APPROXIMATION OF COLOR CHARACTERIZATION MLUTS WITH ARTIFICIAL NEURAL NETWORKS

Michael J. Vrhel

ViewAhead Technology
18200 NE Union Hill Road
Redmond, WA 98052

ABSTRACT

MLUTs used in color characterization require significant memory for embedded systems. Most tools that create device characterization maps, use MLUTs. For this reason, there is interest in approximating MLUT color characterization mappings with more compressed function approximation methods. In this document, artificial neural network approximations for MLUTs are investigated. Color accuracy, computational loads, and memory requirements are discussed.

1. INTRODUCTION

Multidimensional look-up-tables (MLUTs) are commonly used for the approximation of the mapping between device dependent color values and device independent color values [1]. MLUTs are also used for mappings between device dependent values for different devices, such as the mapping from a scanner's RGB values to a printer's CMY or CMYK values, or a digital camera's RGB values to a standard RGB space such as sRGB.

MLUTs have the advantage of being very flexible and relatively easy to adjust. Often, in the creation of an MLUT, the table entries are adjusted to create effects such as saturated colors, or to implement specific gamut mapping methods. A disadvantage of MLUTs is that they can be large for embedded systems with limited memory.

Artificial neural networks (ANNs) provide a more compressed form of function approximation compared to MLUTs. By approximating the MLUT mapping instead of using the original data to create an ANN, it is possible to use the existing MLUT adjusting and optimizing tools to obtain the mapping that provides the ideal visual effect. The ANN is then used to approximate this ideal MLUT mapping.

2. MLUT COLOR MAPPINGS

Let the true ideal color mapping between the M dimensional color space C and the N dimensional color space D be given by the mapping $\mathcal{F}$ where by definition

$$\mathcal{F}(\mathbf{c}) \in D \tag{1}$$

for $\mathbf{c} \in C$.

For simplicity (and to consider only cases of interest) let us assume that the elements of the vectors in C range from 0 to P. In addition, we will assume that $M = 3$. In this case, the entries for a uniform sampled MLUT of size $N * R^3$, which approximates the mapping $\mathcal{F}$ are defined by

$$\mathcal{F}(\mathbf{c}_{i,j,k}) = \mathbf{d}_{i,j,k} \quad i,j,k \in [0,...,R-1] \tag{2}$$

where

$$\mathbf{c}_{i,j,k} = \left[\frac{P*i}{R-1}, \frac{P*j}{R-1}, \frac{P*k}{R-1}\right]^T \tag{3}$$

The approximation of a value $\mathcal{F}(\mathbf{c}) = \mathbf{d}$, which is not on a sample point on the grid (i.e. $\mathbf{d} \neq \mathbf{d}_{i,j,k}$ is calculated by an interpolation method. Two common MLUT interpolation methods are trilinear interpolation and tetrahedral interpolation [2].

Tetrahedral interpolation reduces the number of points used in the interpolation calculation by dividing the cube containing the value $\mathbf{c}$ into subsections. An additional test consisting of three compares is required to determine what subsection contains the vector $\mathbf{c}$. Each subsection uses only 4 points to compute an interpolation as opposed to the 8 points used in the trilinear interpolation.

3. ANN COLOR SPACE MAPPING

Similar to MLUTs, feed-forward ANNs are a general method for approximating nonlinear mappings. The difference between the two methods is that the ANN generally requires significantly less memory than an MLUT to approximate a given nonlinearity. The cost of this memory reduction is computational complexity.

ANNs have been used to implement color mappings [3]. However, they have not been commonly used to approximate color mappings already defined by MLUTs. The advantages of this approach have already been mentioned.

0-7803-7750-8/03/$17.00 ©2003 IEEE.

The architecture of the ANN should ideally be designed based upon *a priori* knowledge of the function that the network is to approximate. It can be shown that a 2-layer feed forward network with a sigmoidal nonlinearity can be used to approximate any function with a finite number of discontinuities, arbitrarily well, given sufficient neurons in the hidden layer [4]. For this reason, we will concentrate on this architecture.

Note that the mapping defined by the MLUT from the three dimensional space C to the N dimensional space D can be defined by N vector valued mappings. It is each of these mappings that we wish to approximate with an ANN.

The input-output relationship of our 2-layer feed-forward ANN with S neurons in layer 1 is expressed as

$$\mathcal{L}(\mathbf{c}) = \mathbf{v}^T \Phi(\mathbf{W}\mathbf{c} + \mathbf{b}) + e \tag{4}$$

where $\Phi(\mathbf{x}) = [\phi(x_1), ..., \phi(x_S)]^T$, $\mathbf{b}$ is an S element vector, $\mathbf{W}$ is an $Sx3$ matrix, $\mathbf{v}$ is an S element vector, e is a scalar, and ϕ is the sigmoidal function, which is given by

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

The coefficients $\mathbf{v}$, $\mathbf{b}$, $\mathbf{W}$, and e are optimized through an iterative back-propagation method. The initial conditions of this method can have a significant affect on the value to which it converges.

The determination of the number of coefficients required to achieve a given level of approximation error depends on the smoothness of the actual function. In practice, some experimentation and inspection of the values after convergence is necessary to optimize the parameter S.

4. MEMORY REQUIREMENTS

Assuming an 8 bit/channel representation for color data, The memory requirements for computing CMYK color values from scanner RGB values with an MLUT of size $RxRxR$ is given by

$$SIZE_{MLUT}(R) = 4 * R^3 \ \text{ BYTES} \tag{6}$$

This is true regardless of the interpolation method.

Assuming the data is well-scaled, the ANN coefficients can be well represented by 16 bit words. Also, the sigmoid function can be well approximated by a 256 element 1-D LUT of 16 bit words. The memory requirements for the ANN to compute CMYK color values is given by

$$SIZE_{ANN}(S) = 512 + 4 * (S * 3 + S + S + 1) * 2 \ \text{ BYTES} \tag{7}$$

Note that for both approaches, it would be possible to compute each color plane separately, which would reduce memory requirements at the cost of more traffic and computations.

5. COMPUTATIONAL REQUIREMENTS

Assuming an embedded system with a 16 bit word length, the computational requirements for mapping from RGB to CMYK were calculated for the MLUT with trilinear interpolation, the MLUT with tetrahedral interpolation and the four ANNs each with S neurons. In this case, no particular optimizations were made in terms of what operations could be performed in parallel. Rather, each addition, multiply, bit-wise shift, look-up, bit-wise AND, compare, and switch were calculated. The results are given in Table 1. Note that the MLUT computation is independent of the table size.

Table 1. Computational requirements for methods

Operation	Trilinear	Tetrahedral	ANN
Look-up	32	28	$24 * S$
Bit-wise shift	12	13	$16 * S + S$
Addition	41	36	$20 * S$
Multiply	47	14	$4 * S$
Bit-wise AND	3	3	0
Compare	0	3	0
Switch	0	1	0
Total	135	98	$65 * S$

6. EXAMPLE

To investigate the feasibility of approximating a real color MLUT, we approximated the MLUT that maps the RGB values from a Brother scanner to CMY values for a Lexmark ink-jet printer. The MLUT was of size $R = 17$. An ANN was trained for each of the three color channels. Networks of several sizes were created. The networks were optimized using the Levenberg-Marquardt algorithm. Five different initial conditions were tried in the training process for each color channel and the convergence that achieved the smallest error was selected. Once the network was computed, several different inputs were applied to the MLUT with trilinear interpolation and to the network to assess its accuracy. All 17^3 grid point vector values were used in training the networks.

6.1. Gray Scale

To assess the accuracy of the fit along the gray scale axis, the values

$$[i, i, i]^T \quad i = 0, 1, 2, ..., 255$$

were provided as an input to the ANNs and to the MLUT. The CMY output for the MLUT and the ANN are shown in Figures 1-3 for $S = 16$.

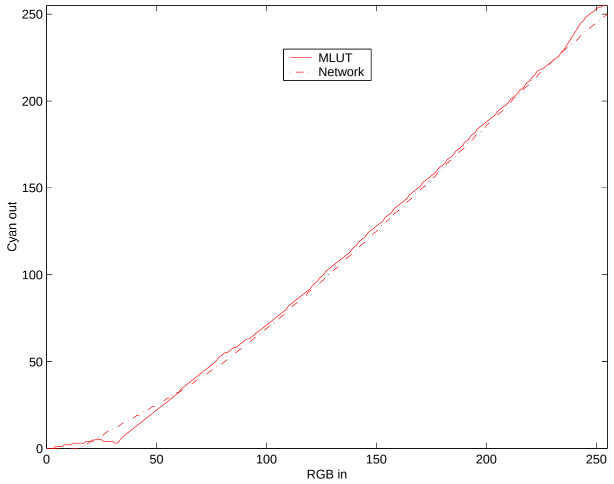

Fig. 1. Cyan output, MLUT and ANN for input $[i, i, i]^T$ $i = 0, 1, ..., 255$

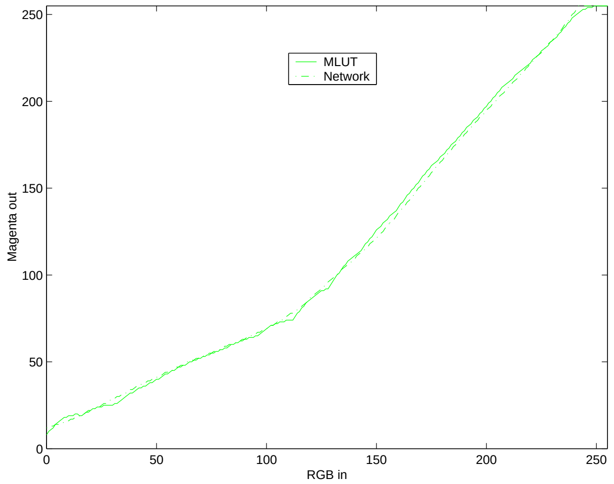

Fig. 2. Magenta output, MLUT and ANN for input $[i, i, i]^T$ $i = 0, 1, ..., 255$

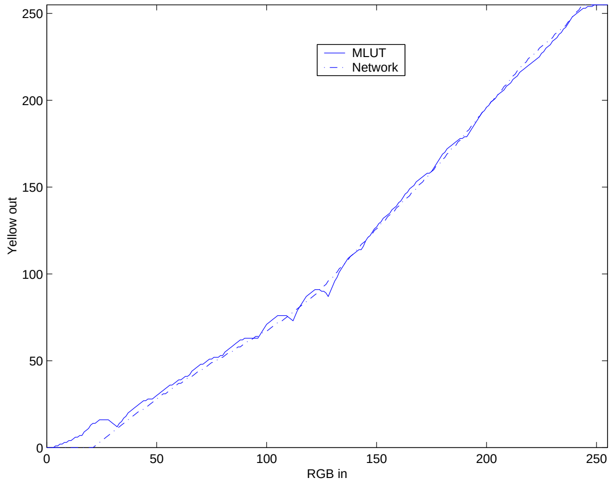

Fig. 3. Yellow output, MLUT and ANN for input $[i, i, i]^T$ $i = 0, 1, ..., 255$

There are differences visible in the plots between the two outputs. One slightly surprising result is the lack of smoothness in the MLUT output, in particular the yellow channel has significant variations, which the ANN smooths.

6.2. Extreme color edges

The RGB spectrum image shown in Figure 4 was applied as input to the MLUT and the ANNs. The spectrum image has the property that at every pixel at least one element of the RGB vector is 0 or 255. This implies that the image consists completely of values that are at the edges of the range of the color space C. This is a useful image to assess the match between the two mappings at the extreme edges of the color space. Figure 5 is the output of the MLUT with trilinear interpolation and Figure 6 is the output of the ANNs with $S = 16$. Note that a real scanner may not even create values at many of these points.

6.3. Color Accuracy

To quantify the color errors introduced by the approximation, a grid of 7x7x7 RGB values were fed through each of the networks and the MLUT. The CMY output values were printed and the CIELab values for each color were measured with a Gretag spectrophotometer. The $\Delta E94$ difference between the MLUT values and the ANN values were computed. Statistics of the $\Delta E94$ differences are contained in Table 2.

Multiple prints were created of the MLUT chart and measured with the spectrophotometer to obtain a baseline variability. Figure 7 displays a plot of the average $\Delta E94$ values and the process variability.

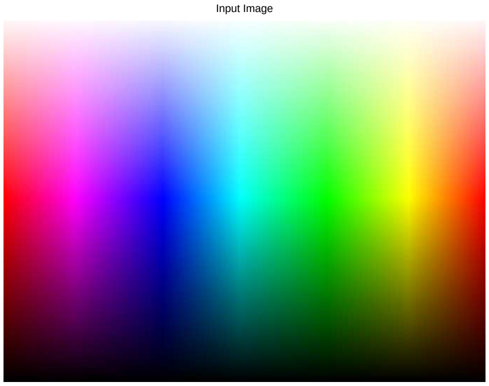

Fig. 4. RGB spectrum input image

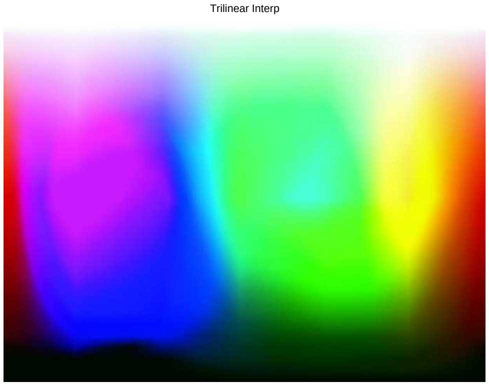

Fig. 5. MLUT output for spectrum input

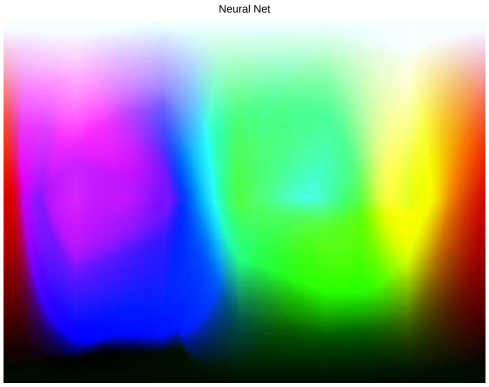

Fig. 6. Neural net output for spectrum input

Table 2. $\Delta E94$ difference between MLUT and ANNs

Network Size	Avg. ΔE	$\%\Delta E > 3.0$
5	2.7	33.8%
10	1.9	14.0%
15	1.7	7.0%
20	1.6	5.8%
30	1.5	5.5%

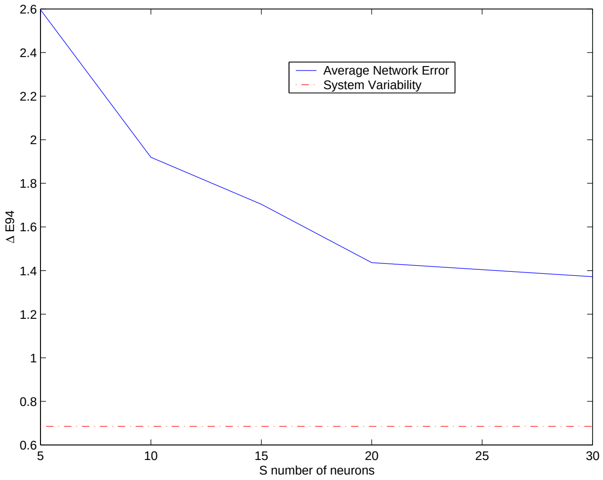

Fig. 7. Average $\Delta E94$ error versus number of neurons

As expected, an increase in the number of neurons decreases the color error. Color errors larger than 3 will be clearly visible, and error less than 1 will not be visible to the standard human observer. The error levels achieved, indicate that in restrictive memory applications with significant computational resources, the ANN approximation may be a useful approach.

7. REFERENCES

[1] M. J. Vrhel and H. J. Trussell, “Color Device Calibration: A Mathematical Formulation,” *IEEE Trans. on Image Proc.*, Vol. 8, No. 12, Dec. 1999.

[2] H. R. Kang, Color Technology for electronic devices, SPIE Press, Bellingham WA, 1997.

[3] H. R. Kang and P. G. Anderson, “Neural Network Applications to the Color Scanner and Printer Calibrations,” J. of Elec. Imag., Vol. 1, No. 2, April 1992.

[4] K. Hornik, M. Stinchcombe and H. White, “Multilayer feedforward networks are universal approximators,” *Neural Networks*, Vol. 2, No. 5, pp. 359-366, 1989.