Device Characterization Using Variable Rectilinear Interpolation

C.T. Yeung and Bob Lin
Peerless System Corp., 2381 Rosecrans Ave.
El Segundo, California 90245

Abstract

We propose variable rectilinear interpolation (VRI) as an alternative method for device characterization. Based on a rectilinear structure, this algorithm constructs a mapping with non-uniform nodes placed to maximize informational entropy while meeting the pre-defined error tolerance threshold. In this paper we describe the VRI algorithm and structure; followed by an example in which we characterize the Minolta Color PageWorks™ color printer.

Introduction

Numerous mapping strategies have been proposed for color transforms. Among the most common techniques are rectilinear interpolation, sequential linear interpolation (SLI), and back propagating neural network (BPN).

In rectilinear interpolation, a uniform grid of nodes is used to build a three dimensional look up table (LUT).[1,2] This table represents transform coordinates from original to destination color spaces. Bisection or other similar quick sort algorithm is used to locate the sub-volume nodes surrounding the interpolation point in the transformation. The advantage of such a system is efficient sorting and simple implementation.

SLI is initialized with a uniform grid structure similar to that used in rectilinear interpolation. To obtain optimal informational entropy, sequential scalar quantization (SSQ) is used to reduce the number of nodes in its three dimensional LUT. This results in freely distributed nodes with minimum redundancy.[3,4]

Using a different approach, neural network develops its mapping by improving its interpolation matrix a little at a time. For most BPN designs, the initial matrix is small and random. The matrix structure, dimension and values are modified based on iterative samples of desired mapping values, until an acceptable error tolerance is reached.[5,6]

We present variable rectilinear interpolation (VRI), which combines characteristics of the above methods, as an alternative. We describe the VRI algorithms, then follow it with results of an experimental application in which we characterize a laser print engine.

VRI Algorithms

Initialization

As the name indicates, the VRI structure is composed of rectangular sub-volumes in variable sizes. The algorithm starts with the largest rectangular volume that circumscribes all points in color gamut (see figure 1).

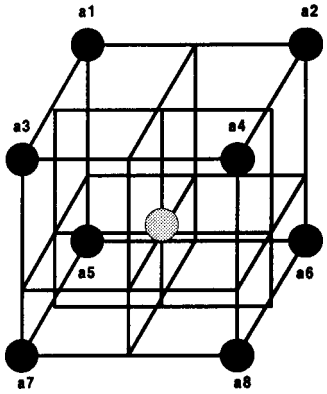

Figure 1. Initial boundary nodes with a potential node in middle.

This rectangular volume is defined by its boundary eight node points, al, a2, a8. In most practical situations, these eight points are out of gamut because color gamut in most spaces is unlikely to match this volume. Normally nearest points or minimum delta E extrapolation is used to estimate these out of gamut boundary points. Since the topic of estimating these points is beyond the scope of this paper, let us assume that boundary nodes (a1...a8) in our original RGB space are equal to sample values (a1'...a8') in our destination space.

Minimizing Error

The objective of VRI, like that of SLI, is to establish nodes at locations that maximize informational entropy. The first step in building the mapping structure is to generate a series of uniform data points representing the original (R_o, G_o, B_o) color space and measure the corresponding points (R_d, G_d, B_d) in the destination color space.

Next, we calculate estimated values (R_i, G_i, B_i) for these points in the destination color space by tri-linear interpolation.[1] The interpolated results are compared with the measured sample values to yield a table of delta errors (see eq 1).

$$\Delta E = \sqrt{\left(R_d - R_i\right)^2 + \left(G_d - G_i\right)^2 + \left(B_d - B_i\right)^2} \qquad (1)$$

Using this error table, a node is added to the mapping structure at the location of highest delta error. This results in a reduction of the maximum mapping error.

Meeting Acceptable Tolerance

As in BPN, the error evaluation and minimization process is iterative.[6] Each rectangular sub-volume is further divided until all errors are below the pre-defined acceptable threshold. The resulting structure is a combination of fine and coarse rectangular sub-volumes (see Figure 2).

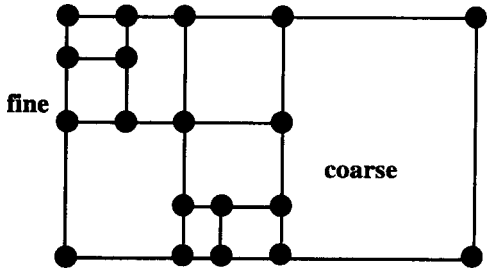

Figure 2. 2-D view of VRI structure

Constructing and Searching the Structure

Since our nodes are randomly distributed, general sorting algorithms are not applicable. Mathematically, the data is organized as a tree is which records the division of each volume into rectangular sub-volumes and nodes. The largest rectangular volume (al...a8) is the root. Each subsequent node divides the volume into eight sub-volumes (see figure 3). This process is repeated to produce a top-down pattern where errors are minimized as we travel down the branches.

To search the tree, compare the to-be-mapped RGB value to those of the root node's eight sub-volumes. Go down the tree in the direction of the volume that contains the RGB value. The correct interpolation sub-volume is iteratively reached when we reach an end volume.

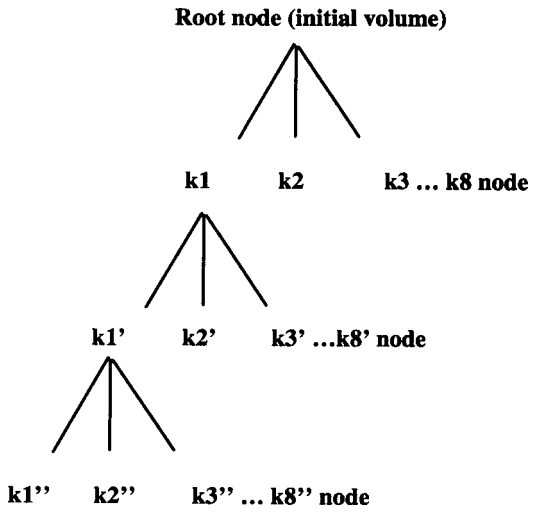

Figure 3. Tree structure for organizing look-up table

Experiment

To validate our algorithm in practice, we evaluated VRI by characterizing a Minolta Color PageWorks™ color printer. Two particular issues were addressed, efficiency and VRI structural integrity.

Data Generation

Data for the Minolta color printer was generated by printing 16 × 16 × 16 uniformly spaced colors in device RGB space. The samples were measured with an X-Rite colorimeter in XYZ space. To correlate the color values between source and destination space, we chose to convert XYZ back to device RGB. The decision of device RGB space over CIE L*a*b* or other device-independent space was arbitrary. Equation (2) and (3) below were used to convert data from RGB to XYZ color space,[8,9] and the transformation from XYZ to RGB was done by multiplying with the inverse of the transformation matrix [T].

$$\begin{bmatrix} v_1 \\ v_2 \\ v_3 \end{bmatrix} = \begin{bmatrix} x_r & x_g & x_b \\ y_r & y_g & y_b \\ z_r & z_g & z_b \end{bmatrix}^{-1} \bullet \begin{bmatrix} X_w \\ Y_w \\ Z_w \end{bmatrix} \tag{2}$$

where $Y_w = 1.00$

$$T = \begin{bmatrix} x_r & x_g & x_b \\ y_r & y_g & y_b \\ z_r & z_g & z_b \end{bmatrix} \bullet \begin{bmatrix} v_1 & 0 & 0 \\ 0 & v_2 & 0 \\ 0 & 0 & v_3 \end{bmatrix} \tag{3}$$

$$\begin{bmatrix} R \\ G \\ B \end{bmatrix} = \begin{bmatrix} T^{-1} \end{bmatrix} \bullet \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} \tag{4}$$

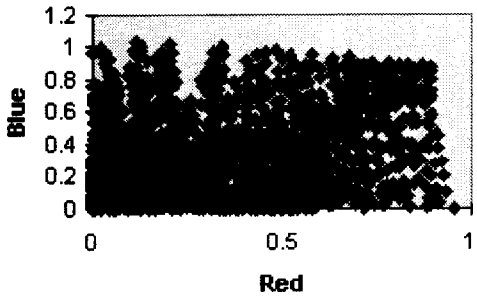

Figure 4.1. Red vs. Blue sample converted to RGB space

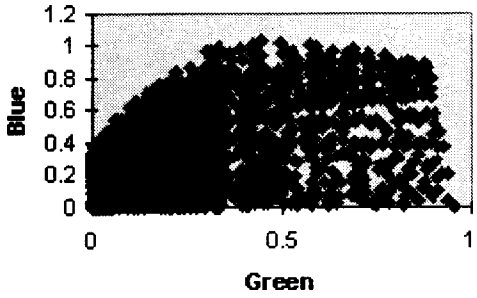

Figure 4.2. Green vs. Blue sample converted to RGB space

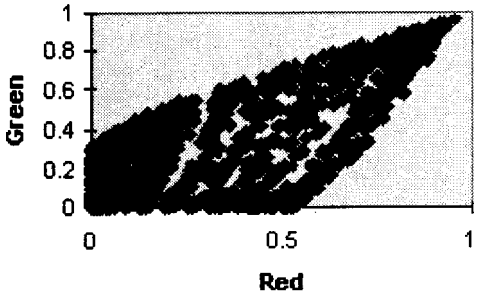

Figure 4.3. Red vs. Green samples converted to RGB space

The measured samples, converted to device RGB space, are presented in Figure 4.1-3. All data are scaled between 0 - 1.

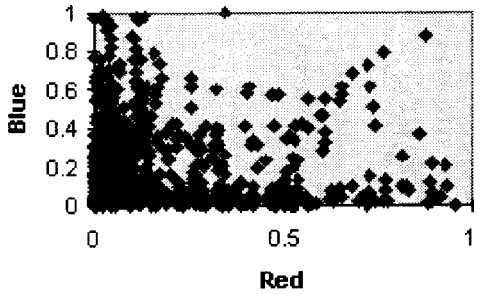

Figure 5.1. Nodes at Tolerance = .10 (Red vs. Blue)

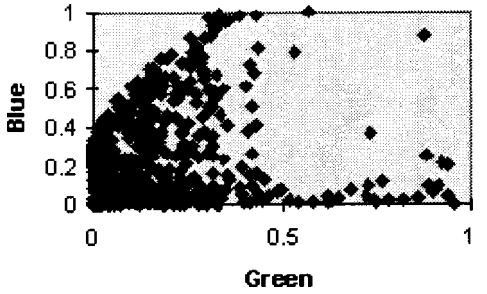

Figure 5.2. Nodes at Tolerance = .10 (Green vs. Blue)

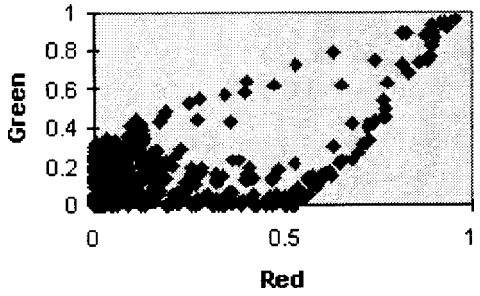

Figure 5.3. Nodes at Tolerance = .10 (Red vs. Green)

Results

A series of VRI structures were generated, in which the error tolerance was set from 0.1 to 1.0 in 0.1 increments. As an example, the node distribution for the 0.1 error threshold is shown in Figure 5.1-3.

Analysis

To illustrate efficiency, Figure 6 shows the number of nodes needed to achieve the desired accuracy. It can be seen that as we lower our error tolerance a much greater number of nodes is required without giving much improvement. The curve is close to logarithmic.

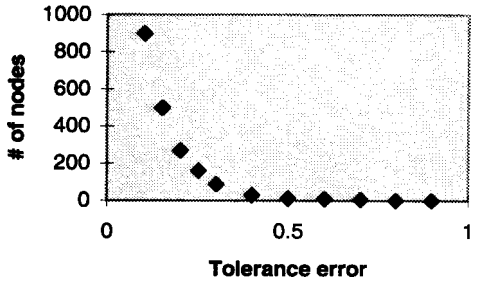

Figure 6. Tolerance error vs. number of nodes

To validate VRI structural integrity, a set of random color samples, different from the training set, was used to evaluate mapping accuracy. The minimum error achieved during the evaluation cases was 0.328. This occurred with a structure of 270 nodes set with tolerance error = 0.20. As illustrated, adding additional nodes did not improve the error for this particular evaluation data set, (see Figure 7).

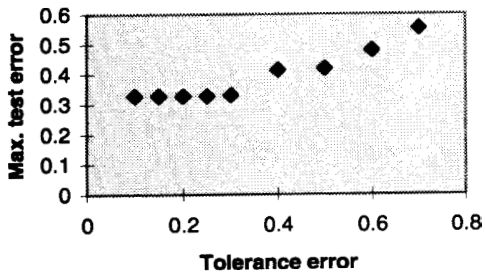

Figure 7. Tolerance error vs. maximum test error

Conclusions

In this paper, we have proposed the concept of VRI and illustrated its behavior for a printer device.

From our experiments, we conclude that an acceptable performance range was obtained between .25 to .40 error tolerance. There were no further accuracy improvements when more nodes were added.

Compared with other methods, we make the following three observations. First, the interpolation grid structure is applicable to characterize a device. VRI may be used for various 3-D color spaces such as L*a*b*, LCH, and YCC. Second, the grid point distribution in VRI results in a consistent accuracy throughout the transform volume. Third, the structure in the VRI method can be adapted to other interpolation methods such as cubic, tetrahedral, pyramidal, and prismatic interpolations since VRI maintains a rectangular structure.

It is our intend to further our evaluation of VRI in different devices and color spaces. Furthermore, additional research of interpolation strategies that yield greater accuracy could also be developed.

Acknowledgments

We appreciate Peerless System Corp. for providing us the necessary research environment. In particular, we thank our

manager Nick Reddingius for his support and direction of research activities.

References

1. P. Hung, Colorimetric Calibration in Electronic Imaging Devices Using a Look-Up-Table Model and Interpolations, *Journal of Electronic Imaging*, vol. **2**, No. 1, pp. 53-61 (1993).
2. I. E. Bell and W. Cowan, Characterizing Printer Gamuts Using Tetrahedral Interpolation, *Proc. of the First IS&T/SID Color Imaging Conference*, Scottsdale, AZ, November 1993, pp. 108-113.
3. I. E. Bell and W. Cowan, Device Characterization Using Spline Smoothing and Sequential Linear Interpolation, *Proc. of the Second IS&T/SID Color Imaging Conference*, Scottsdale, AZ, November 1994, pp. 29-32.
4. E. J. Giorgianni, A Universal Paradigm for Color Management, *Proc. of the Fourth IS&T/SID Color Imaging Conference*, Scottsdale, AZ, November 1996, pp 1-5.
5. D. Adkin, V. S. Cherkassky and E. S. Olson, Color Mapping Using Neural Networks, *Proc. of the First IS&T/SID Color Imaging Conference*, Scottsdale, AZ, November 1993, pp. 45-48.
6. T. Masters, Signal and Image Processing with Neural Networks: a C++ sourcebook, John Wiley & Sons, New York (1994).
7. T. Fumoto, K. Kanamori, O. Yamada, H. Motomura, H. Kotera and M. Inoue, SLANT/PRISM Convertible Structured Color Processor MN5515, *Proc. of the Third IS&T/SID Color Imaging Conference*, Scottsdale, AZ, November 1995, pp. 101-105.
8. D. Travis, *Effective Color Displays: Theory and Practice*, Academic Press, London (1991).
9. R. W. G. Hunt, *Meauring Color*, 2nd ed., pp. 199-208, Ellis Horwood Limited, England (1992).

☆ This paper was previously published in *IS&T/SID 5th Color Imaging Conference Proc.,* p. 193 (1997).