Device Characterization Using Spline Smoothing and Sequential Linear Interpolation

Ian E. Bell and William Cowan
University of Waterloo, Computer Graphics Laboratory, Waterloo, Ontario, Canada

Abstract

Color correction of images for a non-linear device uses its characterization function, often evaluated by rectilinear interpolation of a table of measurements. Sequential linear interpolation (SLI) instead allows more freely distributed grid points, but usually requires remeasure-ment to place them optimally. We smooth the measured data with a tensor-product spline before using a fast SLI look-up table: noise is reduced, and the spline curvature reveals choice SLI grid locations without remeasurement.

Introduction

Modeling the behavior of a non-linear device, like a color printer, requires a mathematical function called the *characterization function* of the device.[4] This function maps pixel values in the device input space to measured colorimetric values; for example, printer CMY pixel values to CIE XYZ measurements. Rendering images to appear the same on different display devices requires *gamut-mapping*,[11] the association of gamut values for one device with gamut values for another. A typical gamut-mapping problem, such as printing an image from a CRT, will require the characterization function of one device to be composed with the *inverse* characterization function of the other: the RGB image from the CRT is converted to XYZ space using the display characterization,[6] and then converted from XYZ space to CMY values using the inverse printer characterization.

We are interested in characterizing devices from measured data in such a way as to remove noise from the measurements. This process leaves us with a model of the device that is smoothly-varying and well-behaved. Unfortunately, no device model can ever be fast enough for the demands of color correction on million-pixel images, so as a final step we must approximate our model with a look-up table and interpolation. For these two steps, smoothing and evaluation, we have chosen as techniques tensor-product spline fitting[10, 8, 7, 2] and sequential linear interpolation.[1] The spline-fitting algorithm works with data of any dimension, and uses adaptive reparametrization and least-squares minimization to find the best fit; sequential linear interpolation (SLI) uses a small look-up table and flexible grid values to approximate the model accurately, at speeds comparable to rectilinear interpolation.[4] In this paper, we show how spline parameters can be chosen to smooth noise in a perturbed Neugebauer printer model, and we discuss methods of SLI grid selection and the error in the SLI stage.

Device Characterization

The characterization techniques in this paper are useful for any device that does not have a simple model. The most obvious examples are printers, which deviate enough from all proposed models that they must be characterized by measured tables and approximation.[11] Liquid crystal displays are a less obvious example. Although they produce color by the additive mixture of several primaries, the primaries are not of constant chromaticity,[5] and so the model is not easily invertible. The strategy outlined here, although aimed at the forward characterization function, could be adapted to invert the liquid crystal display model. For simplicity, however, we assume below that we are dealing with a CMY color printer.

Characterization of a printer is non-trivial: the subtractive combination of inks or dyes behaves non-linearly in tristimulus space.[11] Although simple models like the Neugebauer equation exist, in practice some modification is needed to make them match a printer's behavior, and even then, the model may not be appropriate for printers of a different technology.

Instead of using such a model, a more accurate method is to print and measure many colors, thus sampling the printer gamut, and use interpolation or approximation on the table of data. This approach depends on the printer gamut being continuous and moderately smooth, a reasonable assumption for any printer of good quality. With enough measurement, sampling and approximation can be very successful; the disadvantages are the cost of measurement, and the presence of measurement noise. The most popular approximation functions are rectilinear or subtetrahedral interpolation,[4] since more complex models are too costly to evaluate in real time for large images. These two schemes usually require a dense rectangular lattice of samples, and, being interpolatory, they incorporate measurement noise into the approximation.

At the Computer Graphics Laboratory, we have developed a tensor-product spline fitter to smooth noise from characterization data, yielding a well-behaved model for the printer.[8] The spline fitter reduces any high-dimensional problem to a series of one-dimensional fits, for which it iteratively adds knots and reparametrizes until a least-squares minimum is found.[10] Our experiments with an ideal printer model, discussed below, show how good spline parameters can be chosen for typical printer gamuts. Still, despite the speed of the spline model, a faster evaluation method is preferred for practical image correction. Instead of choosing conventional rectilinear interpolation, we use sequential linear interpolation,[1] a general purpose interpolation scheme with the potential for hardware implementation. The difficult decision with SLI is choosing the best grid for the domain samples: we discuss below alternatives derived from the curvature of the spline model. SLI allows us to maintain a small irregular grid of points that represent the gamut accurately, instead of the dense lattice required by conventional interpolation.

Spline Fitting

The Computer Graphics Laboratory has a spline research group that maintains a large library of C++ software. Bartels and Sreckovic developed a tensor product spline fitter for one-dimensional curves.[10] Given a set of data points, and parameters for degree and number of segments, this fitter iteratively reparametrizes and adds knots to a B-spline curve until a good least-squares fit is obtained. Since tensor product splines are a multiplicative generalization of one-dimensional splines,[7, 2] it was possible to produce a surface fitter as well, which called the curve fitter recursively. Hickey extended the fitter to support data of any dimension.[8] We are currently studying the applications of this fitter to gamut-mapping in high-dimensional reflectance spaces,[3] although we will deal only with three-dimensional color spaces in this paper.

We sample the gamut of a printer by generating a rectangular grid of points in CMY space, printing a color patch for each point, and measuring the patches to obtain tristimulus values XYZ. The spline fitter produces a model of the gamut by approximating this data, implicitly smoothing out the noise. The spline model may be evaluated directly per image pixel (in color correction, for example), or evaluated at many points to produce a look-up table for fast interpolation later. We take this latter approach, and use sequential linear interpolation as our method.

The fitter operates entirely on numerical principles, and because it has no embedded colorimetric concepts, it will fit data in any space we choose. Because it performs least-squares minimization, however, one should ensure that the Euclidean distance in the chosen space is a sensible measure of color difference. For this reason, we fit our data in CIE LAB space, which may be considered perceptually uniform.[12]

Sequential Linear Interpolation

Sequential linear interpolation has been investigated by Allebach et al., as a fast method of interpolating without being restricted to a rectangular grid of domain samples.[1] Recall that, for our printer characterization example, conventional rectilinear interpolation would have us fix values of C, values of M, and values of Y beforehand, and use their Cartesian product as the set of colors to print and measure. SLI allows a more tree-like structure: for each of a set of C values, pick a set of M values; for each CM combination, pick a set of Y values. This fairly general set of points in CMY space can be interpolated almost as quickly as the Cartesian product above;[1] the challenge is to find grid points that best sample the domain. Allebach gives an iterative remeasurement algorithm for his work with the inverse characterization function, but we can use our spline model to avoid this expense.

One obvious method is to examine the curvature of the spline function, and choose grid points clustered around regions of high curvature. Most optimization problems of this nature have fixed knot positions (grid points); this allows a basis of piece-wise polynomial functions to be constructed, and the optimal solution is simply the projection of the target function onto this basis.[9] Our problem instead allows us to move the grid points for the best solution. Even simple problems of this form do not have neat closed-form solutions. The practical approach is to *reuse* the spline fitter, with linear parameters, on the smoothed data. The fitter is designed to solve this kind of approximation problem heuristically: it gives us a piece-wise linear function, though on a rectangular grid, rather than the less regular SLI grid we want. We are currently investigating different algorithms for converting the spline knots and control vertices to the most representative SLI grid points. The speed of the spline fitter gives this strategy great potential.

Fitting Experiments

Spline Fitting without SLI

To evaluate the effectiveness of the spline fitter in smoothing characterization data, we fitted 16×16×16 uniformly-spaced CMY samples of a simple Neugebauer model having varying amounts of Gaussian noise in XYZ space. (As using the SLI look-up table can add further noise, we omitted this step from the algorithm and evaluated the spline directly.) Because of the need for small ΔE values, we converted our samples to LAB space[12] to make best use of the least-squares minimization in the fitting. The spline fitter accepts data in six columns: CMY printer inputs, and LAB color sample measurements. A few parameters are also necessary for the fitter, for each domain dimension: the degree of the curve, and the number of curve segments. We expected that small numbers of curve segments would fail to describe the data well enough, and large numbers of segments would follow the noise too closely.

The simple Neugebauer model[13] was used with XYZ values of primary, secondary, black and white colors from a DuPont 4Cast printer to give an ideal CMY to XYZ model. Gaussian noise was added in 0.5%, 2%, and 5% quantities; the percentages are fractions of the range of XYZ values for the whole gamut. It would be interesting to consider noise specifications more typical of colorimeters, but our approach proved sufficient for this simple experiment. The XYZ values were converted to LAB and processed by the spline fitter, and the resulting spline was then compared to the noise-free LAB values from the Neugebauer model.

It is worth noting that the Neugebauer equation amounts to trilinear interpolation of eight XYZ values. From our spline-based point of view, this is a linear spline with only one segment in each domain dimension, and, if we choose to fit XYZ values, the fitter does model this ideal data exactly. Once noise is added, however, fitting in XYZ space and converting to LAB gives less accurate ΔE values than fitting directly in LAB space.

Table 1. Error in spline fit to (Neugebauer model + 0.5% noise × ΔXYZ)

Degree	*Segments*	*Mean ΔE*	*Max ΔE*
1	1	6.15	33.21
	2	2.13	20.05
	3	1.19	12.80
	4	0.81	8.21
	5	0.77	6.35
2	1	1.49	16.11
	2	0.69	7.77
	3	0.56	3.73
	4	0.58	3.72
	5	0.64	4.30
3	1	0.63	6.58
	2	0.54	3.02
	3	0.57	3.78
	4	0.64	4.18
	5	0.73	4.47
4	1	0.54	3.02
	2	0.57	3.83
	3	0.64	3.96
	4	0.73	4.97
	5	0.81	6.30

In general, there is a decreasing-then-increasing trend in both *Mean ΔE* and *Max ΔE* as we increment the number of segments for each fixed degree. This is as expected, since for small numbers of segments we lose detail, and for large numbers of segments we follow noise. Similarly, for low degrees we lose detail, and for high degrees we follow noise. Note that in Table 1, for a degree 1 spline, we have a small *Mean ΔE* for 5 segments (it might have been even better for a larger number of segments), but that the approximation is better for a degree 2 spline with 3 segments, and best of all, according to this limited data set, for degree 3, 2 segments or degree 4, 1 segment.

In Table 2, the best-fitting spline is degree 3, 1 segment; in Table 3, it is degree 2, 1 segment.

The high *Max ΔE* values in Table 4 make any of the spline fits unlikely to be usable in practice; however, recall that the noise value is 5% of the difference in extreme XYZ values for the gamut, quite a high number. Any interpolation method using these noisy measurements without smoothing the data would be even less satisfactory.

Table 2. Error in spline fit to (Neugebauer model + 2% noise × ΔXYZ)

Degree	*Segments*	*Mean ΔE*	*Max ΔE*
1	1	6.38	32.50
	2	2.80	17.85
	3	2.20	13.53
	4	2.16	10.91
	5	2.34	11.55
2	1	2.25	12.94
	2	1.91	9.33
	3	2.05	9.36
	4	2.27	14.68
	5	2.54	17.77
3	1	1.88	9.34
	2	2.05	10.16
	3	2.25	15.57
	4	2.55	17.27
	5	2.90	19.18
4	1	2.04	11.23
	2	2.25	15.44
	3	2.55	17.10
	4	2.90	20.68
	5	3.24	26.23

Table 3. Error in spline fit to (Neugebauer model + 5% noise × ΔXYZ)

Degree	*Segments*	*Mean ΔE*	*Max ΔE*
1	1	7.43	32.04
	2	5.05	22.82
	3	4.90	21.39
	4	5.31	32.52
	5	5.90	43.17
2	1	4.74	18.24
	2	4.75	28.58
	3	5.27	40.65
	4	5.83	43.25
	5	6.53	57.14
3	1	4.75	29.95
	2	5.27	42.20
	3	5.81	45.63
	4	6.55	54.15
	5	7.44	63.11
4	1	5.27	41.45
	2	5.81	45.72
	3	6.56	51.35
	4	7.45	70.46
	5	8.32	85.46

Spline Fitting with SLI

To observe the effect of adding the SLI stage to the characterization process, we chose an 8×8×8 SLI grid by hand, spaced so that the points would be roughly uniform once in LAB space. Table 4 shows the somewhat surprising results contrasting the spline fit alone to the spline fit with SLI, for a 3% noise model.

A pattern of decreasing-then-increasing trends similar to Tables 1 to 3 has emerged, and it appears that a degree 3, 1 segment spline alone is best. The remarkable point is that approximating the spline with the 8×8×8

SLI look-up table has *reduced* the error in many cases; effectively, this piece-wise linear function is smoothing out some of the noisy bumps in the spline. Experiments with 0.5%, 1%, 2% and 5% noise factors showed similar results.

Conclusions

Rather than giving us hard numerical advice, the simple Neugebauer model and small subset of parameter/noise combinations give us ideas about fitting. We can make the following conclusions:

- noisy characterization data must be smoothed be fore interpolation makes sense
- tensor product splines can be used to smooth char acterization data effectively
- low-degree splines with medium numbers of segments appear to give good fits
- SLI approximations to smoothed data can actually reduce the characterization error

Table 4. Error in spline fit, and spline fit with SLI, to (Neugebauer model + 3% noise × ΔXYZ)

Degree	*Segments*	*Mean ΔE*		*Max ΔE*	
		Spline	*+SLI*	*Spline*	*+SLI*
1	1	6.64	6.64	30.92	30.92
	2	3.43	3.22	19.50	19.50
	3	3.02	2.89	15.34	15.34
	4	3.15	2.97	13.80	13.80
	5	3.48	3.22	17.71	15.56
2	1	2.97	3.01	14.44	14.60
	2	2.81	2.81	14.15	14.15
	3	3.08	2.97	16.93	17.74
	4	3.41	3.17	22.53	22.32
	5	3.82	3.23	27.55	26.94
3	1	2.79	2.80	14.94	14.94
	2	3.08	2.97	17.87	18.84
	3	3.39	3.15	23.98	23.65
	4	3.83	3.31	26.84	26.25
	5	4.36	3.62	31.00	28.68
4	1	3.08	2.97	17.90	18.92
	2	3.39	3.15	23.81	23.74
	3	3.83	3.35	26.73	25.46
	4	4.36	3.61	34.75	30.67
	5	4.87	3.72	41.87	38.89

Once further experiments are completed with a more advanced printer model and some real characterization data, we expect to conclude that

- theoretical tables can be used to give optimal fitting parameters for real characterization data, once the noise level is determined empirically

The effect of the SLI stage on accuracy needs to be studied in more depth, along with methods of determining optimal SLI grid points from a linear spline function. The algorithms used here work effectively in more than three dimensions; it is possible that even better accuracy, or at least more versatile gamut-mapping techniques, can be achieved by fitting in high-dimensional reflectance spaces.[3]

This research allows us to make better judgments about which spline parameters suit characterization data for real-world problems. We believe that spline smoothing, and fast interpolation with SLI, are an effective combination of techniques for characterizing non-linear devices, and performing gamut-mapping.

References

1. J. Allebach, J. Chang and C. Boumann, Efficient Implementation of Nonlinear Color Transformations, *Proc. 1st IS&T/SID Color Imaging Conference: Transforms and Transportability of Color*, November 1993, pp. 143-148.
2. R. Bartels, J. Beatty and B. Barsky, *An Introduction to Splines for Use in Computer Graphics and Geometric Modeling*, Morgan Kaufmann, 1987.
3. I. Bell, A Gamut-Mapping Framework Based on Splines and Reflective Image Representations, Ph.D. Thesis, Dept. of Computer Science, University of Waterloo, Canada, in preparation.
4. I. Bell and W. Cowan, Characterizing Printer Gamuts Using Tetrahedral Interpolation, *Proc. 1st IS&T/SID Color Imaging Conference: Transforms and Transportability of Color*, November 1993, pp. 108-113.
5. W. Cowan, Displays, in *Handbook of Optics*, D. Williams, Ed., *Optical Society of America*, in press.
6. W. Cowan, An Inexpensive Scheme for Calibration of a Colour Monitor in Terms of CIE Standard Coordinates, *Computer Graphics*, **17**(3), July 1983, pp. 315-321.
7. G. Farin, *Curves and Surfaces for Computer Aided Geometric Design: A Practical Guide*, 3rd ed., Academic Press, 1993.
8. B. Hickey, Fitting Data of Arbitrary Dimension with B-Splines and Applications to Colour Calibration, M. Math thesis, Dept. of Computer Science, Univ. of Wateloo, 1994.
9. P. Lancaster and K. Salkauskas, *Curve and Surface Fitting: An Introduction*, Academic Press, 1986.
10. M. Sreckovic, Adaptive Hierarchical Fitting Curves and Surfaces, M. Math Thesis, Dept. of Computer Science, University of Waterloo, 1992.
11. M. Stone, W. Cowan and J. Beatty, Color Gamut Mapping and the Printing of Digital Color Images, *ACM Transactions on Graphics*, October 1988, pp. 249-292.
12. G. Wyszecki and W. Stiles, *Color Science: Concepts and Methods, Quantitative Data and Formulae*, 2nd ed., John Wiley & Sons, 1982.
13. J. Yule, *Principles of Color Reproduction*, John Wiley & Sons, 1967.