Using Shepard's Interpolation to Build Color Transformation Tables

Dr. Robert Rolleston
Xerox Digital Imaging Technology Center, Webster, New York

Abstract

A method for constructing a multidimensional color space transformation look up table (LUT) is presented. The values associated with each node of the LUT are calculated by a weighted sum of known values. Unlike the tetrahedral inversion techniques, this method does not require the tessellation of the space. Changing the weighting function can produce LUT's with different properties, and the method has the ability to extrapolate values beyond the gamut of the device.

Introduction

Color displays and printers have moved into the offices and onto the desktops, and with this move come an assortment of customer expectations. At first customers were happy to simply produce color, but now they want to produce color easily, accurately and with pleasing effects and no surprises. Vendors are trying to meet these demands with the use of device independent color. If device independent color is to succeed, then the assortment of scanners, displays, printers, and applications all involved in the production of color documents will all have to understand the same device independent language. Furthermore, for each device or application to be used to its fullest potential, it will often be necessary to accurately translate either to or from this device independent language to a device dependent language.

The transformation between a device dependent color space and a device independent color space requires the calibration and characterization of a device, and then the determination of an appropriate algorithm for doing the transformation. It has become common to perform some part or all of this transformation by the use of a multidimensional look up table, or LUT. This is especially true for very nonlinear devices such as printers.[1-2] The creation of these LUT's is a nontrivial task, and has been discussed in the literature.[3-5] The technique described below is one method of building these color transformation LUT's.

Statement of the Problem

Consider the case of building a LUT to convert from a device independent space, say a colorimetric red-green-blue space (RGB_C), to a device dependent space, say a printer cyan-magenta-yellow space (CMY_X). The first stage of this process requires the calibration of the printer, which will bring the printer to a known state of operation. Assume that this known state of operation results in a printer which has a response, and that there is a sufficient UCR/GCR strategy in place if this is a four color printer.

The next step of the process involves characterizing the device. This may be done by sending a set of known signals to the printer, i.e. device dependent data, and then measuring the resulting colors, i.e. device independent data. The combination of these two data sets gives a characterization of the device in terms of a mapping from device dependent data to device independent data for some sampling of the device's native color space. The problem now is to use this sampling of the device characterization to create an inverse mapping, i.e., from device independent data to device dependent data.

The device dependent data can usually be arranged to lie on a known regular grid in 3-space, and is thus amenable to any of several different interpolation schemes. For speed of computation in using the LUT it is often desirable to have the device independent data calculated for a regular grid in 3-space. Unfortunately, the colorimetric data will most likely lie in a rather scattered fashion throughout the RGB_C space, and will be located only within the gamut of the device. Values for every node in the LUT, which form a regular grid and spans the entire RGB_C space, must be calculated from a sampling of data which is scattered in the space and does not fill the space.

Discussion of the Solution

For the one dimensional problem there exist a multitude of ways to interpolate a regular set of data from an irregular set of data. Many of these methods do not have extensions to more than one dimension. One method which can be extended into multidimensional spaces is Shepard's Interpolation.[6-8] The method used here is a modification which interpolates the change in a function, rather than the function itself.[9,10] This method will be described here in terms of a two dimensional example, but it is readily extended to higher order dimensions.

Figure 1 is a plot of the data used in the characterization of a two color printer (Red and Green) printer. For simplicity, the Cyan and Magenta coordinates, (C_X, M_X), have been plotted as their color complements (R_X=1-C_X and G_X=1-M_X). If we assume the printer units are the same as the colorimetric units, we can scale both sets of data (i.e. the device dependent and device independent colorimetric data) and plot them together. In Fig. 1, colorimetric data are denoted by solid squares; device data are denoted by ends of the arrow heads; and the vectors are pointing from the colorimetric coordinates of a color sample, (R_C, G_C), to the printer coordinates used to make that same sample, (R_X, G_X). Notice that the device data lie on a rectangular 4x4 grid, while the resulting colorimetric data are scattered irregularly in 2D space.

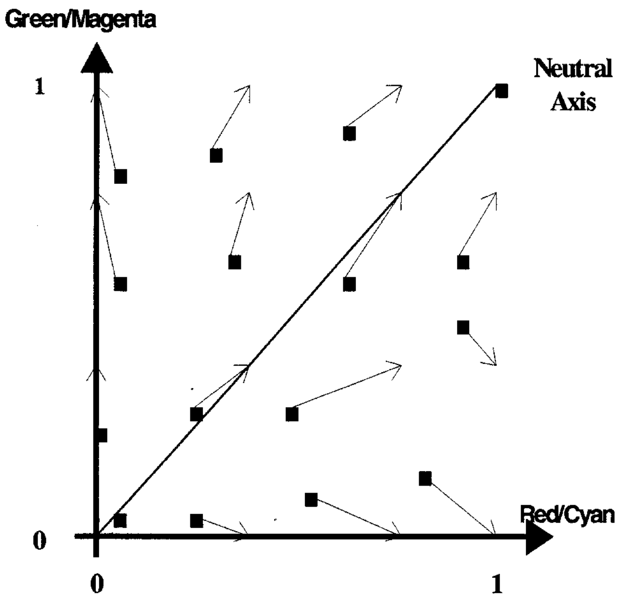

Figure 1. Vector field mapping colorimetric colors to printer colors. Solid squares are colorimetric data, arrow heads point to device data.

If the data were plotted for device dependent CMY_X coordinates instead of device dependent RGB_X coordinates, then for many colors, the vectors would be dominated by the transformation from an additive RGB space to a subtractive CMY space and small color changes may be lost. The use of this modification of Shepard's interpolation to build the LUT is most suitable when there are only small vector changes in the transformation.

The task is to find a function (or in this case a LUT) which will describe this vector field over the entire range of possible input colorimetric data. Note that while only a two dimensional example is shown here with 16 vectors, in practice it is not uncommon to have a three dimensional space with 1000 (=10x10x10) vectors.

For this example, consider using only the 4x4=16 vectors shown in Figure 1, to build an 8x8=64 element table. The table node points and the locations of the known vector transformations are shown in Figure 2. This table uses a nonlinear scaling of the axis so that there is a finer sampling of the color space where there is a high density of colorant.

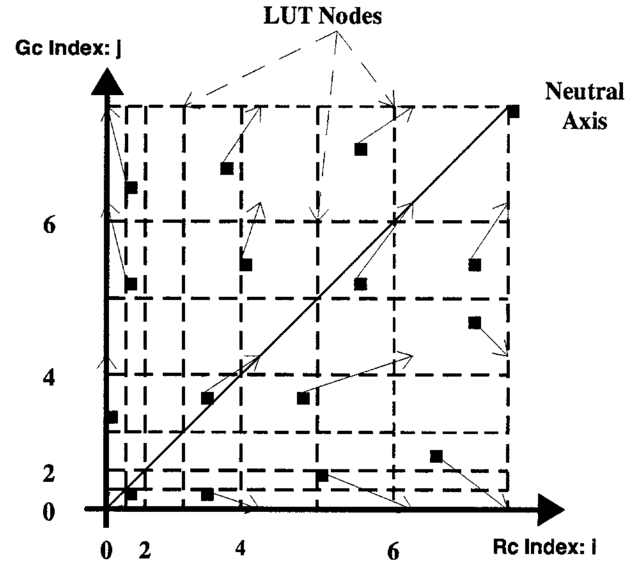

Figure 2. Vector field indicating desired mapping from colorimetric colors to printer colors. Dotted lines indicate the location of nodes in an 8x8 LUT.

A weighted average of all the sample vectors is used to calculate the value for a node in the table. Consider the table node location $(R_C[i=6], G_C[j=5])$. There is no calibration data at that particular point, but there are some vectors which are near this table location. The vectors near the point of interest should have a greater influence than vectors far away, so a weighting function which is inversely proportional to distance is used. This can be expressed in mathematical terms by calculating the weighted average of each of the vector components, (E_R, E_G), as:

$$E_R=(\Sigma(R_X[k] - R_C[k]) * W(d[k])) \; / \; \Sigma(W(d[k])) \qquad (1a)$$

$$E_G=(\Sigma(G_X[k] - G_C[k]) * W(d[k])) \; / \; \Sigma(W(d[k])) \qquad (1b)$$

where [k] is the calibration patch index, and d is the distance from the patch $(R_C[k], G_C[k])$ to the lattice node$(R_C[i], G_C[j])$, and W(d) has been chosen as the weighting function whose value is inversely proportional to the distance d. The summation is over some sets of the k known values. For node location $(R_C[i], G_C[j])$ and patch number [k]:

$$d[k]^2=(R_C[i] - R_C[k])^2+(G_C[j] - G_C[k])^2 \qquad (2)$$

Finally, the device dependent values, (R_X, G_X), inserted into the table at this node are given by:

$$R_X = R_C+E_R \qquad (3a)$$

$$G_X = G_C+E_G \qquad (3b)$$

Notice that if the printer is perfect, so that $R_C=R_X$ and $G_C=G_X$, all the vectors are of length zero, and $E_R =$

$E_G = 0.0$. The interpolation is being done such that the interpolated values are calculated as a change to the input values, and not simply a weighted average of the known values. This means that in regions far from any known data values, you will get a mapping which is the same as the average change imposed by the system, and not a mapping which is the average of the known values.

Example

Some of the effects of using different weighting function will be presented here using an even simpler one dimensional example. A one dimensional function that has been sampled at 10 points over the range [0,255] is plotted in Figure 3a. In this one dimensional example, it is easy to draw straight lines between the points, and get a piece-wise linear approximation to the actual function. In a multidimensional space it may not be so easy to find such a tessellation of the space. As plotted, this function can be thought of as representing a mapping from colorimetric space to device space. The ten known values of the independent variable corresponds to requested colorimetric values, and the dependent variable corresponds to the needed device value. The goal is find the general mapping from colorimetric values to device values, at locations other than the ten known values.

Figure 3b is a plot of the mapping functions obtained when the weighting function is either 1/d, or $1/d^4$, and all of the known points are used in the summation. Notice that both of these functions are exact at the locations of the known values, and are thus exact interpolating functions. This will be true of any weighting function which has a pole at d=0. Also notice that the linear power of 1/d produces a severe scalloping effect in between the known data values, while the higher order power of 1/d is much smoother. The low order power is seeing the effects of too many of the known values, the weighting function does not fall off fast enough. The extrapolation properties can also be observed for values less than zero and greater that 255.

The effects of limiting the number of values used in the summation can be observed in Figure 3c and Figure 3d. The mapping shown in Figure 3c is calculated using a linear inverse power of distance, but limiting the summation to include only the nearest two or five neighbors. In this case the use of only two neighbors produces an almost piece wise linear fit, while the use of five neighbors still suffers for some of the scalloping effects discussed above. The mapping shown in Figure 3d is calculated using a cubic inverse power of distance, but limiting the summation to include only the nearest two or five neighbors. In this case, there is almost no difference between the use of two neighbors, or five neighbors, the weighting function falls off rapidly enough so that the effects of far away known values are negligible. The weighting functions which are proportional to lower powers of distance are more sensitive to the choice of the number of values to use in the summation, than those weighting functions which are proportional to higher powers of distance.

The mappings obtained when using two weighting functions which do not have poles at d=0 are shown in Figure 3e and Figure 3f. The weighting function used to calculate the curves in Figure 3e is of the form exp(-Ad), where the two curves are the result of two different choices for A. The weighting function used to calculate the curves in Figure 3f is of the form $\exp(-Ad^2)$, where the two curves are the result of two different choices for A. With these choices of weighting functions, the inversion is no longer guaranteed to be correct even at the known data values. This may be advantageous if it is known that the given data is noisy, and exact interpolation is not required. In regions far from any known data values, the inversion tends to produce a discontinuity. This is most noticeable near the colorimetric value of 150.

In addition to these simple weighting functions and limitations on the summation, several other options are easily implemented. It is possible to alter the weighting function dependent upon what region of the function is being approximated. In the multidimensional case it is also quite easy to have the summation and/or weighting function have an angular dependence. Perhaps sum over the set of points with a similar hue, saturation, and chroma. The expanse of possibilities and variations makes any optimization a very interesting task.

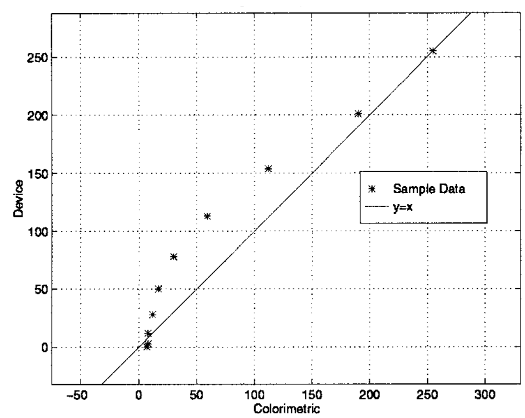

Figure 3a. Original Sample data (*), and line y=x.

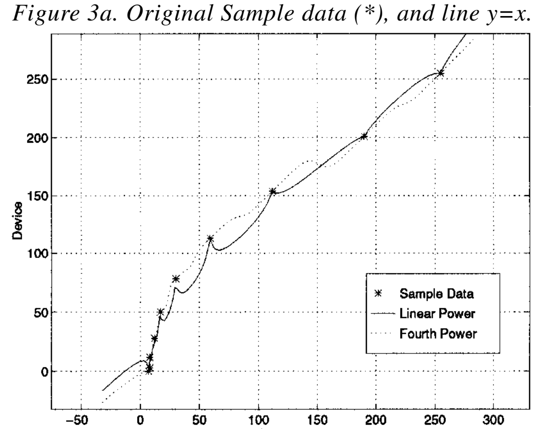

Figure 3b. Effects of using inverse power weighting with all sample points.

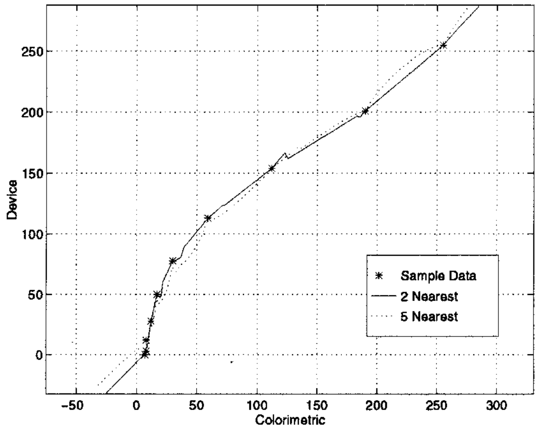

Figure 3c. Effects of using inverse linear weighting, and only some of the sample points

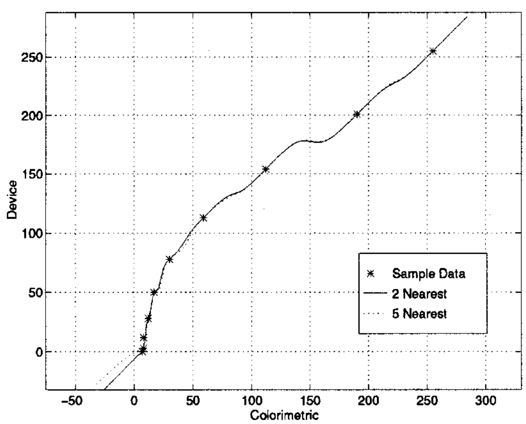

Figure 3d. Effects of using inverse cubic weighting, and only some of the sample points.

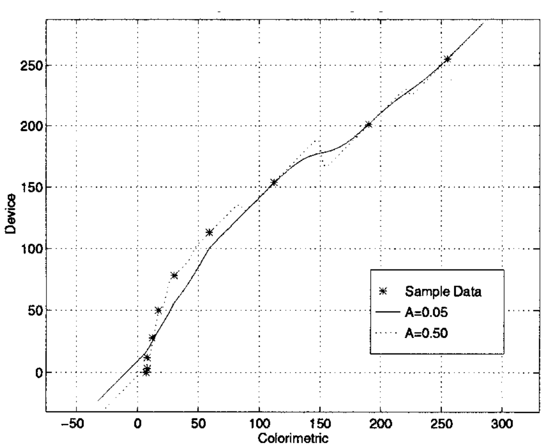

Figure 3e. Effects of using exponential weighting with different decay constants.

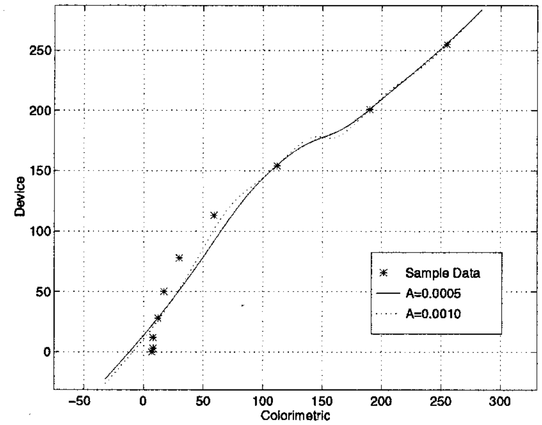

Figure 3f. Effects of using Gaussian weighting with different standard deviations.

Conclusions

There are several advantages and disadvantages to this method, as well as ways to control various properties of the LUT by changing the weighting function and the values over which the summation is carried out. The fact that sample points do not have to be regularly spaced, and that LUT node locations outside the gamut are automatically extrapolated are two advantages to this method. Two of the disadvantages are that the error vectors should be relatively small, and it is difficult to control the bounds of the calculated values. This interpolation technique offers enough flexibility for the user to investigate ways to exploit the advantages and minimize the disadvantages inherent in the method.

References

1. See for example; *PostScript Color Rendering Dictionaries*.
2. See for example; *Intercolor 3.0 Profile Specification*
3. Po-Chieh Hung, "Tetrahedral Division Technique Applied to Colorimetric Calibration for Imaging Media", *Annual Meeting IS&T*, NJ, May, 1992, pp. 419-422.
4. Po-Chieh Hung, "Colorimetric Calibration for Scanners and Media", *SPIE*, Vol. **1448**, Camera and Input Scanner System, (1991).
5. Sigfredo I. Nin, et al., "Printing CIELAB Images on a CMYK Printer Using Tri-Linear Interpolation", *SPIE Proceedings*, Vol. **1670**, 1992, pp. 316-324.
6. D. Shepard, "A two-dimensional interpolation function for irregularly-spaced data," *Proc. 1968 ACM Nat. Conf.*, pp. 517-524.
7. William J. Gordon and James A. Wilson, "Shepard's Method of Metric Interpolation to Bivariate and Multivariate Interpolation," *Mathematics of Computation*, Vol. **32**, No. 141, January 1978, pp. 253-264.
8. P. Lancaster and K. Salkauskas, "Surfaces Generated by Moving Least Squares Methods", *Mathematics of Computation*, Vol. **37**, No. 155, July 1981, pp. 141-158.
9. Pekka Laihanen, "Colour Reproduction Theory Based on the Principles of Colour Science", *Advances in Printing Science and Technology*, W. H. Banks ed., Pentech Press, London, 1988, pp. 1-36.
10. Pekka Laihanen, "Optimization of Digital Color Reproduction on the Basis of Visual Assessment of Reproduced Images", *Proceed. of SID*, Vol **30**. No. 3, 1989, pp. 183-190.