Color Camera Characterisation Using Artificial Neural Networks

T.L.V. Cheung and S. Westland
Colour & Imaging Institute, University of Derby, Kingsway House
Kingsway, Derby, United Kingdom

Abstract

A number of different methods exist for the color characterization of imaging devices such as digital camera systems. In this study, the use of high-order polynomials and artificial neural networks for color camera characterization are compared and contrasted. A quantitative evaluation of their performance is determined for a typical commercial camera system. The importance of independent training and testing sets is stressed and the effect of the number of samples in the training set is evaluated. The results show that, if the best performance is considered, the two models are approximately comparable. Any performance advantage obtained from using a neural network for device characterization does not seem to be warranted given the additional risks of using such systems. The effect of training set size seems surprisingly small for both polynomial and neural systems with generalization performance only being seriously affected for training set sizes less than about 100.

Introduction

Several methods exist[1,2] to enable camera RGB data to be transformed into device-independent CIE XYZ data and such colorimetric transformations are referred to as device characterization[3]. The characterization procedure typically involves two main processes. Firstly, the camera system is used to ascertain sensor values for targets with known color characteristics (CIE values). Secondly, these sensor values are transformed to match the target CIE co-ordinates. The aim of this study is to consider two of the most widely used techniques for camera characterisation (polynomials and artificial neural networks) and to quantitatively compare their performance when used to characterize a digital camera system. The approach described uses separate sets of data for developing and testing the models; these two sets of data are referred to as training and testing data and performance is evaluated for sets of training data containing various numbers of different samples.

Characterization Methods

If camera spectral sensitivities are identical to, or a linear transform of, the human spectral sensitivities or the CIE color-matching functions, then camera responses can be mapped to CIE tristimulus values using a simple matrix equation thus

$$\boldsymbol{t} = \boldsymbol{M}\boldsymbol{r} \quad (1)$$

where $\boldsymbol{t}$ is a 3×1 column vector of tristimulus values, $\boldsymbol{r}$ is a 3×1 column vector of camera responses, and $\boldsymbol{M}$ is a 3×3 transfer matrix that defines a linear transform. Given sufficient examples of the mapping (that is, camera response vectors and their corresponding tristimulus values) it is possible to compute the entries of the matrix $\boldsymbol{M}$ using simple techniques of linear algebra. These techniques solve $\boldsymbol{M}$ to provide a least-squares fit to the CIE tristimulus values. In practice, however, the responses of many camera systems are non-linear combinations of the CIE color-matching functions and consequently simple linear transforms may be inadequate to characterize the camera. However, characterization using such higher-order polynomials can still be represented by an equation similar to Equation 1 but the matrix $\boldsymbol{M}$ becomes larger and the CIE tristimulus values are a function of non-linear camera-response terms and interactions of these terms. Thus, the relationship given as

$$X = a_1R+a_2G+a_3B+a_4RG+a_5RB+a_6GB+a_7R^2+a_8G^2+a_9B^2+a_{10} \quad (2)$$
$$Y = a_{11}R+a_{12}G+a_{13}B+a_{14}RG+a_{15}RB+a_{16}GB+a_{17}R^2+a_{18}G^2+a_{19}B^2+a_{20}$$
$$Z = a_{21}R+a_{22}G+a_{23}B+a_{24}RG+a_{25}RB+a_{26}GB+a_{27}R^2+a_{28}G^2+a_{29}B^2+a_{30}$$

where XYZ are the tristimulus values and RGB are the camera responses, can be represented by a linear system using a 3×10 transfer matrix $\boldsymbol{M}$. Fortunately, the entries of the matrix $\boldsymbol{M}$ can be determined using linear algebra in exactly the same way as for the linear transform. The number of terms that could be used in the polynomial are theoretically unlimited but in practice must be limited by the number of known samples that are used to solve the equations for $\boldsymbol{M}$.

An alternative procedure is to use artificial neural networks to perform mappings between camera responses and CIE values. There are many types of artificial neural network that have been developed from a style of computing, known as neural computing, that was inspired by an understanding of the operation of biological nervous systems.[4] Of particular interest, however, are a type of neural network known as a multi-layer perceptron (MLP). Such MLPs are of great practical interest since it has been shown that an MLP with a single hidden layer can approximate any function to any degree of accuracy. A detailed description of MLPs is not given here but some key features are now described. AN MLP consists of layers of processing units where the units in the first (or input) layer take their input from an input vector (in this case the vector ***r***) and those in the final (or output) layer generate an output vector (in this case ***t***). Each processing unit computes a non-linear function of its input which then becomes the input for the units in the next layer. Between the input and output layers are one or more hidden layers. Only one hidden layer has been used in this work (the number of units in the input and output layers are fixed at three (being defined by the dimensions of the vectors ***r*** and ***t***) but the number of units in the hidden layer needs to be empirically determined. The network thus performs a non-linear mapping between the input vector ***r*** and the output vector ***t***. The input to each unit is a weighted sum of the outputs in the previous layer and the values of these weights need to be determined to allow the network to perform the mapping. Standard iterative techniques exist to determine the values of the weights that minimize the squared error between the output of the network and the vector ***t*** given a number of examples of the input-output relationship ***r*** → ***t***.

Experimental

An Agfa digital StudioCam camera was used in this study. The camera is a three-chip CCD device with 12-bit resolution for each channel and 4500 × 3648 pixels. During the experiment the automatic white-balance setting was disabled. The lighting system consisted of two gas-filled tungsten lamps arranged approximately in a 0/45 illumination/viewing geometry. Two imaging targets, the Macbeth ColorChecker DC chart and the Macbeth ColorChecker chart, were used as characterization stimuli for memorization and generalization respectively. Memorization represents the ability of a system to back-predict the training data that were used to determine the system. Generalization represents the ability of a system to predict testing data that were not used to develop the system and this is a more critical test of the characterization models. The spectral properties of the patches on the chart and the illumination used were measured using a Macbeth ColorEye 7000A spectrophotometer and a Minolta CS1000 spectroradiometer respectively and CIE tristimulus values were computed for the patches using the 1964 CIE observer data. A series of Munsell grey chips were used to allow a gamma correction for the camera so that the camera responses could be converted to values that were linearly related to input luminance. Spatial correction was also performed to minimize the effect of any spatial non-uniformity of the illumination.

The 192 patches in the Macbeth ColorChecker DC chart were used as training data and the 24 patches of the Macbeth ColorChecker chart were used as testing data. Smaller training sets were derived by randomly sub-sampling the 192 patches to generate training sets containing 160, 130, 100, 70 and 40 samples. All computations were performed using MATLAB. The coefficients of the polynomial methods (determined using pseudoinverse methods) and the weights in the neural networks (determined using a standard optimization algorithm) were both obtained to yield least-square fits to the training data.

Results

Figure 1 shows the performance of the polynomial models for both training and testing sets using the full 192 training set. In theory, as the complexity of the model increases we should expect the training error to consistently decrease whereas the testing error should reach a minimum and subsequently increase as the model over-fits the training data. Performance for the training set does generally decrease, whereas there is some evidence that the performance on the test set is approaching a minimum for the 3 × 16 model.

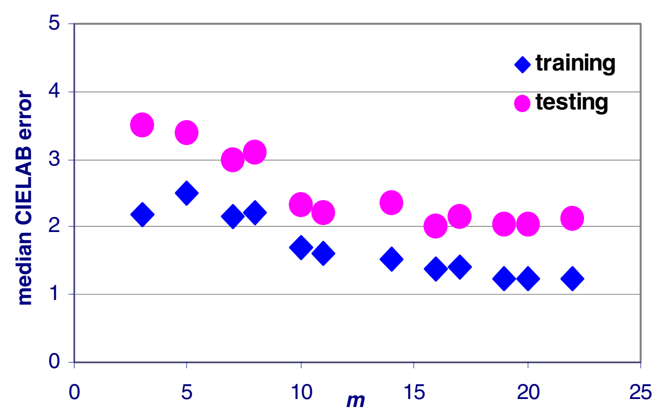

Figure 1. Performance of polynomial models of order 3 × m.

Figure 2 shows similar data for the neural-network models. It is important to note that the values of *m* and *n* (in Figures 1 and 2) cannot be directly compared. The number of free parameters in the neural networks is invariably greater than in the polynomial models. The key point of using separate training and testing sets, however, is to be able to correctly ascertain the limit of the complexity of the model that can be used given the complexity of the problem being solved and the number of training examples available to characterize that problem. In Figure 2 it is evident that, given a training set size of 192, a neural network with about 18 hidden units is optimum and increasing the complexity of the network thereafter only leads to poorer generalization performance as the network over-fits the training data.

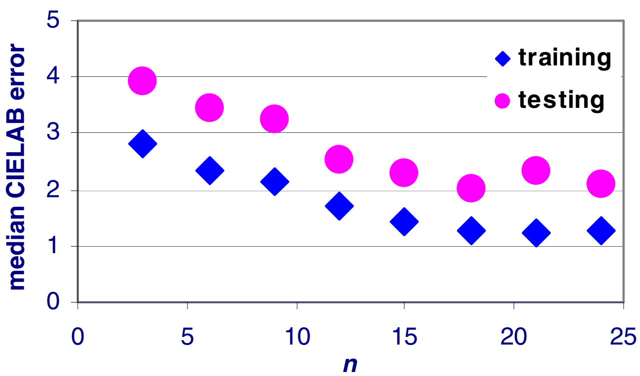

Figure 2. Performance of neural models with n hidden units.

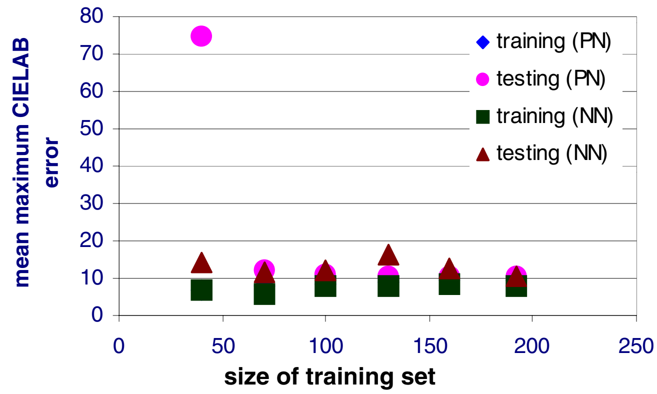

Figure 4. Effect of size of training set on maximum error for best neural (NN) and polynomial (PN) models.

The effect of reducing the number of training examples for both models is illustrated in Figure 3. Figure 3 shows data for the neural network model with 18 hidden units and the polynomial model with a matrix of size 3×16.

In Figure 3, the median error scores were averaged over five separate runs with the training set being randomly each time. For both models, the memorization error decreases as the size of the training set decreases but, as anticipated, the reverse trend is seen for the generalization performance. Somewhat surprisingly, however, generalization error is quite stable until the training set size falls to less than about 100 samples. In the limit, as the number of training samples becomes large, the performances of the models are statistically indistinguishable.

Finally, Figure 4 shows data similar to Figure 3 but the maximum color-difference error is reported rather than the median. The maximum test error for the neural network is generally larger than for the polynomial except when the number of training examples is very small, when the neural network maintains a reliable performance.

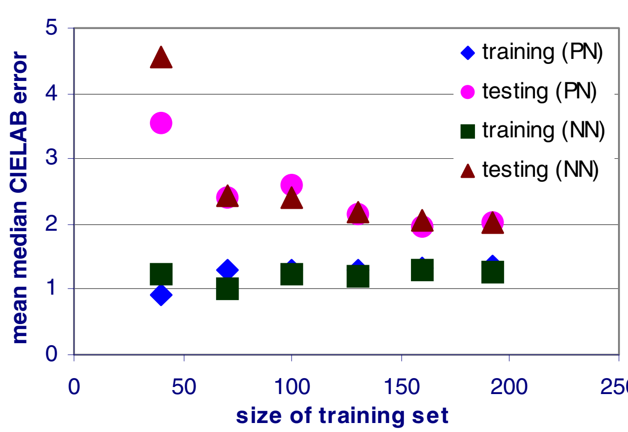

Figure 3. Effect of size of training set for best neural (NN) and polynomial (PN) models.

Discussion

This study has shown that, when correctly assessed, the abilities of camera characterization models based on neural networks and polynomials are approximately the same. This should not be surprising, since MLPs have often been described as being non-linear curve fitters or generalized polynomial models. There seems to be no advantage in using a neural network model rather than a polynomial models for this sort of problem. With polynomial models the user needs to ascertain the exact nature of the best polynomial and this can only be achieved by rigorous experimentation as in this study. However, with neural networks the user faces a similar problem since the number of hidden units in the network needs to be empirically determined. The effect of the size of the training set on generalization performance was similar for the two types of model with a large difference only evident for very small training sets. There was some evidence that the maximum color difference error was greater for the neural network than for the polynomial. There are disadvantages, however, to the neural approach. The training of the network can be very slow, sometimes taking more than one hour, whereas the polynomial model can be solved in fractions of a second. There is also some evidence that the neural network models may not always converge to a global minimum whereas for the polynomial implementation of training and testing is easy and reliable.

References

1. G. Hong, M. R. Luo and P. A. Rhodes, *Color Research and Application*, **26**, 76 (2001).
2. S. Tominaga, Colour co-ordinate conversion via neural networks, in *Colour Imaging, Vision and Technology*, eds. L.W. MacDonald and M.R. Luo, John Wiley (1999).
3. T. Johnson, *Displays*, **16**, no. 4, 183 (1996).

4. I. Aleksander and H. Morton, *An introduction to neural computing*, Chapman & Hall (1991).

Biography

Tsz Lock Vien Cheung graduated from The Hong Kong Polytechnic University with a BSc degree in Textile Chemistry. She then obtained a MSc in Colour Imaging at the Colour & Imaging Institute at University of Derby. She is currently a postgraduate student in the Colour Imaging Institute working on methods for device characterization and multispectral imaging.