Color Space Transformation from RGB to CIELAB Using Neural Networks

Nawar Fdhal[1,*], Matthew Kyan[1], Dimitri Androutsos[1], and Abhay Sharma[2]

[1] Department of Electrical and Computer Engineering
[2] School of Graphic Communications Management
Ryerson University, 350 Victoria Street
Toronto, Canada
M5B 2K3
nfdhal@ryerson.ca

Abstract. Transformations in digital color imaging from RGB to CIELAB are compared between conventional ICC profiles and a newly developed neural network model. The accuracy of the transformations are computed in terms of Delta E and a comparison is made between the ICC profile and a neural network implemented in MATLAB. The transformations are used to characterize and test the color response of an Epson 4800 inkjet printer. A number of data preprocessing techniques are described. The results demonstrate a back propagation Levenberg-Marquardt neural network algorithm with accuracy of 0.28 Delta E for non-training set data.

Keywords: Neural networks, color transformation, ICC color management, color lookup tables.

1 Introduction

Digital color imaging systems are used to capture, display and print image data. Due to the inherent differences in each device, pixel data in an image is manipulated in a specific manner that is based on the characteristics of each device, so that, for example, pixel values being sent to a display are adjusted for the condition of the monitor, while data for printing is adjusted to take account of the color characteristics of the printing system. Most practical color management systems adjust image values by transforming device-dependent pixel values (RGB and CMYK) into, and out of, a central, device-independent CIELAB color space [1]. A device's color response is measured and modeled, this data is then used to determine a transform relationship that is incorporated into the conversion of pixel data from RGB to CIELAB and/or from CIELAB to CMYK.

A well established color management implementation exists in the form of International Color Consortium (ICC) color profiles. In ICC-based systems, device characterization information is stored in single and multi-dimensional lookup tables (LUT) within profile files, such that an input profile provides a mapping between

* Corresponding author.

P. Muneesawang et al. (Eds.): PCM 2009, LNCS 5879, pp. 1011–1017, 2009.

input RGB and CIELAB, and an output profile provides a mapping between CIELAB and output RGB/CMYK values [2]. The accuracy of color from input to the displayed image, to the printed image, depends in part on the quality of the characterization.

There are a number of ways to define the characterization and transform relationship between device-dependent values and CIELAB. Typically a test chart of equi-spaced RGB/CMYK values is printed or displayed, and the resultant CIELAB values are measured using a colorimeter or spectrophotometer, this physical data is then used to model the response of the system. It is possible to use data fitting processes that range from a simple linear matrix approximation to higher order polynomial regression [3] or Newton-Raphson iterative techniques [4]. Researchers have described parametric dye modeling that uses Neugebauer-type mathematical models to predict the color produced when different amounts of dye colorant are present [5]. It is possible to use an empirical approach to construct a lookup table [6] and the literature also describes attempts to use both lookup tables and curves to expand data in certain areas [7]. In general, devices such as additive color, computer monitors can be characterized by a linear expression (the phosphor matrix), combined with a nonlinear expression (the gamma curve), while more complex subtractive color systems, such as print devices require fitting techniques that can adequately model a non-linear color response.

Artificial Neural Networks (ANNs) can be used to approximate the non-linear relationship between device-dependent and device independent data sets [8-10]. A recent paper uses a Generalized Regression Neural Network (GRNN) in order to model the transformation between CMYK and CIELAB [11]. In another ANN approach, Resilient Back Propagation (RBP) was used to train a system using 56 patches [12]. Generally, RBP is not recommended for function approximation problems. Despite this, the approach showed some potential for forming accurate transformations. This research relates to the use of an ANN to identify faster and more robust color conversion methods.

The method proposed in this paper is based on a Levenberg-Marquardt (LM) back propagation training algorithm which has a high computer memory requirement, but accelerated convergence. This research compares the accuracy of the Levenberg-Marquardt (LM) back propagation artificial intelligence model implemented via MATLAB, with a standard ICC output profile made with X-Rite ProfileMaker 5.0.8 – a commercial ICC profiling software application. The accuracy of the transform relationship between RGB and CIELAB is studied for an Epson 4800 inkjet printer, addressed in RGB mode using a ColorBurst RIP. This research describes use of the LM algorithm and proposes a process to filter and pre-sort the data, which is important to remove non-unique solutions, speed up the training process and increase the accuracy of the algorithm.

2 Experimental Procedure

The project seeks to establish a transform between RGB instructions and measured CIELAB response for an Epson inkjet printer. The Epson 4800 is a 7-color printer (CMYK + light cyan + light magenta + light black), but was treated as an RGB device to avoid the complexity of the redundancy introduced by the black (K) channel.

Training data was obtained by printing a RGB test target called ‘TC9.18’. This target was chosen because the CIELAB measurement data could be used to both create an ICC profile and the same data could be used to train the MATLAB implemented neural network algorithm. A fair comparison between the two processes could then be undertaken, as the same training data was presented to each system. The test target was printed, allowed to dry, and then measured using an X-Rite i1i0 spectrophotometer.

2.1 Generating an ICC Profile

A conventional ICC output profile was made using X-Rite ProfileMaker 5.0.8. In the user GUI, the operator requested an output profile of ‘large’ size, this increases the nodes in the lookup table from 25 to 33. The larger number of cube nodes increases the file size of the profile but provides better accuracy during interpolation.

Training (measurement) data was presented to ProfileMaker that uses a proprietary internal fitting procedure to model the device response and then populate a color lookup table tag and save an ICC profile on disk. In creating the training data it is necessary to consider the number of patches used. For a non-linear device (e.g. a printer), a large number of patch samples allows the algorithms to create a better transform relationship and thus color conversions with better quality. The trade-off is between quality and chart size, as the larger number of patches necessitate a larger target which takes up more space on the printer and will take longer to measure. The difference in the characterization due to a smaller target was studied by making an ICC output profile from a 360 patch target and comparing that to a 936 patch target made on the same device, Fig 1. The 3-dimensional L*a*b* gamut visualization shows limitations where sparse data results in a non-smooth response. In the subsequent experiment, the larger target with 936 patches was used.

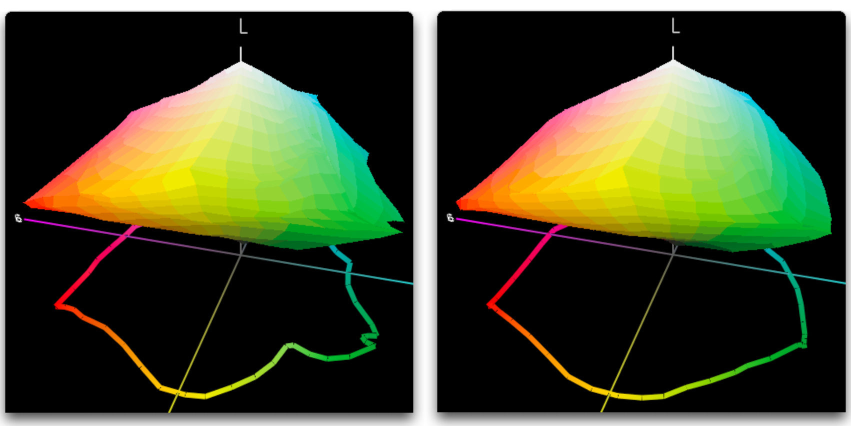

Fig. 1. When viewed in 3-dimensional L*a*b* color space, a small color target creates a jagged color gamut (left), while more data points creates a smoother characterization (right)

2.2 Data Preprocessing

The TC9.18 test target contains 936 patches, but these are not unique R, G, B combinations. In test targets it is normal to include a duplication of certain important colors, such as the white point RGB of (255, 255, 255) and/or a neutral gray (128, 128, 128). In some targets there are extra pixel values near skin tones in order that the training data contain adequate information in these colors as they are important for making visually pleasing images.

The TC9.18 test chart has a number of patches with the same RGB value such as (255, 255, 255), in practice their measured L*a*b* value will be slightly different for each patch, due to measuring instrument variability and printer deviation. This will affect the training process as there are different L*a*b* values for the same RGB value. In this experiment, a filtering process was developed that checks for the same RGB patches in the data set and then averages the corresponding L*a*b* values so that each RGB pixel value has a unique L*a*b* solution.

There is another aspect that also requires preprocessing of the data. When a target is printed, there may be pixel values that are outside the color gamut of the printer, these colors will be mapped to the outer extent of the printer gamut. This creates a situation where differing RGB pixel values create essentially the same measured L*a*b* value. This effect will happen at the very light and very dark ends of the color scales. Once again this leads to non-unique solutions in the training data set. A second filter was therefore developed that sought to reduce the effect that gamut clipping has on the data set. The use of the two filters creates unique training data and improves the accuracy of the neural network model. The effect of the two filtering processes was to reduce the 936 patches to 909 'filtered' data points. Another aspect of preprocessing to improve the quality of the raw data is manipulation of the RGB target values in order to create equi-spacing of the measured data points in L*a*b* color space. While this is considered beneficial this type of filtering was beyond the scope of the project at this time.

In the last step before using the data for training, it was necessary to map or scale the data to a suitable range. The activation function chosen for each hidden node was 'tansig'. As such, within the range of [+3,-3], the hidden nodes exhibit sufficient sensitivity and nonlinearity to allow the network to efficiently model the non-linear nature of the overall transfer function. Beyond such a range, hidden nodes would easily become saturated, leading to less stable, prolonged convergence. In general, raw RGB pixel values are in the range of [0, 255], L* ranges from 0 to 100, and a* and b* have a scale from -128 to +127. Each channel of the input and the output was normalized to have a mean of zero and standard deviation of 1.0 [13]. The normalized data was then necessarily mapped to a range of [+2.5,-2.5] to comply with the hidden node configuration.

2.3 Neural Network LM Back Propagation Model

A robust back propagation, Levenberg-Marquardt (LM) neural network solution [13] was considered to model the response of the inkjet printer. The network model has 3 inputs (RGB) and 3 outputs (L*a*b*). The total number of layers was chosen to be 3. There were two layers for input and output. The number of nodes in the hidden layer was chosen to be 360 as it gave a good balance between speed and accuracy.

The total number of weights in the network had to be limited because the Levenberg-Marquardt back propagation training algorithm only works with several hundreds of weights. The transfer functions for all the hidden nodes were chosen as ‘tansig’ and the ‘purlin’ for the output layer. The function was called ‘trainlm’, and it works best when training networks with several hundreds of weights and is recommended for function approximation problems especially when accuracy of the training is an issue.

3 Results

An ICC profile was compared to the back propagation algorithm. To automate the processing of data, the lookup table structures of the ICC profile were imported into MATLAB using ‘iccread’. The MATLAB command ‘makecform’ was used to create a concatenated lookup table structure akin to the function of a color management module (CMM) [14].

In order to confirm that MATLAB was processing the ICC profile contents correctly, the result was compared with the use of the same profile in Adobe Photoshop. The use of the ICC profile structure within MATLAB was compared to a ‘standalone’ mode where the profile was used in Adobe Photoshop CS3 with the Adobe CMM. Table 1 shows a comparison between MATLAB R2007b and Adobe Photoshop CS3 when converting specific RGB to L*a*b* values. In Table 1, $\Delta E^{*}{}_{ab} = \sqrt{(L_1 - L_2)^2 + (a_1 - a_2)^2 + (b_1 - b_2)^2}$, where L*, a* and b* are CIELAB coordinates. Absolute colorimetric rendering intent was used in both cases. There is good correlation between an ICC profile being used within MATLAB or in Adobe Photoshop CS3.

Table 1. Experiment to confirm that an ICC profile used within MATLAB or Photoshop produces equivalent results

RGB pixel value	L*a*b value when the ICC profile is used in MATLAB R2007b	L*a*b value when the ICC profile is used in Photoshop CS3	ΔE^*_{ab}
0, 0, 0	5.88, -1, 0	6, -1, 0	0.12
128, 128, 128	65.49, 0, 1	66, 0, 1	0.51
255, 255, 255	95.29, 0, 1	95, 0, 1	0.29
0, 0, 255	42.35, 3, -54	42, 3, -54	0.35

We now consider the results of preprocessing the data using the two preprocessing filters and then training the network.

The TC9.18 test target with 936 patches was printed and measured. The data was used to make an ICC profile using the process described earlier. Next the measurement data was filtered and then used to train the back propagation LM neural network. The effect of the two filtering processes reduced the 936 patches to 909 patches. The 909 patches form the *training* data set. In general a more useful test of accuracy is when non-training set data is used to estimate the accuracy of the transform. A different, proprietary target, called TC2.83 with 360 patches was printed and measured, this became the non-training or *testing* data set. Before using the testing set, patch values were compared to the training set, to ensure that the testing set is indeed unique, any identical patch values were removed, this process reduced the testing set from 360 to 256 unique patch values.

The 256 RGB pixel values were processed via the ICC profile and also via the neural network and in each case the predicted L*a*b* value was noted. The efficiency of each process was estimated by comparing the predicted L*a*b* values with the values measured from the testing color target. The mean and maximum ΔE^*_{ab} error for the ICC profile was 0.46 (3.64) and for the neural network the mean and maximum ΔE^*_{ab} error was 0.28 (2.29), when computed for 256 color patches. Some individual results are also shown in Table 2.

Table 2. Some example data points were processed via an ICC profile and also by the neural network algorithm

RGB Value	(1) Measured target L*a*b	(2) L*a*b (via ICC profile in MATLAB)	ΔE Column (1)-(2)	(3) L*a*b (via neural network algorithm in MATLAB)	ΔE Column (1)-(3)
0, 0, 51	11.46,-0.8,-21.47	11.76,0,-21	0.73	11.45,-0.68,-21.39	0.13
0, 255, 127	79.35,-30.9,19.26	79.6,-31,20	0.78	79.42,30.99,19.50	0.27
15, 15, 0	8.22,-1.54,1.39	9.01,-2,2	1.1	8.7,-2.08,1.39	0.73
112, 94, 94	53.41, 10.44,5.38	53.72, 10,5	0.66	53.57,10.51,5.73	0.4

We compare our results to other researchers who also used a back propagation algorithm to train their network for a RGB printer [12]. In that research, the average ΔE^*_{ab} error for 14 (testing) color patches was 2.35 when converting from RGB to L*a*b*. In other research, a Generalized Regression Neural Network (GRNN) model was used to model the transformation between CMYK and L*a*b using the ECI2002 printer target [11]. The accuracy of the GRNN process for 171 (testing) patches was 1.82 ΔE^*_{ab}. In general, for a conventional ICC-based lookup table, generated with commercial software, the accuracy of an output profile for a CMYK device is around 2.0-4.0 ΔE^*_{ab} [15]. These results provide a 'ball-park' figure for the expected accuracy of any transform relationship and show that the algorithm developed in this project is extremely accurate.

4 Discussion

The research produced a valuable comparison between a standard ICC profile and a neural network solution. Experimentation and variables were examined to determine the most suitable neural network solution, that produced the lowest fit error.

It is important to note that an ICC profile is a quantized lookup table that may suffer interpolation inaccuracies, while the neural network algorithm is a model that can naturally generalize, in essence acting as a continuous mapping. For run-time applications it is often necessary to use a quantized lookup table and accept the lower quality; however interpolation is then needed as a final stage. In a feedforward state, the trained neural network is fast and can function using a small memory footprint, without the need for interpolation.

Some major areas of this investigation are ongoing. In this project we considered the "forward", i.e. the device to L*a*b* transform, it is usually also necessary to consider the "reverse" transform, from L*a*b* to device. Another aspect that is being examined is extension of the model to include CMYK to L*a*b* characterizations, i.e. the situation of mapping a CMYK (4-channel) input to a L*a*b* (3-channel) output.

Acknowledgements

N. Fdhal and A. Sharma are supported by a Canadian NSERC Discovery Grant (341606-2008).

References

1. Sharma, A.: Understanding Color Management. Thomson, New York (2004)
2. Sharma, A.: ICC Color Management: Architecture and Implementation. In: Lukac, R., Plataniotis, K.N. (eds.) Color Image Processing: Methods and Applications. CRC Press, Boca Raton (2006)
3. Kang, H.R.: Color Technology for Electronic Imaging Devices, Bellingham, WA (1997)
4. Connolly, C., Fliess, T.: A Study of Efficiency and Accuracy in the Transformation from RGB to CIELAB Color Space. IEEE Trans. Imag. Proc. 6, 1046–1048 (1997)
5. Berns, R.S., Shyu, M.J.: Colorimetric characterization of a desktop drum scanner using a spectral model. J. Elec. Imag. 4, 360–372 (1995)
6. Sharma, A., Gouch, M.P., Rughani, D.N.: Generation of an ICC profile from a proprietary style file. J. Imag. Sci. 46, 26–32 (2002)
7. Asakawa, K., Sugiura, H.: High-Precision Color Transformation System. IEEE Trans. Cons. Elect. 41, 304–312 (1995)
8. Vrhel, M.J.: Approximation of color characterization MLUTS with artificial neural networks. In: Proc. Int. Conf. Image Processing, pp. 465–468 (2003)
9. Chuan, Z., Shi-Sheng, Z.: Research on Color Space Transformation Model between RGB and L*a*b* Based on BP Neural Network. In: Int. Conf. on Comp. Sci. and Software Eng., vol. 6, pp. 306–308 (2008)
10. Abet, S., Marcu, G.: A neural network approach for RGB to YMCK color conversion. IEEE Frontiers of Computer Technology 1, 6–9 (1994)
11. Congjun, C., Jing, S.: Study on Color Space Conversion between CMYK and CIE L*a*b*. In: Int. Conf. on Comp. Sci. and Software Eng., vol. 6, pp. 275–277 (2008)
12. Chuan, Z., Shi-Sheng, Z.: Research on Color Space Transformation Model between RGB and L*a*b* Based on BP Neural Network. In: Int. Conf. on Comp. Sci. and Software Eng., vol. 6, pp. 306–308 (2008)
13. Duda, R.O., Hart, P.E., Stork, D.G.: Pattern Classification. Wiley, New York (2000)
14. Wallner, D.: Building ICC Profiles - the Mechanics and Engineering, Sun Microsystems (2000)
15. Sharma, A.: Methodology for Evaluating the Quality of ICC Profiles. J. Imag. Sci. Tech. 50, 469–480 (2006)