An Empirical Camera Model for Internet Color Vision

BMVC 2009 Submission # 364

Abstract

Images harvested from the Web are proving to be useful for many visual tasks, including recognition, geo-location, and three-dimensional reconstruction. These images are captured under a variety of lighting conditions by consumer-level digital cameras, and such cameras have color processing pipelines that are diverse, complex, and scene-dependent. As a result, the color information contained in these images is difficult to exploit. In this paper, we analyze the factors that contribute to the color output of a typical camera, and we explore the use of parametric models for relating these output colors to meaningful scenes properties. We evaluate these models using a database of registered images captured with varying camera models, camera settings, and lighting conditions.

1 Introduction

The increasing availability of large online photo collections is enabling new approaches to difficult vision problems. Internet photo collections are often associated with significant textual metadata, and this provides an impressive data source that can be exploited to design, build, and evaluate vision systems. We have already seen "Internet vision" approaches to three-dimensional reconstruction; image-based rendering; face, object, and scene recognition; camera calibration; geo-location; and content-based image retrieval. And this basic idea seems likely to spread in the future.

The vast majority of online images are captured in color, and most of those are from consumer-level cameras. These cameras output intensity values that are nonlinearly related to spectral scene radiance, and for many visuals tasks—including image matching, recognition, color constancy, and any sort of photometric analysis—we can benefit from compensating for these nonlinear effects.

Neutralizing the nonlinearities of consumer cameras is difficult because their processing pipelines are trade secrets. A consumer camera succeeds by producing images that are visually pleasing when viewed on small-gamut, low-dynamic-range displays, and doing this well requires complex, scene-dependent color adjustments that sacrifice physical accuracy.

The goal of this paper is to determine an efficient representation for the color processing pipelines of consumer-level digital cameras. We seek a parameterized map that takes spectral radiance distributions to output color vectors in a standard nonlinear color space (sRGB), and we want this map to be "efficient" in the sense of being complex enough to accurately model real cameras but simple enough for use by vision systems. Discovering such a map requires a phenomenological approach, and accordingly, we acquire and study a database of registered images from varying camera models, camera settings, and lighting conditions.

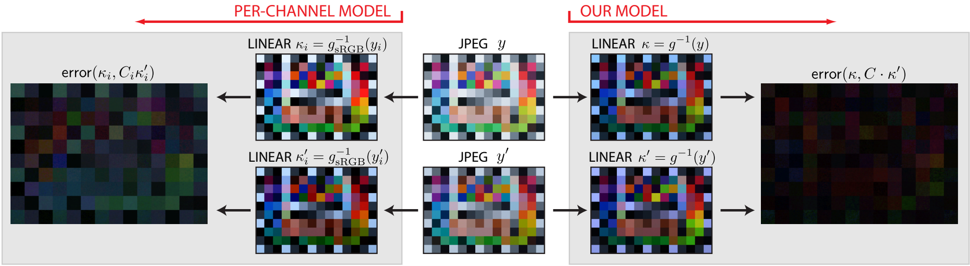


Figure 1: Modeling camera color processing. Pixel intensities are commonly assumed to be standard sRGB maps of spectral image irradiance, i.e., white-balanced linear RGB with standard per-channel nonlinearity. Alternatively, the proposed model accounts for different cameras having different spectral sensors and different nonlinear maps. *Middle*: Two JPEG images of the same scene under the same illuminant captured by different cameras; we seek to match their colors. *Left*: Images matched using the standard sRGB model with per-channel gain leads to high residual error (RMSE: 19.5 gray levels). *Right*: The proposed model properly accounts for variations across cameras, and achieves higher accuracy (RMSE: 7.5 gray-levels).

Our analysis suggests that a twenty-four parameter model is sufficient for most cameras in wide-gamut scenes, and that fixed per-channel nonlinearities—as used in traditional radiometric camera calibration [26]—are often inadequate (Fig. 1). Implications of our model for vision systems are described in Section 5.

2 A camera model

We start by examining the factors affecting the color-vector y that is stored at one pixel of a typical image file harvested from the Web. Our goal is to develop a forward color imaging model that is simple enough to be used for inverse vision problems, and to achieve this goal we are willing treat many secondary effects as unspecified "noise" and ignore them. To simplify the following discussion, we assume y to be in a standard three-primary output color space (sRGB [28]) with JPEG encoding, but our basic approach generalizes to arbitrary encodings and output color spaces.

Consider a small static surface patch that projects to a single elementary pattern in a camera's color filter array (e.g., a GRGB block of a Bayer filter). We restrict our attention to opaque materials and assume that the observation scale is appropriate for the patch's appearance to be accurately summarized by a spectral bi-directional distribution function (BRDF). In this case, the spectral irradiance e incident on the sensor plane depends on the orientation of the patch, the spectral and angular distributions of its local lighting hemisphere, and the position and optics of the camera. This spectral irradiance is sampled by a small number of spectral filters. (Again, we assume three spectral sensors for simplicity, but four-sensor devices are not uncommon, and our model handles these without difficulty.) We summarize this process as

$$\kappa(\ell, v) = \pi \cdot e(\ell, v), \tag{1}$$

where ℓ represents the spectral and angular distributions of the lighting, v represents the viewing direction and optics, and the operator π represents transmission and sensing through the camera's three spectral filters. We assume this process to be linear, which is justified by the experimental results in Sect. 4. In an increasing number of digital cameras, the data κ can be accessed through a RAW output format, and in the sequel, we refer to κ as *linear data* because it is linearly related to image irradiance. It is important to remember, however, that Eq. 1 is an approximation to a camera's RAW output, which may also include the effects of dark current compensation, flare removal, filling/marking of "dead pixels", quantization, and noise removal [20, 27]. Here, we consider these as sources of noise and ignore them.

In the next stages of the camera processing pipeline, the linear data κ is used to render an image in a output color space (sRGB) that is suitable for display purposes [20, 27]. First, there are "pre-processing" operations such as flare and noise removal (if not already done for the RAW data), white balancing, demosaicing, sharpening, and a linear transformation to an internal working color space. As above, we model most of these effects as a noise process. One exception is white balance, which we model as a scene-dependent linear transform C. The scene-dependence results from the transform being determined by an "estimated illuminant" or "chosen white point" that is output from a computationally-efficient color constancy algorithm, such as a variant of gray world [2]. The other exception is the color space transformation, which we model as a fixed linear transform that maps three-vectors in the camera's sensor space (i.e., in terms of the three spectral sensitivities) to colorimetric tristimulus values (say, CIE XYZ) where tone adjustment is applied. Note that a camera's spectral sensors are generally not exact linear combinations of the human standard observer's; so this linear map is approximate in the sense of producing colorimetric tristimulus values that are slightly different from what the standard observer would have measured in the same scene. (We evaluate this difference experimentally in Sect. 4.) For notational convenience, we absorb the fixed linear color-space transform into the white-balance transform C.

The subsequent stage of the pipeline is the most important to our model, and it is also the most mysterious. At this stage, the camera modifies the tristimulus values so that they fit within the limited gamut and dynamic range of the output color space, and it does so in a way that is most visually pleasing (as opposed to most accurate). Referred to as *color rendering*, this is a proprietary art that may include luminance histogram analysis, corrections to hue and saturation, and even local corrections for things such as skin tones. In most cases, this nonlinear color rendering process is scene-dependent and is not a fixed property of a camera. Finally, at the end of the processing pipeline, the rendered image is encoded via re-quantization and compression (usually JPEG), which we again treat as noise and ignore.

Putting this all together, we write the output color vector as

$$y(\ell, v) = g(C \cdot \kappa(\ell, v)), \tag{2}$$

with $C \in GL(3)$ as described above, and $g\colon \mathbb{R}^3 \to \mathbb{R}^3$ a nonlinear function. Note that both C and g depend on global image properties, and that g is a composite of the camera's scene-dependent color rendering processes and the standard compressive nonlinearity (approximately a "gamma" of 2.2) that is part of the sRGB representation.

The remainder of this paper is devoted to evaluating the accuracy of the model in Eqs. 1 and 2. We are particularly interested in developing an efficient representation for the scene-dependent nonlinearity g.

3 Applications and related work

Before describing our experiments, it is worth considering the potential utility of the model we propose. There are at least three broad categories of applications, and while we do not explicitly consider these applications in this paper, they influence the form and complexity of the model we develop.

Radiometric calibration. Many vision algorithms require accurate measurements of scene radiance to succeed; high-dynamic-range imaging, photometric stereo, shape from shading, and reflectometry are but a few examples. For these algorithms to be effective for a given input y, it is desirable to first "undo" the effects of the nonlinearity by computing $x = g^{-1}(y) = C \cdot \kappa$. If this can be achieved, the resulting image values x are linearly related to image irradiance and (assuming one compensates for optical effects like vignetting) scene radiance, and the algorithms described above can be applied directly.

To accomplish this task, one must have access to a low-parametric model for g as well as an algorithm for estimating its parameter values from image data. This is similar to the problem of "grayscale" radiometric calibration [5, 25, 26], which has received significant attention [6, 15, 21, 22, 24]. In fact, our work draws inspiration from Grossberg and Nayar's empirical study of that problem [16]. In much of this work, it is assumed that the nonlinearity (sometimes called the *radiometric response function*) is a fixed property of the camera, and in some cases, this has been extended to handle color by computing separate (and fixed) nonlinearities for each color channel (e.g., [21, 26]). For the reasons described above, a fixed per-channel nonlinearity is unlikely to accurately model the functions g in Eq. 2 for all images acquired by a given camera. One of the key goals of this paper is to derive a functional form for g that improves accuracy.

Color constancy. Though it can be formulated in many different ways, the basic goal of computational color constancy is to infer a representation of surface spectral reflectance that is invariant to changes in the spectral distribution of a scene's illumination. One approach is to define a "canonical" linear representation of scene color

$$\kappa_o(\ell_o, v) = \pi_o \cdot e(\ell_o, v), \tag{3}$$

i.e., the color corresponding to a canonical set of spectral sensors π_o and canonical illuminant ℓ_o (often the equal-energy illuminant E). The goal, then, is to infer κ_o from a camera's nonlinear output $y(\ell, v)$ that has been captured under unknown illuminant ℓ.

A common approach is to first calibrate the camera radiometrically, so that $x(\ell, v) = g^{-1}(y(\ell, v)) = C \cdot \kappa(\ell, v)$ can be computed, and then assume that $x(\ell, v)$ is related to the desired canonical representation by a linear (or diagonal) transform: $x(\ell, v) = M\kappa_o$, $M \in GL(3)$ [1, 11, 23, 29]. The transform M depends on the illuminants (ℓ, ℓ_o) and sensors (π, π_o), and, for any realistic scenario, is a coarse approximation. (The map $\kappa_o \to x$ is usually not bijective, for example.)

The accuracy of the linear (and diagonal) model $x(\ell, v) = M\kappa_o$ has been well studied for the case of a single camera in a Lambertian world. In this scenario, $\pi = \pi_o$, and the conditions for the linear mapping to be accurate can be stated very precisely [3, 8, 30]. The problem becomes more complicated when multiple cameras are involved because the spectral filters of the camera cannot be easily related. One of the goals of this paper is to perform an empirical evaluation of the linear model for a broad collection of common cameras.

Image matching. Multi-view stereo; object and scene recognition; and content-based image retrieval are all applications that rely on matching colors between images. Generically, one is given two images of the same scene from different cameras under different lighting and viewing conditions, and one seeks to determine corresponding image points. This requires knowledge of the mapping from colors in one image to colors in the other, $y(\ell,v) \rightarrow y'(\ell',v')$, and according to our model, this mapping would be of the form $y' = g'(C'C^{-1} \cdot g^{-1}(y))$. So one approach to matching is to first estimate the parameters of the mapping.

An alternative approach to matching colors y and y' is to compute a so-called *color invariant* that is camera- and illumination-independent. This amounts to computing a function $h\colon \mathbb{R}^3 \rightarrow \mathbb{R}^k$ that satisfies $h(y) = h(y')$ for all pairs (y, y') that are measurements from the same surface patch. The most common example assumes that a camera's output color vectors are related to a canonical linear representation by a six-parameter model [7]

$$y(\ell,v) = (M_D \cdot \kappa_o)^{\gamma_D}, \tag{4}$$

where M_D is a diagonal 3×3 matrix and $(\cdot)^{\gamma_D}$ represents independent exponentiation in each color channel. In this case, an invariant can be derived using a per-channel logarithm followed by a normalization, and this has been used, for example, for illumination-invariant stereo matching with a single camera [17, 18, 19] and many different cameras [14]. (There are also color invariants that are designed for linear data (κ and κ') [9, 12, 13], but these are less relevant to this paper.) The model in Eq. 4 is a special case of what is proposed in Eq. 2, and in Sect. 4 we evaluate its accuracy relative to other possibilities.

4 Experimental results

To evaluate the models proposed in the previous section, we exploit the increasing availability of consumer-level cameras that output both RAW data (κ) and JPEG-encoded data (y). This allow us to separately examine the components (C, π, and $g(\cdot)$) of our models.

4.1 A database for camera analysis

Our database contains registered images of color checker patterns under controlled lighting, as well as registered RAW/JPEG pairs of general scenes. It currently includes over 1000 images taken with 35 different camera models, ranging from simple point-and-shoot cameras to professional DSLRs. These images will be made available on the Web for other researchers.

Color checker data. We use two color calibration targets, the 24-patch *ColorChecker*, and the 140-patch *Digital ColorChecker SG*, both manufactured by X-Rite. (In this paper we only report results for the latter.) We photograph each pattern under two fixed lighting conditions, using Tungsten (3200K) and Daylight (4800K) photo flood light bulbs. In each case, we take both JPEG and (if supported by the camera) RAW images with 4 different exposures (stops +1, 0, -1, and -2). We use a fixed "Tungsten" white balance setting for all cameras, and, for a subset of the cameras, we use "auto" white balance as well. Our database includes cameras by most major manufacturers (Canon, Casio, Fuji, Kodak, Leica, Nikon, Olympus, Panasonic, Pentax, and Sony), and currently contains 11 JPEG-only cameras and 24 cameras with both RAW and JPEG support. We use the program `dcraw` [4] to render RAW images as PNGs in standard linear RGB colorspace, using the camera's white balance multipliers.

In each source image we compute the homography that maps the pattern to a canonical position, and resample to obtain cropped and aligned patterns. We generate both point-sampled and "smoothed" (4x linearly down-sampled) versions of these images. The former represent true samples of the original intensities, sensor noise, and JPEG compression artifacts, while the latter (used in the experiments below) attenuate such effects. To remove remaining misregistrations, which are mainly due to lens distortion, we construct our final registered images by conservatively cropping the individual color squares of the checker pattern and compositing them into a single image (see Figure 1 for examples).

This data is sufficient for evaluating the portion of the model (C and $g(\cdot)$) that relates RAW data to JPEG data. But in order to evaluate the other portion of the model (π), we must compute image irradiance by correcting for optical effects (vignetting) and spatial variations in the incident illumination. We do this by fitting 2D spatial gain functions over the registered patterns, composed of a linear function per illuminant and a quadratic radial function per camera. These gain functions are estimated using the gray patches around the perimeter of the color checker, and they correct the spatial variations of the aligned RAW images to a residual non-uniformity of less than 1.5%. These "spatially-corrected" images can then be directly compared to the known relative radiance values of the color checker squares.

General scenes. A subset of the RAW-capable cameras in our database allow simultaneous capture of RAW/JPEG pairs, and with these devices we can capture registered pairs in natural environments. Our database includes a total of 85 such pairs taken of general indoor and outdoor scenes with 12 different camera models. We use these images in Sect. 4.3.

4.2 Camera sensor characteristics

We first evaluate the validity of Eq. 1 by exploring the relationship between spectral image irradiance and cameras' sensor measurements. We are interested in assessing the degree to which output RAW values are linearly related to image irradiance, as well as understanding the nature of each camera's spectral filters π. For these experiments, we use a canonical linear representation κ_0 of the color checker (provided by the manufacturer) consisting of CIE XYZ values under illuminant D65, and we compare these to the spatially-corrected RAW images described above.

In the first experiment, we find that in the overwhelming majority of cases, a camera's RAW output is indeed linearly related to image irradiance. A representative example is shown in Fig. 2(a), where we see that once the illumination variation and vignetting effects are removed, the RAW values form a near-exact linear relationship with the known relative scene radiance of the color checker. If we measure deviation from linearity using RMS residual error in the total-least-squares linear fit, we find that the average residual over the 24 RAW-capable cameras in our database is 1.9 gray levels, while the average RMS noise level (estimated from the variance within the squares) is 1.05.

In the second experiment, we explore (somewhat indirectly) the spectral composition of each camera's sensors. One expects a camera's spectral sensors (π) to be approximate linear combinations of the color matching functions of the CIE standard observer. To assess the degree to which this is true, we evaluate the ability of a general linear transform to map the standard κ_0 values to (spatially-corrected) camera RAW values. Note that this test is approximate because the RAW images are observed under different illuminants (3200K and 4800K) than the standard values (D65). Due to this and the manufacturing limitations on π, we do not expect the linear mapping to provide an exact fit. Nonetheless, as shown by the representative examples in Fig. 2(b,c), the linear transform does a reasonable job for most of the cameras and illuminants in our database. The average RMS residual error in this case is 3.17 gray levels over all RAW images—approximately three times the noise level.

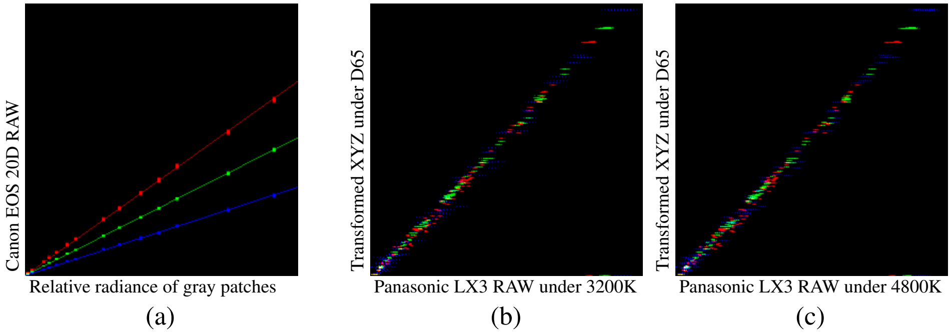


Figure 2: *(a)* Camera RAW vs. image irradiance. The plot shows a typical, almost perfectly linear relationship between a camera's RAW output (Canon EOS 20D) and relative scene radiance, as given by the 14 unique gray squares of a color checker. Typical joint intensity histograms for Camera RAW and CIE XYZ, showing that a general 3x3 linear transform can provide a reasonable fit. Shown are the histograms comparing the RAW intensities of a Panasonic LX3 camera under two different illuminants *(b)* 3200K, and *(c)* 4800K, with the best C-transformed color checker CIE XYZ values under Illuminant D65.

4.3 Nonlinear processing

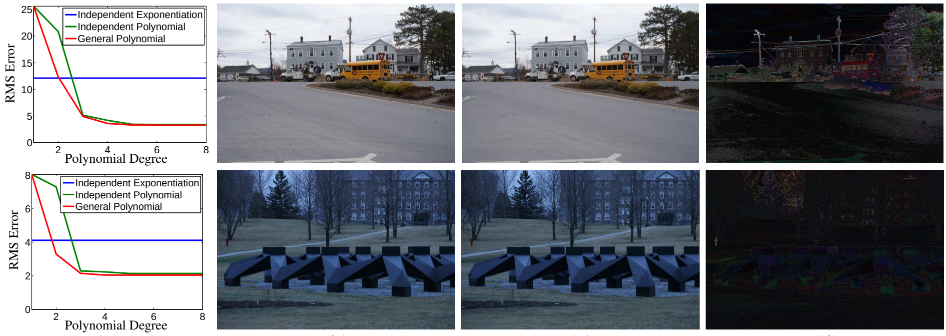
Next, we evaluate three different models, composed of $C \in GL(3)$ and $g\colon \mathbb{R}^3 \to \mathbb{R}^3$, for the cameras' color rendering pipelines. These models increase in complexity.

1. *Independent exponentiation*. Recent work has considered the model of Eq. 4 for cases in which the per-channel exponents are equal and known [14], equal and unknown [17, 18], and arbitrary [7]. We generalize these by replacing the diagonal transform by a general linear transform and allowing the exponents to be arbitrary. The resulting model is $y = (C \cdot \kappa)^{\gamma_D}$, and it has 12 parameters (9 for the entries of C, and 3 for γ_D).

2. *Independent polynomial*. A more general model is obtained by replacing the per-channel exponential by an nth-degree polynomial. This is partially motivated by the success of polynomial model for traditional "per-channel" radiometric calibration [21, 26]. We write our model as $y_i = g_i([C \cdot \kappa]_i)$ where y_i is the value of the ith color channel, and $g_i(x) = \sum_{p=0}^{n} \beta_{i,p} x^p$ is constrained to be monotonic in the typical range of x. Note that the scale of each column of C can be absorbed into the corresponding polynomial, so the total number of parameters in this model is $3(n+3)$.

3. *General polynomial*. More general than restricting the nonlinearity to operate independently in each C-transformed color channel is to consider an nth-degree polynomial map from $\mathbb{R}^3$ to $\mathbb{R}^3$. This is written $y_i = \sum_{p_1+p_2+p_3 \leq n} \beta_{i,p_1 p_2 p_3}\ \kappa_1^{p_1} \kappa_2^{p_2} \kappa_3^{p_3}$, with parameters $\{\beta_{i,p_1 p_2 p_3}\}$ that capture both the effect of the linear transform C and the nonlinearity. The total number of parameters in this model is considerably larger at $\frac{1}{2}(n+1)(n+2)(n+3)$.

These models are evaluated by estimating parameters for the best least-squares fit between a number of RAW/JPEG pairs—each pair providing a set of (κ, y) pairs. We measure the quality of the fit by reporting the root-mean-squared-error (RMSE) over the training set. For the independent exponentiation model, simple regression is used to find the optimal C



(a) (b) (c) (d)

Figure 3: RAW→JPEG maps using different models for general scenes. *(a)* Plot of residual RMS error for different models, *(b)* JPEG from camera, *(c)* JPEG fitted from RAW images using independent polynomial model with $n = 5$, and (d) Absolute error value image (scaled up by 10 for visibility). Note that most errors occur in high-frequency regions where we also expect unmodeled errors due to sharpening and compression.

corresponding to every choice of γ_D, and the optimal γ_D is determined by exhaustive search in a large range of possible values. The parameters for the general polynomial model can be estimated with simple regression as well.

For the independent polynomial model an iterative approach is needed. Given an estimate of C, we compute the parameters of the $g_i(\cdot)$ functions using quadratic programming to minimize the least-squares error with monotonicity constraints. Then, C is updated with a step along the error gradient, which is computed assuming fixed $g_i(\cdot)$. We choose our initial estimate for C such that the $[C \cdot \kappa]_i$-values corresponding to the same y_i in the training set $\{\kappa_k, y_k\}_{k=1}^{K}$ are close to being equal: We partition the domain of y_i into a finite set of values $\mathbb{V}$. For each $v \in \mathbb{V}$, a weight vector $w^{vi} \in \mathbb{R}^K$ measures the "membership" of every $y_{k,i}$ to the partition corresponding to v (we use $w_k^{vi} = \exp(-\lambda(y_{k,i} - v)^2)$). The ith row c_i^T of C is computed to minimize the weighted variance $S_i = c_i^T A_i c_i$ with $A_i \in \mathbb{R}^{3\times 3}$ defined as

$$A_i = \sum_{v\in\mathbb{V}} \sum_{k=1}^{K} w_k^{vi} \left(x_k^{vi} x_k^{vi\,T} \right), \qquad \text{with } x_k^{vi} = \kappa_k - \frac{\sum_k w_k^{vi} \kappa_k}{\sum_k w_k^{vi}}. \tag{5}$$

The nontrivial solution for c_i is the smallest eigenvector of A_i.

Figure 3(a) shows the typical performance of the models when applied to RAW/JPEG pairs of natural scenes. The independent exponentiation model is the simplest and performs worse than polynomial models with degree greater than two. The general polynomial model provides only a marginal benefit over independent polynomial model, even though it has a much larger number of parameters. Based on these results, we settle on the independent polynomial model with degree $n = 5$ as a good balance between accuracy and complexity, and we use this model for the remainder of the paper. Figures 3(b-d) compare the true JPEGs and corresponding mapped RAW images using this 24-parameter model.

Having settled on the independent polynomial model with $n = 5$, we next explore this model more systematically using the color checker images. Since the color checker provides a very wide gamut (much larger than that of any one natural scene), these tests help best evaluate the model's ability to represent the camera's processing pipeline. Figure 4 summarizes the results of applying the model to the color checker images from all 24 cameras with

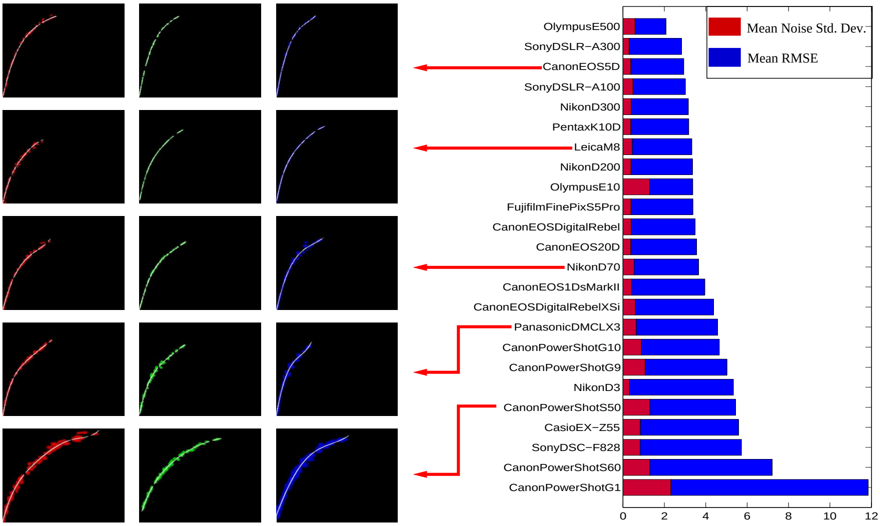


Figure 4: Results for the independent polynomial RAW→JPEG map estimation: On the right, bar graph showing mean RMSE values for each camera (with inset red bar showing mean noise standard deviation). On the left, joint histograms of y_i and $[C \cdot \kappa]_i$ for images from five different cameras, with the estimated g_i super-imposed in white.

RAW support, across different illuminants, and white balance and exposure settings. Joint histograms showing the nonlinear relationship between $[C \cdot \kappa]_i$ and y_i are included for five of the cameras, along with the estimated degree-5 polynomial $g_i(\cdot)$ that best approximates this map. It is clear from these histograms that the nonlinear maps are camera dependent, and for cameras, channel dependent. Also, in most cases these maps are well approximated by the independent polynomial model.

5 Analysis and discussion

Our findings suggest the following. First, when it is available, the RAW output of most cameras is proportional to image irradiance. We tested 24 different RAW-capable cameras and for all of them, the deviation from linearity is at the same scale as sensor noise. Second, the mapping from (demosaiced) RAW color three-vectors to colorimetric tristimulus values (CIE XYZ) can often be represented by a general linear 3×3 transform even when (limited) changes in the illuminant spectrum occur. For all of the RAW-capable three-sensor cameras in our database, we found that a 3×3 transform yields errors less than three times larger than the sensor noise. Third and finally, a twenty-four parameter model, consisting of a general linear 3×3 color transform and a per-channel 5th-degree polynomial, is able to represent the nonlinear color processing pipelines of a large number of consumer cameras. This representation provides a good balance between accuracy and model complexity.

The next step is to explore applications of such a model to visual tasks such as color constancy, radiometric calibration, and image matching. Here, the goal is to estimate the model parameters from natural input image data. We view the image matching problem as particularly interesting because it is likely that matching image patches $y(\ell, v) \leftrightarrow y'(\ell', v')$ can be achieved through a "shortcut" model $y = g(y')$ or through the use of invariants (perhaps a modification of [10]) that do not require estimating a full set of parameters for each camera. It may also be possible to isolate local image effects, such as specular highlights and shading changes, from the global image differences caused by camera-dependent color processing.

In order to fully exploit the Internet as a data source for computer vision, we must use the color information that is available in its images. Doing so requires compensating for the scene-dependent nonlinear color processing performed by consumer cameras, and deriving models like those proposed here is an important first step in this direction.

References

[1] D. Brainard and W. Freeman. Bayesian color constancy. *J. Optical Society America A*, 14(7):1393–1411, 1993.

[2] G. Buchsbaum. A spatial processor model for object colour perception. *J. Franklin Institute*, 310(1):1–26, 1980.

[3] H. Chong, S. Gortler, and T. Zickler. The von Kries hypothesis and a basis for color constancy. In *Proc. IEEE Intl. Conf. Computer Vision*, 2007.

[4] dcraw. Decoding raw digital photos in linux. http://www.cybercom.net/~dcoffin/dcraw/, Last accessed: April 27, 2009.

[5] P. Debevec and J. Malik. Recovering high dynamic range radiance maps from photographs. In *SIGGRAPH '97: Proc. Conf. Computer Graphics*, pages 369–378, 1997.

[6] H. Farid. Blind inverse gamma correction. *IEEE Trans. Image Processing*, 10(10): 1428–1433, 2001.

[7] G. Finlayson and R. Xu. Illuminant and gamma comprehensive normalisation in log RGB space. *Pattern Recognition Letters*, 24(11):1679–1690, 2003.

[8] G. Finlayson, M. Drew, and B. Funt. Color constancy: generalized diagonal transforms suffice. *J. Optical Society America A*, 11(11):3011–3019, 1994.

[9] G. Finlayson, S. Hordley, and M. Drew. Removing shadows from images. In *Proc. European Conf. Computer Vision*, pages 823–836, 2002.

[10] G. Finlayson, S. Hordley, G. Schaefer, and G. Yun Tian. Illuminant and device invariant colour using histogram equalisation. *Pattern Recognition*, 38(2):179–190, 2005.

[11] D. Forsyth. A novel algorithm for color constancy. *Intl. J. Computer Vision*, 5(1), 1990.

[12] J. Geusebroek, R. Van den Boomgaard, A. Smeulders, and H. Geerts. Color invariance. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 23(12):1338–1350, 2001.

[13] T. Gevers and A. Smeulders. Color-based object recognition. *Pattern recognition*, 32 (3):453–464, 1999.

[14] M. Goesele, N. Snavely, B. Curless, H. Hoppe, and S. Seitz. Multi-view stereo for community photo collections. In *Proc. IEEE Intl. Conf. Computer Vision*, 2007.

[15] M. Grossberg and S. Nayar. Determining the camera response from images: what is knowable? *IEEE Trans. Pattern Analysis and Machine Intelligence*, 25(11):1455–1467, 2003.

[16] M. Grossberg and S. Nayar. Modeling the space of camera response functions. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 26(10):1272–1282, 2004.

[17] Y. Heo, K. Lee, and S. Lee. Illumination and camera invariant stereo matching. In *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 2008.

[18] Y. Heo, K. Lee, and S. Lee. Mutual information-based stereo matching combined with SIFT descriptor in log-chromaticity color space. In *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 2009.

[19] H. Hirschmüller and D. Scharstein. Evaluation of stereo matching costs on images with radiometric differences. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 2009.

[20] J. Holm, I. Tastl, L. Hanlon, and P. Hubel. Color processing for digital photography. In P. Green and L. MacDonald, editors, *Colour Engineering: Achieving Device Independent Colour*, pages 179–220. Wiley, 2002.

[21] S. Kuthirummal, A. Agarwala, D. Goldman, and S. Nayar. Priors for large photo collections and what they reveal about cameras. In *Proc. European Conf. Computer Vision*, 2008.

[22] S. Lin, J. Gu, S. Yamazaki, and H.-Y. Shum. Radiometric calibration from a single image. In *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 2004.

[23] L. Maloney and B. Wandell. Color constancy: a method for recovering surface spectral reflectance. *J. Optical Society America A*, 3(1):29–33, 1986.

[24] C. Manders, C. Aimone, and S. Mann. Camera response function recovery from different illuminations of identical subject matter. In *Proc. Intl. Conf. Image Processing*, volume 5, 2004.

[25] S. Mann and R. Picard. Being ‘undigital’ with digital cameras: Extending dynamic range by combining differently exposed pictures. In *Proc. IS&T Annual Conf.*, pages 422–428, 1995.

[26] T. Mitsunaga and S. Nayar. Radiometric self calibration. In *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 1999.

[27] R. Ramanath, W. Snyder, Y. Yoo, and M. Drew. Color image processing pipeline. *IEEE Signal Processing Magazine*, 22(1):34–43, 2005.

[28] M. Stokes, M. Anderson, S. Chandrasekar, and R. Motta. A standard default color space for the internet: sRGB. Microsoft and Hewlett-Packard Joint Report, Version 1.10. Available at http://www.color.org/sRGB.xalter., 1996.

[29] J. van de Weijer, T. Gevers, and A. Gijsenij. Edge-Based Color Constancy. *IEEE Trans. Image Processing*, 16(9):2207–2214, 2007.

[30] G. West and M. Brill. Necessary and sufficient conditions for von Kries chromatic adaptation to give color constancy. *J. Math. Bio.*, 15(2):249–258, 1982.