https://doi.org/10.2352/ISSN.2169-2629.2017.32.306

Use of Simulated Reflectance Spectra in Camera Transform Creation

Michael J. Vrhel, Artifex Software, 7 Mt. Lassen Drive A-134, San Rafael, CA 94903

Abstract

Spectral simulation methods can be useful in the creation of mappings from recorded values to colorimetric values for cameras and scanners. In this work, a spectral data set is expanded using simulated spectra created from a real data set. The expanded data set is used to create a multi-dimensional look-up-table (MLUT). The performance of the MLUT is compared to the performance of an MLUT created without the simulated spectral data. Results suggest that the use of simulated spectra in creating an MLUT could help to reduce maximum errors.

Introduction

The design of a mapping from raw camera values to colorimetric values should consider a number of different parameters. These include the mapping's computational performance, its colorimetric performance, its ability to adjust to changing illumination conditions and the device noise to name a few. In achieving colorimetric performance, there tend to be two primary approaches, a black box and an analytical method.

In the black box approach, a characterization target such as a Macbeth chart is recorded by the camera providing camera measured values. The target samples are also measure with a colorimeter providing CIEXYZ (or equivalent) values. A mapping is then constructed between the two sets of data. This approach has an advantage in that it can be quickly implemented using a variety of generic fitting algorithms. It is however limited by the fact that the training data is fixed to a specific training chart.

The analytical method uses a physical model for the imaging system. This approach is limited by the need to obtain accurate estimates for the spectral sensitivities of the imaging system, its nonlinearities and its noise characteristics. It has an advantage over the black box approach though in that a huge variety of spectral samples can be pushed through the physical model providing the opportunity for more robust training as well as improved analysis of the performance and error properties.

In both the black box and analytical approaches, the mapping that is constructed can be any number of methods including but not limited to a linear matrix mapping [1, 2], a multi-dimensional look up table [3], a polynomial fit [4, 5], or an artificial neural network [6]. In addition, the cost function that is minimized can be any number of perceptual colorimetric measures such as CIELAB ΔE_{ab}, CIECAM02, Δ JMh etc.

While the analytical method has the advantage of being able to use a large set of spectral samples in its training, the training is limited by the need to have on hand a large set of measured spectra. There are a number of publicly and commercially available data sets of spectral reflectances including [7, 8, 9, 10]. The set from [10] is particularly large but is still much smaller than the set of spectra that exist in the real world. With mapping methods that have a few parameters such as a linear matrix, the need for a huge number of spectral samples is not going to be as great as it would be for a mapping method like a multi-dimensional look-up-table that has a large number of parameters.

Spectral simulation methods are techniques that can be used to expand a set of existing reflectance into a much larger set that has similar properties. In [11], a number of techniques for simulating spectra were introduced. These methods made use of the statistical properties of an existing data set. Ideally, the much larger generated data set can provide a wider variety of realistic spectral samples that might be encountered by the camera.

The goal of this work is to investigate the impact of using spectral simulation methods on creating camera mappings. In particular, we will focus on multi-dimensional look-up-tables due to the large number of parameters that exist in these mappings. The paper is organized to first pose the analytical camera mapping problem in a general form. We will then discuss spectral simulation methods and MLUT creation. Finally, we will conclude with training and performance testing using real and simulated reflectance spectra.

Analytical Camera Mapping

To model the imaging system of the camera as well as the human visual system, we will use a vector space approach outlined in [12]. With this approach, the noise free camera recording model can be expressed as

$$\mathbf{c}_i = \mathbf{N}^T \mathbf{L}_c \mathbf{r}_i \tag{1}$$

where $\mathbf{r}_i$ is the spectral reflectance vector at spatial location i, $\mathbf{L}_c$ is a diagonal matrix whose elements are the spectral power distribution of the capture illumination, the columns of the matrix $\mathbf{N}$ represent the spectral sensitivities of the camera and the vector $\mathbf{c}_i$ is the recorded color value for the camera at spatial location i. For most cameras, the dimension of the vector $\mathbf{c}_i$ would be three for the red, green and blue spectral sensitivities.

In a similar form, the human visual system can be expressed as

$$\mathbf{v}_i = \mathscr{F}(\mathbf{A}^T \mathbf{L}_v \mathbf{r}_i, P) = \mathscr{F}(\mathbf{t}_i, P) \tag{2}$$

where $\mathbf{L}_v$ is a diagonal matrix representing the spectral power distribution of the viewing illumination, the columns of matrix $\mathbf{A}$ contain the CIEXYZ color matching functions, $\mathbf{t}_i$ is the CIEXYZ value of reflectance $\mathbf{r}_i$ under $\mathbf{L}_v$, $\mathscr{F}$ is a mapping from CIEXYZ to some perceptual color space and P is the conditions/adaptation parameters required to perform the mapping $\mathscr{F}$.

Given the above two equations, we can form an optimization problem to find a mapping $\mathscr{G}$ that minimizes the perceptual color error in the color space defined by the mapping $\mathscr{F}$. This problem can be expressed as

$$\min_{\mathscr{G}} = ||\mathscr{F}(\mathbf{A}^T\mathbf{L}_v\mathbf{r}_i, P) - \mathscr{F}(\mathscr{G}(\mathbf{N}^T\mathbf{L}_c\mathbf{r}_i), P)|| \tag{3}$$

over some set $\mathbf{r}_i$. Note that as discussed above, the result that we get for $\mathscr{G}$ is going to be highly dependent upon the set $\mathbf{r}_i$. Our goal here is to see if creating a larger set can help provide improved results when the mapping $\mathscr{G}$ is a multi-dimensional look-up-table.

Spectral Simulation

Much of the modeling of reflectance spectra has been focused upon the mean and the covariance structure of the data. For example, many of the principal component methods rely upon a small set of basis vectors to describe the set of spectra. If the spectral data was a multi-variate Gaussian data set, the mean and covariance would completely describe the properties of the spectra. Since reflectance spectra are bounded between zero and one (ignoring fluorescence) the data is not Gaussian distributed. Histogram analysis of the data in [7] and [10] reveals that the data is far from Gaussian and the partial distributions vary significantly across the wavelengths.

In [11], several methods for generating simulated reflectance spectra were investigated. One of the more promising methods was one that made use of set theoretic techniques. Set theoretic methods have been used on a number of color problems [12, 13]. For spectral generation it can be used to create a set of vectors such that they are properly bounded, have a specific mean and have a variation distribution similar to another data set. In this way, one can generate reflectance spectra that are similar in properties and provide a useful means to expand our set of reflectance spectra that we can use to train and or test our system.

Another method from [11] that can be used to generate spectral reflectances is through the use of an artificial neural network (ANN). In this case, an ANN introduces the correlations between the spectal wavelengths as well as the bounding constraints. For example, one can create an ANN that produces simulated spectra when given white noise as input. Once the network is trained, this ends up being a way to rapidly create spectra compared to the set theoretic method, which relies upon an iterative technique.

Given a white noise input vector $\mathbf{e}$ of size Mx1, the output of the 2-layer feed-forward ANN with S neurons in layer one is expressed as

$$\mathbf{r} = \mathscr{L}(\mathbf{e}) = \Psi[\mathbf{V}\Phi(\mathbf{W}\mathbf{e}+\mathbf{b})+\mathbf{d}] \tag{4}$$

where $\Phi(\mathbf{x}) = [\phi(x_1), ..., \phi(x_S)]^T$, $\mathbf{b}$ is an S element vector, $\mathbf{W}$ is an SxM matrix, $\mathbf{V}$ is an MxS element vector, $\mathbf{d}$ is a M element vector, ϕ is the sigmoidal function, which is given by

$$\phi(x) = \frac{2}{1+\exp(-2*x)} - 1 \tag{5}$$

and the output function is given by $\Psi(\mathbf{x}) = [\psi(x_1), ..., \psi(x_{171})]^T$

$$\psi(x) = \frac{1}{1+e^{-x}}, \tag{6}$$

which insures the output values will be bounded between zero and one. Note that we are representing our reflectance spectra with 171 samples in the visible spectrum. Based upon the principal component analysis in [7], the variability of most reflectance data is completely contained within an eight dimensional space. This points to selecting S to be approximately this value.

The distribution function used for the white noise input can be determined by performing a whitening transformation on the real reflectance spectra data and computing a histogram. The whitening transformation is given by

$$\mathbf{w} = \mathbf{D}^T[\mathbf{r} - \bar{\mathbf{r}}] \tag{7}$$

where

$$\mathbf{K}_r = \mathbf{D}\Lambda\mathbf{D}^T \tag{8}$$

is the covariance matrix of the reflectance spectra and $\bar{\mathbf{r}}$ is the mean of the reflectance spectra.

Applying the above whitening transformation to a set of reflectance spectra, we can then use the whitened data and the reflectance spectra to train our network. In this work, the network was optimized using the Levenberg-Marquardt algorithm.

Since we wish to maintain the statistical distribution of the original spectral data (and use data similar to which the network was trained), whitened data with the same probability distribution of the training whitened data was provided as input to the network. The whitened data for creating new spectra was generated by creating uniform distributed uncorrelated data for each wavelength and transforming each wavelength to the desired probability density function (PDF) through the use of

$$[\mathbf{w}]_i = F_i^{-1}([\mathbf{u}]_i) \tag{9}$$

where F_i is the cumulative distribution function (CDF) for element i of the whitened spectral data. The CDF was estimated using a histogram of the whitened training data.

MLUT Creation

The methodology for creating the MLUT is described in [14]. To review, that method uses a locally linear approximation of nearby training spectra to interpolate and extrapolate grid point values in the MLUT. Figure 1 displays a 2-D example (red-blue plane), where the training data points are shown in blue (small circles) and we wish to interpolate the grid point shown in red (large circle). To do this, we use training data that surrounds our MLUT grid location and compute a matrix $\mathbf{M}$

$$\min_{\mathbf{M}} = ||\mathscr{F}(\mathbf{A}^T\mathbf{L}_v\mathbf{r}_i, P) - \mathscr{F}(\mathbf{M}(\mathbf{N}^T\mathbf{L}_c\mathbf{r}_i), P)|| \tag{10}$$

where i indexes across the points that surround our grid point. With matrix $\mathbf{M}$ calculated, we then use the matrix to compute the value to place at the grid point.

Similarly for the extrapolation of grid points outside the gamut of our training data, we use data values near the grid point to compute a matrix and use it to compute the grid point value. A 2-D example is shown in Figure 2 for illustration. In this work, the error metric used in optimizing the table was CIECAM02 ΔJMh.

Use of Simulated Data

For our testing, the camera spectral sensitivities, $\mathbf{N}$, are as shown in Fig. 3. The viewing illumination $\mathbf{L}_v$ and capture illumination $\mathbf{L}_c$ were both D50. To understand the usefulness of simulated data on the creation of an MLUT, the data in [7] and [10]

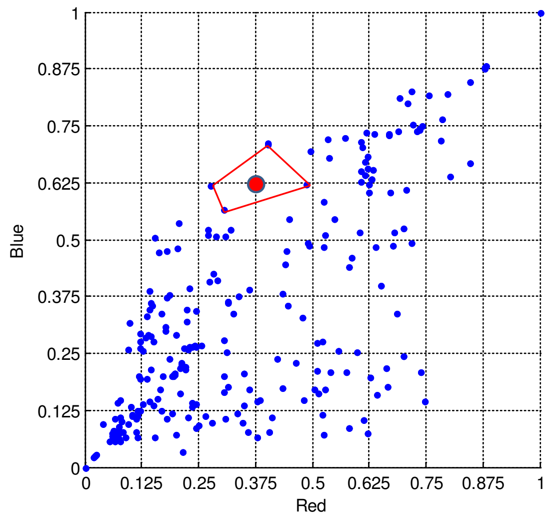

Figure 1. *Example of interpolation of point in MLUT*

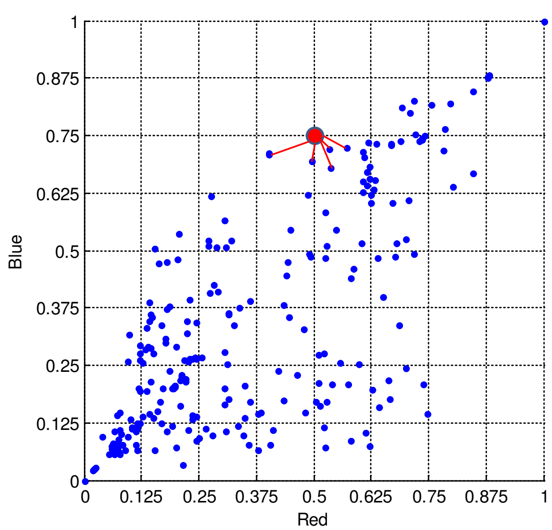

Figure 2. *Example of extrapolation of point in MLUT*

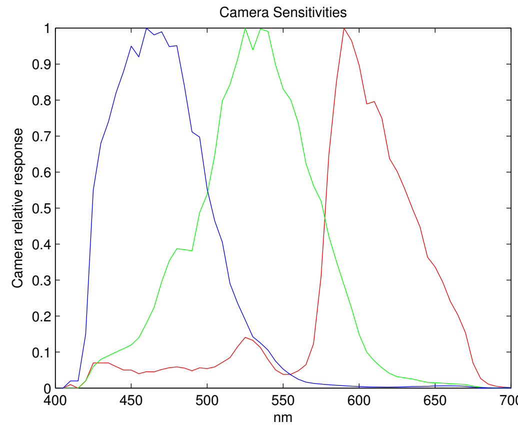

Figure 3. *Spectral sensitivities used for simulation*

were combined into a single data set of 2616 spectral samples. The combined data set was randomly divided into two equally sized sets. One set was for training and the other set was for testing. The training data set was used to create a 9x9x9 MLUT that used no simulated data. We will refer to this MLUT as the standard (std) MLUT. The testing portion of the data was then used to test the performance of this MLUT. In addition, the training data set was expanded using an ANN created with the method described in Section . With this ANN, a data set was created that had 11308 samples (10000 simlulated spectra were added to the training set). This expanded data set was then used to create another 9x9x9 MLUT that we will refer to as simulated (sim) MLUT. The previously used testing data was then used to test the performance of this MLUT. This process was repeated a total of ten times to provide a measure of the variability due to the random splitting of the data and the ANN training. The results are shown in Table 1.

At the bottom of the table, the average is shown over the ten iterations and the standard deviation is computed. The results show a small increase in the average Δ JMh error but a large (10 percent) reduction in the maximum error. This indicates that it may be possible to make use of the simulated spectra in creating the MLUT for the purpose of reducing the maximum error. Note however, that the variation in the results is high with a 2.39 estimated standard deviation on the maximum error when using the simulated spectra.

Discussion and Summary

The use of simulated spectral data in creating mappings for color managing digital camera values was investigated. The results indicate that simulated spectral data could be helpful in reducing the maximum errors when creating an MLUT for the mapping. The use of simulated spectra however did result in a slight increase in the MLUT average error over the testing data. In addition, a fair amount of variation occurred across the results. The next step in this work is to investigate how to select those spectra that result in a reduction in the maximum error without increasing the average error. Typically the MLUT maximum error occurs in the areas of the table that are being extrapolated. Including only simulated spectra that are used in extrapolation would be the logical next step in this work.

Performance results for MLUTs

Iteration	MLUT	Avg Δ JMh	Max Δ JMh
1	std	1.69	9.84
	sim	1.82	7.99
2	std	1.66	7.57
	sim	2.13	7.66
3	std	1.59	11.08
	sim	2.07	9.78
4	std	1.72	10.93
	sim	1.97	7.97
5	std	1.42	10.21
	sim	2.12	14.93
6	std	1.17	10.08
	sim	1.93	8.13
7	std	1.65	10.06
	sim	1.72	7.07
8	std	1.59	9.96
	sim	2.20	11.32
9	std	1.61	10.57
	sim	2.08	8.30
10	std	1.73	10.27
	sim	1.90	7.71
Avg	std	1.64	10.06
	sim	1.99	9.09
σ	std	0.09	0.97
	sim	1.99	2.39

References

[1] G. Finlayson and M. Drew, White-Point Preserving Color Correction, Proc. IS&T Fifth Color Imaging Conference, pp. 258-261, (1997).

[2] P. Hubel, J. Holm G. Finlayson and M. Drew, Matrix Calculations for Digital Photography, Proc. IS&T Fifth Color Imaging Conference, pp. 105-111, (1997).

[3] P. C. Hung, Colorimetric calibration in electronic imaging devices using a look-up table model and interpolations, J. Electronic Imaging, vol. 2, pp. 53-61, (1993).

[4] G. Hong, M. R. Luo and P. A. Rhodes, A Study of Digital Camera Characterisation Based on Polynomial Modeling' Color Research and Application, vol. 26, no. 1, pp. 76-84, (2001)

[5] G. Finlayson, M. Mackiewicz and A. Hurlbert, Root Polynomial Colour Correction, Proc. IS&T 19th Color Imaging Conference, pp. 115-119, (2011).

[6] H. R Kang and P. G. Anderson, Neural Network Applications to the Color Scanner and Printer Calibrations, J. Electronic Imaging, vol. 1, pp. 125-134, (1992).

[7] M. J. Vrhel, R. Gershon and L. Iwan, The Measurement and Analysis of Object Reflectance Spectra, Color Research and Application, vol. 19, no. 1, pp. 4-9, (1994).

[8] R. Lenz, M. Osterberg, J. Hiltunen, T. Jaaskelainen and J. Parkkinen, Unsupervised Filtering of Color Spectra, Journal of the Optical Society of America, vol. 13, no. 7, pp. 1315-1324, (1996).

[9] K. Barnard, L. Martin, L., B. V. Funt and A. Coath, A Data Set for Color Research, Color Research and Application, vol. 27, no.3, pp. 148-152, (2002).

[10] D. Wueller, In Situ Measured Spectral Radiation of Natural Objects, Seventeenth Color Imaging Conference: Color Science and Engineering Systems, Technologies, and Applications Albuquerque, New Mexico, pp. 159-163, (2009).

[11] M. J. Vrhel and H. J. Trussell, Reflectance Spectra Modeling Methods, 15th IEEE International Conference on Image Processing (ICIP), pp. 489-492, (2008).

[12] H. J. Trussell, Application of Set Theoretic Models to Color Systems, Color Research & Application, vol. 16, pp 31-41, (1991).

[13] G. Sharma, Set Theoretic Estimation for Problems in Subtractive Color, Color Research & Application, vol. 25, no. 5, pp. 333-348, (2000).

[14] H. J. Trussell and M. J. Vrhel, Color Estimation Under Poisson Noise, IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 1914-1918, (2013).

Author Biography

Michael Vrhel was awarded his PhD from North Carolina State University in 1993; during his PhD, he was an Eastman Kodak Fellow. He has many years' experience working in digital imaging, including biomedical imaging and signal processing at NIH; color instrument and color software design at Color Savvy Systems Ltd, and positions at Conexant Systems and TAK Imaging. A senior member of the IEEE, he has a number of current and pending patents and is the author of numerous papers in the areas of image and signal processing. His current interest is in efficient computational color rendering methods, which is the focus of his work at Artifex Software.