COLOR MANAGEMENT OF PRINTERS BY REGRESSION OVER ENCLOSING NEIGHBORHOODS

Erika M. Chin[*]

University of Virginia
Dept. of Computer Science
Charlottesville, VA 22904

Eric K. Garcia[†], *Maya R. Gupta*[‡]

University of Washington
Dept. of Electrical Engineering
Seattle, WA 98195

ABSTRACT

A popular color management standard for controlling color reproduction is the ICC color profile. The core of the ICC profile is a look-up-table which maps a regular grid of device-independent colors to the printer colorspace. To estimate the look-up-table from sample input-output colors, local linear regression has been shown to work better than other methods. An open problem in local linear regression is how to define the locality or neighborhood for each of the local linear regressions. In this paper, new adaptive neighborhood definitions and regularized local linear regression are proposed to address this problem. The adaptive neighborhood definitions enclose the test sample, and are motivated by a result showing they yield bounded estimation variance. An experiment shows that both regularization and the proposed neighborhoods can lead to a significant reduction in error.

Index Terms— color management, linear regression, regularization, printers

1. INTRODUCTION

Color management is the term for actively managing how colors are processed and rendered across different devices, such as scanners, monitors, and printers. The goal is to reproduce colors accurately, or in an otherwise controlled manner. Color management of printers is difficult because printers are nonlinear devices, and the colors they produce depend on the printer hardware, the halftoning method, the ink or toner, paper type, humidity, and temperature [1, 2].

A common approach to the color management of printers is an empirical characterization of the way the device transforms the device-dependent input colors (i.e. RGB) to printed device-independent colors (i.e. CIELab). First, sample (RGB, CIELab) pairs for the device are obtained by printing RGB patches and measuring the printed patches as CIELab colors. Based on these sample pairs, a multi-dimensional look-up table (LUT) is estimated. The LUT defines directly how colors on the grid are modified, and indirectly (by interpolation of the LUT values) how non-grid colors are modified. The LUT is stored in a standardized format known as an ICC profile, which was developed by the International Color Consortium. ICC profiles are a widely-adopted standard for characterizing and correcting color changes between devices [2]. Color management modules that process ICC profiles are implemented in many common hardware and software systems [2].

It is challenging to accurately estimate the color management LUT from the sample (RGB, CIELab) pairs that are measured for a particular printer. For this estimation problem, local linear regression has been shown to work better than neural networks, polynomial regression, and splines [1], and better than color management techniques using models of ink-substrate interactions [3]. We propose using a regularized local linear regression, which is discussed in Section 2. An open issue in local linear regression is how to define an appropriate local neighborhood for each test point. In Section 3 we consider neighborhoods for local linear regression that automatically adapt to the spatial distribution of the data and do not require cross-validation. The neighborhoods investigated, which we term *enclosing neighborhoods*, enclose a test point in the convex hull of the neighborhood, ensuring that extrapolation is avoided in favor of interpolation when possible. We motivate the use of enclosing neighborhoods by showing they result in bounded variance of the regression estimate. In Section 4 and 5 we detail a color management experiment and results showing that both the proposed enclosing neighborhoods and regularizing linear regression can yield significant improvements for the color management of printers.

2. LOCAL LINEAR REGRESSION

For a given printer, denote a training set of measured output CIELab colors $\{x_i\}_{i=1,\ldots,N}$ and the corresponding input RGB colors $\{y_i\}_{i=1,\ldots,N}$, where the printer accepts input images in RGB. For a grid point g of the LUT, representing a desired output CIELab color, we would like to find the RGB color $\hat{y}$ that is an estimate of the input RGB color y that produces the desired CIELab color g.

[*] E. Chin's work was supported by the CRA-W DMP.

[†] E. K. Garcia's work was supported by an Intel/GEM Fellowship.

[‡] M. R. Gupta's work was supported in part by the United States ONR.

For each gridpoint in the LUT, local linear regression fits the least-squared error hyperplane to a set of neighbors of the gridpoint. Each output color component is estimated separately. For example, let $\hat{y}_R$ denote the red component of $\hat{y}$, and let y_{i_R} denote the red component of the ith training color y_i. Suppose $\mathcal{J}_g$ is a set of indices of training samples in a neighborhood of g. Then the local linear regression estimate is $\hat{y}_R = \hat{\beta}^T g + \hat{\beta}_0$, where

$$(\hat{\beta}, \hat{\beta}_0) = \arg \min_{(\beta, \beta_0)} \sum_{i \in \mathcal{J}_g} \left(y_{i_R} - \beta^T x_i - \beta_0 \right)^2 . \tag{1}$$

There are a number of factors to consider when choosing the neighborhood $\mathcal{J}_g$. If neighborhoods are too large, the regression is too smooth and does not accurately represent the color mapping of the printer. Conversely, if neighborhoods are too small, the locally-fitted hyperplane estimations can be ill-posed and hence inaccurate.

One way to mitigate ill-posed estimation when using a small neighborhood is by regularizing the local linear regression. A common form of regularized local linear regression is ridge regression [4], which forms a hyperplane fit as in equation (1), but the ridge regression coefficients $\hat{\beta}_{ridge}$ also minimize a penalized least-squares criteria that discourages fits with steep slopes, forcing a smoother regression. To estimate the red color component (for example), the local ridge regression coefficients solve

$$\left(\hat{\beta}_{ridge}, \hat{\beta}_0 \right) = \arg \min_{(\beta, \beta_0)} \sum_{i \in \mathcal{J}_g} \left(y_{i_R} - \beta^T x_i - \beta_0 \right)^2 + \lambda \beta^T \beta, \tag{2}$$

where the parameter λ controls the trade-off between minimizing the error and penalizing the magnitude of the coefficients. Larger λ results in lower estimation variance but higher estimation bias.

3. ENCLOSING NEIGHBORHOODS

In local learning it is common to use k nearest-neighbors for the neighborhood, where k is chosen by cross-validation. In color management, cross-validating is expensive and impractical in many workflows. Using a fixed small neighborhood size can be expected to have high estimation variance, whereas using a fixed large neighborhood size is likely to have high estimation bias. In this paper, we consider neighborhoods that do not require cross-validation and that are designed to enclose the test point in the convex hull of the neighborhood when possible.

Definition: An *enclosing neighborhood* for a test point g is a set of training points $\{x_j\}_{j \in \mathcal{J}_g}$ from which g can be formed as a convex combination. That is, there exist weights $\{w_j\}_{j \in \mathcal{J}_g}$, $w_j \in [0, 1]$, $\sum_{j \in \mathcal{J}_g} w_j = 1$ such that $g = \sum_{j \in \mathcal{J}_g} w_j x_j$.

Note that when g falls outside the convex hull of $\{x_j\}_{j \in \mathcal{J}_g}$, then an enclosing neighborhood cannot be found. Examples of two different neighborhoods that are enclosing when possible are shown in Fig. 1.

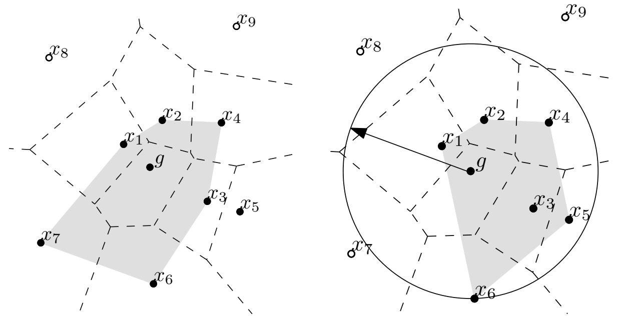

Fig. 1. Left: The *natural neighbors* of g are $\{x_i\}_{i=1,2,3,4,6,7}$. Right: The *enclosing k-NN* neighbors of g are $\{x_i\}_{i=1,\ldots,6}$.

The following proposition shows that the estimation variance using linear regression over an enclosing neighborhood results in bounded estimation variance if the underlying mapping is well-modeled by a locally linear trend with finite variance perturbations.

Theorem: *Consider a test point $g \in \mathbb{R}^p$ and training points $\{x_i\}_{i \in \mathcal{J}_g} \in \mathbb{R}^p$. Suppose g and each x_i are drawn independently and identically from a continuous distribution. Let $f(x) = a^T x + a_0 + n$ where $a \in \mathbb{R}^p$, $a_0 \in \mathbb{R}$, and $n \sim \mathcal{N}(0, \sigma^2)$. Given sample pairs $\{x_i, f(x_i)\}$, consider the linear estimate $\hat{f}(g) = \hat{\beta}^T x + \hat{\beta}_0$, where $(\hat{\beta}, \hat{\beta}_0)$ solve (1) with $y_{i_R} = f(x_i)$. Then if $\{x_j\}_{j \in \mathcal{J}_g}$ is an enclosing neighborhood, the estimation variance is bounded*:

$$E[\, (\hat{f}(g) - E[\, \hat{f}(g) \,] \,)^2 \,] \leq \sigma^2 .$$

The proof is omitted due to lack of space.

3.0.1. Natural Neighbors

Let $\mathcal{V}$ be the Voronoi tessellation of the complete set of training points and test point g. The ***natural neighbors*** of g are defined to be those training points whose Voronoi cells are adjacent to the cell containing g. Sibson's *local coordinates property* of the natural neighbors [5] can be used to prove that the natural neighbors form an enclosing neighborhood when the test point is contained in the convex hull of the training set. To the authors' knowledge, this is the first application of natural neighbors as a neighborhood for linear regression.

3.0.2. Enclosing k-NN neighborhood

Given a test point g and a set of training sample indices $\mathcal{J}_g$, we define the *distance to enclosure* $D(g, \mathcal{J}_g)$ to be the Euclidean distance between g and the convex hull of the set indexed by $\mathcal{J}_g$. That is,

$$D(g, \mathcal{J}_g) = \min_{w} \| \sum_{j \in \mathcal{J}_g} w_j x_j - g \|_2 , \tag{3}$$

where $w \in [0,1]^{|\mathcal{J}_g|}$ and $\sum_{j\in\mathcal{J}_g} w_j = 1$. Note that if g is in the convex hull of the neighborhood $\mathcal{J}_g$, then $D(g,\mathcal{J}_g) = 0$. Let $\mathcal{J}_g(k)$ be the k nearest neighbor indices of g. If there are N training sample pairs, then it must be that $D(g,\mathcal{J}_g(N))$ is the minimal achievable distance to enclosure. Let $\mathcal{K}$ be the set of all k such that $D(g,\mathcal{J}_g(k)) = D(g,\mathcal{J}_g(N))$. We define the ***enclosing k-NN neighborhood*** to be the fewest k-nearest neighbors that achieves the minimal distance to enclosure. That is, let the enclosing k-NN neighborhood be $\mathcal{J}_g(k^*)$, where

$$k^* = \min_{k\in\mathcal{K}} k. \qquad (4)$$

This neighborhood definition is related to one previously proposed by one of the authors [6], which defined the neighborhood as the smallest number of neighbors needed to enclose the test sample, and if no set of neighbors could enclose the test sample, the neighborhood was the nearest k_{max} neighbors, with k_{max} some pre-set value.

4. EXPERIMENT

A standard color management system is shown in Fig. 2: a 3D LUT characterizes the device, followed by parallel 1D LUTs on each channel that pre-linearize the device channels independently [1]. These LUTs are estimated from the set of training color pairs (see Section 2 for more details on the training color pairs). For our experiments we used the 918 sample Chromix chart for training (available from www.chromix.com), which consists of 729 uniformly spaced samples in the printer's input RGB colorspace and 189 additional neutrals and highly saturated primaries.

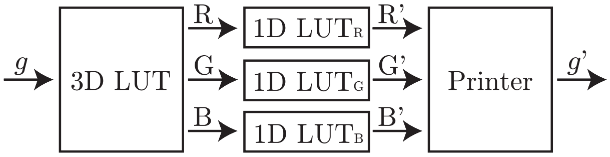

Fig. 2. Color management: A desired CIELab color g is transformed to an appropriate RGB color to input to a printer in order to print out a patch with CIELab color g' that approximates g.

The 1D LUTs enact gray-balance calibration, linearizing each channel and enforcing that neutral RGB color values (R=G=B=d) print gray patches (as measured in CIELab). The LUTs map the integers from 0 to 255 to the test colors' estimated R'G'B' colors so that if one inputs the RGB color (d,d,d) for some integer $d \in [0,255]$, the 1D LUTs will output R'G'B' values that correspond to uniformly-spaced neutral gray steps in CIELab space.

The 3D LUT characterizes the transformation from input to measured output of the printer. For the 3D LUT in our experiment, we used a 17 × 17 × 17 grid that spans the CIELab color space with $L^* \in [0,100]$ and $a^*, b^* \in [-100,100]$. Previous studies have shown that a finer sampling than this does not yield a noticeable improvement in accuracy [1]. To construct the 3D LUT, the input RGB colors must be first be adjusted since the training (RGB,CIELab) pairs were obtained without the linearization performed by the 1D LUTs. This is done by creating inverse 1D LUTs (using 1D linear interpolation on each LUT) and passing each RGB value through the three inverse 1D LUTs to produce an adjusted RGB value. Thus, when the 1D LUTs are inserted before the printer, the pairs of adjusted RGB values and measured CIELab values act as training samples for the system described by the 1D LUTs and the printer. The pairs of adjusted RGB values and measured CIELab values are used to estimate the 3D LUT.

The 1D LUT is constructed by local linear regression for each of the 255 evenly spaced CIELab gray values. For each grid point g of the 1D LUT, a neighborhood of CIELab colors is calculated. Regression for R, G, and B is done separately over g's neighborhood, yielding three locally-fitted hyperplanes for R, G, and B. Each hyperplane is then evaluated at g to estimate the R'G'B' 1D LUT output color for that grid point.

The 3D LUT is constructed in a similar fashion, with g drawn from the 17 × 17 × 17 CIELab grid, g's neighbors calculated, and the adjusted RGB values used as the outputs in the regression.

For each neighborhood method, all of the LUTs were estimated using either local linear regression or local ridge regression with ridge regularization parameter $\lambda = .1$. This λ value was chosen to minimize RGB error in a small preliminary experiment and is a robust parameter (that is, small changes in λ have little effect). The different enclosing neighborhood methods were compared to a baseline neighborhood of 15 nearest neighbors, which is a neighborhood size for this application that has worked well in other experiments [7]. Also implemented was a variation of the enclosing k-NN neighborhood that used a minimum of $k = 15$ neighbors, that is, nearest neighbors were added to the enclosing k-NN neighborhood until there were fifteen neighbors if the enclosing k-NN neighborhood had fewer than fifteen neighbors. More generally, to avoid severely ill-posed estimation, if any neighborhood consisted of fewer than four neighbors, then nearest neighbors were added to make a minimum of four neighbors.

The different neighborhoods and regression methods were tested on a Ricoh Aficio 1232C (laser engine) printer. Color measurements of the printed patches were done with a GretagMacbeth Spectrolino spectrophotometer at a 2° observer angle with D50 illumination.

To ensure that all test samples are in gamut of the printer, 729 RGB test color values were drawn randomly and uniformly from the RGB colorspace, printed, and then the printed color patches were measured in CIELab. These measured CIELab values were used as the test samples for each neighborhood and estimation method. That is, the measured CIELab values were input into the system shown in Fig. 2 where the LUT's were estimated by some combination of neighborhood method and regression method. Given a test sample, the estimated LUT is interpolated using trilinear interpolation [8]. The 729 test color samples produced by each estimated LUT were input to the printer, printed, measured in CIELab, and the ΔE^*_{94} error (a standard for measuring color management error [1]) was computed with respect to the input CIELab values.

5. RESULTS

Table 1 shows the average error, the 95th percentile error, and the maximum error for the Ricoh printer for each neighborhood definition and both linear and ridge regression. The baseline comparison is the $k = 15$ neighbors with local linear regression. In color management, small errors may not be noticeable, but even one large error may be objectionable, so the maximum error and 95th percentile errors are important metrics.

Table 1. ΔE^*_{94} Errors for Ricoh Aficio 1232C

Regression Method		*Average Error*	*95%-ile Error*	*Max Error*
Enclosing k-NN	Linear	4.27	8.47	53.83
	Ridge	3.66	7.38	20.63
Enclosing k-NN; $k \geq 15$	Linear	4.03	8.30	12.43
	Ridge	**3.45**	**6.77**	10.78
Natural Neighbors	Linear	3.74	7.55	**9.96**
	Ridge	3.69	7.10	10.23
$k = 15$ Neighbors	Linear	4.41	9.84	17.66
	Ridge	4.16	8.61	13.00

Using ridge regression with 15 neighbors eliminates 26% of the maximum error, and is seen to reduce error compared to linear regression throughout the results. Also, all of the enclosing neighborhoods result in lower average error and lower 95th percentile error than the baseline of 15 neighbors and linear regression. The lowest average error and lowest 95th percentile error is produced by the enclosing k-NN with a minimum of 15 neighbors and ridge regression: its 95th percentile error is 20% smaller than 15 neighbors with ridge regression, and 45% smaller than the baseline 15 neighbors with linear regression. The natural neighbors with linear regression achieves the lowest maximum error, 44% lower than the 15 neighbors with linear regression. The enclosing k-NN neighborhood was occasionally too small to ensure well-posed linear regression causing a high maximum error, but still has lower average and 95th percentile error than the baseline.

6. DISCUSSION

The experiments show that using an enclosing neighborhood is an effective alternative to using a fixed neighborhood size. In particular, the enclosing k-NN neighborhood with a minimum of 15 neighbors and ridge regression achieved a 39% reduction in the maximum error, and the lowest average and 95th percentile error rates, resulting in error reductions of 10% and 15%, respectively. These enclosing neighborhoods may also be useful in other applications where it is difficult to cross-validate a neighborhood size. Enclosing neighborhoods may perform better than cross-validated k-nearest neighbors for local learning because the neighborhood size varies automatically, rather than being fixed. Enclosing neighborhoods limit the estimation variance when the underlying function does have a (noisy) linear trend.

The experiments also showed that regularized linear regression can yield strong improvements for this application over linear regularization.

7. REFERENCES

[1] R. Bala, *Digital Color Handbook*. CRC Press, 2003, ch. 5: Device Characterization, pp. 269–384.

[2] B. Fraser, C. Murphy, and F. Bunting, *Real World Color Management*. Berkeley, CA: Peachpit Press, 2003.

[3] M. Shaw, G. Sharma, R. Bala, and E. N. Dalal, "Color printer characterization adjustment for different substrates," *COLOR Research and Application*, vol. 28, no. 6, pp. 454–467, December 2003.

[4] A. E. Hoerl and R. Kennard, "Ridge regression: biased estimation for nonorthogonal problems," *Technometrics*, vol. 12, pp. 55–67, 1970.

[5] R. Sibson, *Interpreting multivariate data*. John Wiley, 1981, ch. "A brief description of natural neighbour interpolation", pp. 21–36.

[6] M. R. Gupta, "Custom color enhancements," *Proc. of the IEEE Intl. Conf. on Image Processing*, pp. 968–971, 2005.

[7] M. R. Gupta and R. Bala, "Personal communication with Raja Bala," June 21 2006.

[8] H. Kang, *Color Technology for Electronic Imaging Devices*. United States of America: SPIE Press, 1997.