latter works well; we found that Lin et al.’s method did not perform well for many examples in our validation set. This example demonstrates that under conditions favorable to Lin et al.’s method, the proposed CRF-net produces comparable or better results.

Currently, for linearizing large image datasets, a simple but robust gamma correction is often favored instead of existing advanced single image radiometric calibration methods. To compare the accuracy of CRF-net to gamma correction, we compute the estimation (and linearization) error using both methods on the validation dataset. The average errors are $1.607 \times 10^{-5}$ for CRF-net versus $3.132 \times 10^{-5}$ for gamma correction. The error on CRF-net was lower in $78\%$ ($86\%$) of the examples in the validation set. This clearly demonstrates that CRF-net is a robust and more accurate alternative to gamma correction.
Figure 4.19: Results from CRF-net applied to captured (tone-mapped) photographs obtained with different camera models (Canon 600D, Nikon D700, Nikon D750), and a single well-exposed photograph from an exposure stack [11]. The respective estimation (and linearization ($\times 10^{-5}$)) errors are: 2.1758 (0.0332), 0.4221 (0.0239), 0.7517 (0.0606), and 1.7897 (0.3951).

4.6 Conclusion

In this chapter we presented a CNN-based solution for radiometric calibration from a single input photograph. We have experimentally verified the robustness of CRF-net for a wide range of conditions. We believe CRF-net can serve as a valuable pre-processing step for computer vision algorithms that require a linear relation between pixel intensities and scene radiance on large datasets mined from uncalibrated repositories.
Chapter 5

Conclusion

In this dissertation, we proposed solutions to implement data-driven radiometric photo- linearization.

First, we presented a lightweight method for radiometrically linearizing an uncalibrated target image based on an exemplar calibrated photograph of the same scene recorded from a different viewpoint and under different lighting conditions. Key to our method is the observation that for many patches, their change in appearance (from different viewpoints and lighting) forms a 1D linear subspace. This allows us to reformulate the problem in a form similar to classic radiometric calibration from an exposure stack.

Second, we presented a CNN-based solution for radiometric calibration from a single input photograph. We have experimentally verified the robustness of CRF-net for a wide range of conditions. We believe CRF-net can serve as a valuable pre-processing step for computer vision algorithms that require a linear relation between pixel intensities and scene radiance on large datasets mined from uncalibrated repositories.

As future work, we will work on the following aspects:

First, we would like to improve the robustness for our single-image radiometric calibration algorithm. For images captured under extreme lighting conditions or under vastly different weather conditions, we want to obtain a more robust estimation of the camera response functions.

Second, for photographs downloaded from large online photo collections, we would
like to extract information about camera parameters which are used to captured those photos, and use this information to improve the accuracy of radiometric calibration.

Last, we would like to explore applications that build on top of radiometric calibration, such as composing HDR images or videos from photos captured at the same scene from large photo collections, or relighting images captured at different times.
Bibliography


