

Learning custom color transformations with adaptive neighborhoods

Maya R. Gupta

Eric K. Garcia

University of Washington
Department of Electrical Engineering
Box 352500
Seattle, Washington 98195-2500

Andrey Stroilov

Google

Kirkland, Washington 98033

Abstract. Custom color transformations for images or video can be learned from a small set of sample color pairs by estimating a look-up table (LUT) to describe the enhancement and storing the LUT in an International Color Consortium profile, which is a standard tool for color management. Estimating an accurate LUT from a small set of sample color pairs is challenging. Local linear and ridge regression are tested on six definitions of neighborhoods for twenty color enhancements and twenty-five color images. Excellent results were obtained with local ridge regression over proposed enclosing neighborhoods, including a variant of Sibson's natural neighbors. The evaluation of the different estimation methods for this task compared the fidelity of the learned color enhancement to the original sample color pairs and the presence of objectionable artifacts in enhanced images. These metrics show that enclosing neighborhoods are promising adaptive neighborhood definitions for local classification and regression. © 2008 SPIE and IS&T.
[DOI: 10.1117/1.2955968]

1 Introduction

Digital designers would benefit from tools that allow them to define custom color enhancements that are easy to use, share, edit, and do not require expert knowledge of color processing or statistical learning. A color enhancement is defined by a mapping from an original colorspace to another colorspace. For example, one color enhancement would be to transform all color values to pastel colors. As recently proposed,¹ one practical architecture for custom color enhancements is for a user to input a small set of example color transformations from which the complete enhancement's color mapping can be estimated. Estimating the envisioned color-space mapping from only a few samples is a difficult estimation problem, but a solution that uses linear regression over a local neighborhood was shown to give promising results.² We expand on that work in this paper.

Paper 07191R received Oct. 2, 2007; revised manuscript received Apr. 9, 2008; accepted for publication Apr. 9, 2008; published online Jul. 28, 2008. This paper is based on the revision of a paper presented at the SPIE conference on Computational Imaging, San Jose, California, January 2005. The paper presented there appears (unrefereed) in SPIE Proceedings Vol. 5674.

1017-9909/2008/17(3)/033005/9/\$25.00 © 2008 SPIE and IS&T.

Three examples of user-defined sample color pairs are shown in Figs. 1–3. In Fig. 1, the sample pairs are shown as vectors in the CIELab color space; each vector connects one of the 24 color patches of the standard Gretag Macbeth color chart photographed under a specific illuminant to the same color patch photographed under a different illuminant; more details are available in Section 6.1. An example of an image enhanced with a look-up-table (LUT) estimated from the cloudy-to-sunset sample pairs is shown in Fig. 2. Another example of color sample pairs is given in Fig. 3, which shows how 16 colors would be displayed using the Cinecolor film process; results with the Cinecolor enhancement are shown in Fig. 5. The goal for estimating a color enhancement from a small set of sample color pairs is to accurately capture the user's envisioned color enhancement. More practically, we consider two metrics for the estimation: how well the estimated color enhancement accurately reproduces the user's sample pairs and whether images enhanced with the estimated mapping are free of objectionable artifacts.

Once estimated, the custom color enhancement can be stored as an International Color Consortium (ICC) profile. ICC profiles are the most widely adopted standard for characterizing and correcting color changes between devices, such as printers and monitors.^{3–5} The core of an ICC profile is a multidimensional LUT that spans the color space, such as a $17 \times 17 \times 17$ grid of points. The LUT defines how the colors on the grid are modified. The transformation of non-grid input colors are interpolated from the colors in the LUT. Color management modules that process ICC profiles are already implemented in many common hardware and software systems.⁴ Though developed and used to manage color between devices, ICC profiles provide a standardized and flexible architecture for defining any color transformation.

ICC profiles were standardized for color management. To estimate a LUT for color management of a device, one can expect to have on the order of a few hundred color sample pairs that span the color space and an underlying color transformation that is generally monotonic, though

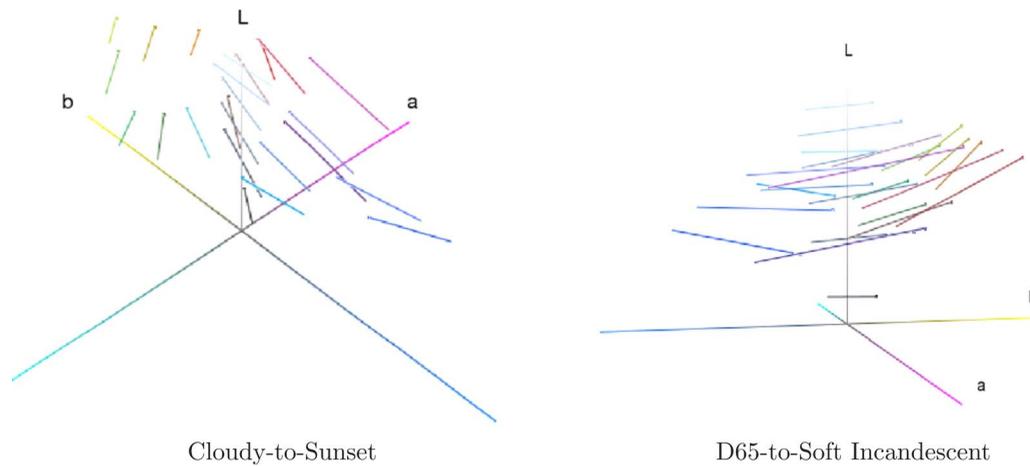


Fig. 1 Two examples are shown of sample pairs that describe a change of illuminant. Each sample pair is displayed as a vector in CIELab space that connects an input color to its enhanced color. The colors correspond to the colors of the standard 24 sample Gretag Macbeth color chart photographed under different illuminations.

nonlinear. In contrast, to estimate a custom color enhancement LUT, it must be possible to estimate the LUT based on the order of 20 color sample pairs that do not necessarily span the color space. Additionally, the custom enhancement to be learned may be nonmonotonic as well as nonlinear. A first exploration of learning ICC profiles from sample pairs¹ showed that there were estimation trade-offs between over-smoothing the color enhancement and creating objectionable artifacts in enhanced images, such as unwanted false contours.

Building on previous work,² new neighborhood definitions are proposed for local linear and ridge regression in order to estimate custom color enhancements from a small set of color sample pairs. Ridge regression is a penalized form of linear regression that can reduce estimation variance. Both linear and ridge regression are flexible estimation methods when applied to local neighborhoods. We investigate adaptive neighborhood methods that attempt to enclose the points being estimated within a convex hull of neighborhood training samples. Extensive experimentation establishes the effectiveness of the new methods.

First, related research on color enhancements is reviewed in Section 2. Then, different estimation approaches are considered for estimating the ICC profile from the given color sample pairs in Section 3. A local learning ap-

proach gives the user both flexibility and control, but requires a local neighborhood to be defined. In Section 4, the literature on neighborhood definitions is reviewed, and we introduce the term *enclosing neighborhood*. New enclosing neighborhoods are proposed in Section 5. Experimental details are given in Section 6, followed by results in Section 7. The paper ends with a discussion on the future usage of ICC profiles and how enclosing neighborhoods may be advantageous for general learning problems.

2 Related Research on Color Enhancements

Related to this work, Trussell *et al.* proposed using a LUT to help consumers correct color by fitting a low-order polynomial to a small number of user-defined samples.⁶ Gatta *et al.*⁷ and Rizzi *et al.*⁸ proposed using spatially local LUT's to approximate complicated color and image enhancements such as Retinex. For that application, no estimation is needed. Other researchers have worked on enhancing images using statistical learning techniques; for example, Hertzmann *et al.* showed that learning could be used to enhance images by image analogies.⁹ Their work uses a pair of input images to define a transformation. Using input image pairs allows them to create spatial enhancements as well as color enhancements. Other researchers have developed methods to create custom color image enhancements by transforming the color palette of an original image based on the color palette of a single reference image.^{10,11}



Fig. 2 Photograph (a) was taken under an unknown illuminant, possibly a cloudy morning. Photograph (b) is a color-enhanced version of the left photograph, using a cloudy-to-sunset enhancement estimated from the cloudy-to-sunset sample pairs shown in Fig. 1. (Color online only.)

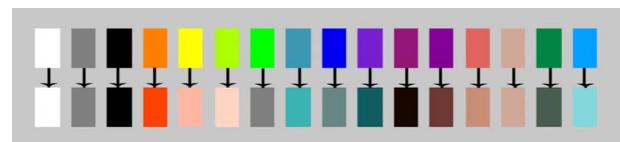


Fig. 3 Input and enhanced sample colors are shown that map original colors to how they would approximately appear in a Cinecolor film, according to the American Widescreen Museum Web site [34]. The Cinecolor color film process was used in the 1930s. Cinecolor used only two film colors (roughly described as red and cyan). (Color online only.)

3 Approaches to Estimation

Let $\mathbf{g} \in Lab$ be a gridpoint of a 3-D LUT of an ICC profile. Let $\hat{\mathbf{y}} = \{\hat{L}, \hat{a}, \hat{b}\}$ be the estimated enhanced CIELab color corresponding to \mathbf{g} . Let $\{\mathbf{x}_i, \mathbf{y}_i\}$ for $i=1, \dots, n$ be the user-defined sample color pairs, where $\mathbf{x}_i \in Lab$ and $\mathbf{y}_i \in Lab$. The components of \mathbf{y}_i are denoted $\mathbf{y}_i = \{L_i, a_i, b_i\}$. If the user-defined color sample pairs are not originally in CIELab, then it is assumed that they are transformed to CIELab before processing. The problem is to estimate $\hat{\mathbf{y}}$ for every gridpoint \mathbf{g} of the LUT based on the given sample pairs $\{\mathbf{x}_i, \mathbf{y}_i\}$.

A key issue behind the estimation is that the user's small set of n given color samples may not cover the full color space; thus, many gridpoints fall outside the convex hull spanned by the given color samples. It was shown in earlier work¹ that interpolative methods, such as tetrahedral linear interpolation or LIME,¹² clip colors outside the convex hull of the given color samples to the gamut defined by that convex hull. Instead, extrapolative methods, such as local linear regression,^{13,14} must be used, at least for the outer gridpoints. Local linear regression has also been shown to work better than other regression methods for estimating LUTs for color management, including a neural net, polynomial regression, and splines.¹⁵

Local linear regression fits the least-squared error hyperplane to some local neighborhood of each gridpoint of the LUT. For a gridpoint \mathbf{g} ,

$$\hat{L} = l^T \mathbf{g} + l_0$$

$$\hat{a} = \alpha^T \mathbf{g} + \alpha_0 \quad (1)$$

$$\hat{b} = \beta^T \mathbf{g} + \beta_0,$$

where l, l_0 are regression coefficients of a least-squares hyperplane fit to the neighborhood samples $\{\mathbf{x}_i, L_i\}$, α, α_0 are the vector of regression coefficients of a least-squares hyperplane fit to the neighborhood samples $\{\mathbf{x}_i, a_i\}$, and β, β_0 are the vector of regression coefficients of a least-squares hyperplane fit to the neighborhood samples $\{\mathbf{x}_i, b_i\}$.

Using too large a neighborhood can result in extrapolations that are too smooth and do not capture the sense of the desired color transformation. Fitting a plane to a neighborhood that is too small results in a plane with a slope that is too steep, causing extrapolated colors to be grossly incorrect or clipped at the boundary of the display colorspace, which results in images with objectionable flat regions of clipped color. A related danger occurs when bright white is mapped to a nonwhite color, which can have the unintended effect that specular reflections and highlights in the image become colored in a manner inconsistent with the perceived whitepoint for the image.¹ Though affecting few pixels, the eye is very sensitive to such deviations from the perceived neutral axis⁴ and such colored highlights appear unnatural.

In mock workflow simulations with custom color enhancements, the ability to fine-tune a particular region of the color space was important to designers. Compared to global regression surface-fitting techniques, such as neural networks, local learning methods enable designers to lo-

cally edit the colors of a transform while minimizing changes to other parts of the colorspace. Another concern with neural networks is overfitting the surface to the training set. In related work by Trussell *et al.*,⁶ color corrections were implemented by fitting low-order global polynomials to a few user supplied color sample pairs. Local learning allows more flexibility to define a more complicated or precise color enhancement.

Ridge regression is used to stabilize the estimation.^{13,16} Ridge regression forms a hyperplane fit, as in Eq. (1), but the ridge regression coefficients $\bar{\beta}$ minimize a penalized least-squares criteria, which discourages fits with steep slopes. For example, for the luminance plane the ridge regression coefficients solve

$$l^{\text{ridge}} = \arg \min_{\beta, \beta_0} \left(\sum_{i=1}^n L_i - \beta^T \mathbf{g} - \beta_0 \right)^2 + \lambda \sum_{d=1}^3 \beta_d^2. \quad (2)$$

The parameter λ controls the trade-off between minimizing the error and penalizing the coefficients, with larger λ leading to flatter sloped hyperplane estimates.

Higher-order polynomial fits or spline fits are also possible solutions to estimating the LUT. These smoother fits would have more continuous derivatives. Because colors are perceptually indistinguishable if they are close enough, a point discontinuity is unlikely to be noticed, and we question the importance of continuous derivatives for this application. However, the slower variation of a higher-order polynomial fit could be an advantage over the disjoint fit formed by local linear regression. The disadvantages of such higher-order fits are an increased difficulty for the designer to fine-tune the enhancement, and potentially wild nonlinear estimates for colors that fall outside the convex hull of the training samples.

4 Research into Neighborhoods

Local learning, such as local linear regression, requires a definition of a local neighborhood for each test point. For this application, each of the unknown gridpoints in the LUT will, in turn, be considered a test point. A common neighborhood choice is to use the k nearest neighbors as defined by Euclidean distance, or as defined by some locally adapted distance.¹⁷⁻²⁴ In practice, it is common to choose k by cross-validation.¹³

The challenge is to do well on average with a small set of training data. On the basis of experiments reported in Ref. 2 we hypothesized that using a neighborhood that encloses a test point in the convex hull of the neighborhood points when possible will work well on average. Define an enclosing neighborhood to be any set \mathcal{J} of indices such that the gridpoint $\mathbf{g} = \sum_{j \in \mathcal{J}} w_j \mathbf{x}_j$ where $\mathbf{w} \in [0, 1]^{|\mathcal{J}|}$ is a weight vector over the set \mathcal{J} subject to the constraint $\sum_{j \in \mathcal{J}} w_j = 1$. The neighborhood defined by Sibson²⁵ as the "natural neighbors" is an enclosing neighborhood (when possible). Although Sibson proposed natural neighbors with a specific generalized linear interpolation formula (called natural-neighbors interpolation), we consider them to be a promising neighborhood definition for more general learning tasks. To define Sibson's natural neighbors, let \mathcal{V} be the Voronoi tessellation of the complete set of training points

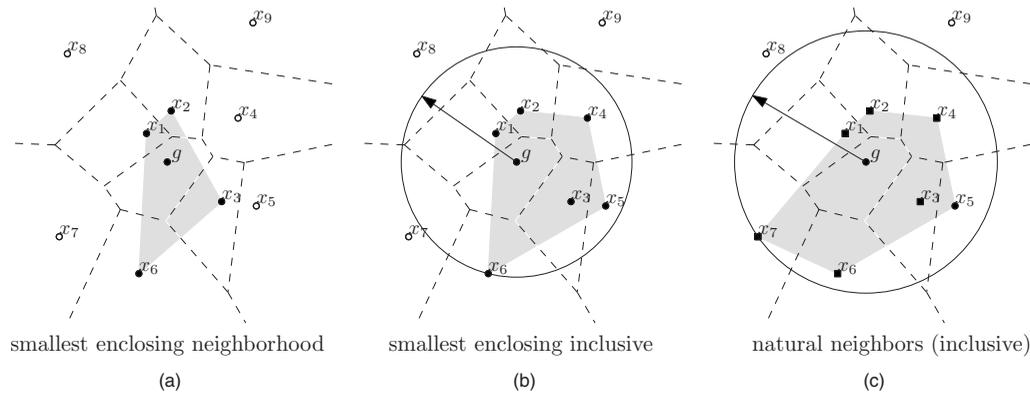


Fig. 4 The four enclosing neighborhoods are compared. Image (a) shows the smallest enclosing neighborhood. Here, the point x_6 was needed to make it an enclosing neighborhood. Points x_4 and x_5 did not decrease the distance to enclosure after including x_3 and are thus excluded. Image (b) shows the smallest enclosing inclusive neighborhood, which includes x_4 and x_5 , as they are nearer than the furthest neighbor. Image (c) shows the natural neighbors marked as squares ($x_1, x_2, x_3, x_4, x_6, x_7$) and the natural neighbors inclusive neighborhood, which additionally includes x_5 .

$\{x_i\}$ 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 .

Natural neighbors has been reported to be an accurate method for linear interpolation in two and three dimensions.²⁵ Unfortunately, to find the natural neighbors the entire Voronoi tessellation must be computed, which is computationally problematic as the dimension rises.^{26,27} For 3-D color problems, such as the application considered in this work, natural neighbors are a feasible solution.

Sibson's local coordinates property of the natural neighbors²⁵ proves that the natural neighbors form an enclosing neighborhood when possible. The local coordinates property establishes that, for a point g inside the convex hull of the entire training set, a weight vector $\lambda(g)$ exists given g 's natural neighbors \mathcal{N} such that

$$\sum_{i \in \mathcal{N}} \lambda_i(g) x_i = g,$$

where the weight vector λ is non-negative and its components sum to one.

Other neighborhoods have been defined based on spatial relationships between sample points.^{28,29} For example, defining a test point's neighbors as all Gabriel neighbors from the training set has been investigated^{26,24} (pg. 90). Decision trees can be viewed as an adaptive neighborhood definition where the neighborhood is chosen to be (typically) a refined hyperrectangle or half-space that minimizes some empirical risk. A variation uses the decision tree to restrict the k nearest-neighbor search to only those samples in the same branch of a learned decision tree.³⁰

5 Enclosing Neighborhood Definitions

In this section, new enclosing neighborhood definitions are proposed. It is useful to define a distance to the enclosure of the neighborhood, which is the Euclidean distance between a test point and the convex hull of a set of training points. Given a gridpoint g , consider a set of neighborhood indices \mathcal{J} . Let the distance to the enclosure of the neighborhood \mathcal{J} about g be denoted $D(g, \mathcal{J})$, and defined

$$D(g, \mathcal{J}) = \min_w \left\| \sum_{j \in \mathcal{J}} w_j x_j - g \right\|_2, \quad (3)$$

where $w \in [0, 1]^{|\mathcal{J}|}$ is a weight vector over the \mathcal{J} subject to the constraint

$$\sum_{j \in \mathcal{J}} w_j = 1. \quad (4)$$

Then, a neighborhood can be proposed that is the smallest enclosing neighborhood.

Smallest enclosing neighborhood. Reorder the samples by distance from the gridpoint g so that x_j is the j 'th nearest neighbor to g . Then, consider each x_j in turn for $j = 1, \dots, n$. After considering k neighbors, denote the set of neighborhood indices \mathcal{J}_k . Then the j 'th neighbor is added to the neighborhood if it reduces the distance to the enclosure of the neighborhood D , that is, if

$$D(g, \{\mathcal{J}_k, j\}) \leq D(g, \mathcal{J}_k). \quad (5)$$

The smallest enclosing neighborhood is defined by the set of neighbor indices \mathcal{J}_n . An example is shown in Fig. 4. For a given set of samples and a gridpoint g , one can solve for the neighborhood using quadratic programming.

Smallest enclosing inclusive neighborhood. Given the set of smallest enclosing neighborhood indices \mathcal{J}_n , define the smallest enclosing inclusive neighborhood to include x_j if

$$\|g - x_j\|_2 \leq \max_{i \in \mathcal{J}_n} \|g - x_i\|_2.$$

Natural neighbors was defined in Section 4. Here, a variant is proposed that may lead to lower estimation variance:

Natural neighbors inclusive neighborhood. Let \mathcal{N} be the set of indices of the natural neighbors for a given gridpoint g . The natural neighbors inclusive neighborhood includes any sample point x_j that is closer to the gridpoint than the fur-

Table 1 Errors for the Cinecolor transform. Errors have been rounded for display.

	Mean ΔE			Median ΔE		
	$\lambda=0$	$\lambda=0.5$	$\lambda=1$	$\lambda=0$	$\lambda=0.5$	$\lambda=1$
Four neighbors	29	8	10	9	5	8
All-but-one neighbors	13	13	14	11	12	13
Natural neighbors	11	11	12	10	11	11
Natural neighbors inclusive	12	12	13	11	12	13
Smallest enclosing neighborhood	22	10	11	15	7	9
Smallest enclosing inclusive neighborhood	17	10	11	14	8	8

the natural neighbor. That is, \mathbf{x}_j is included in the neighborhood if

$$\|\mathbf{g} - \mathbf{x}_j\|_2 \leq \max_{\mathbf{x}_i \in \mathcal{N}} \|\mathbf{g} - \mathbf{x}_i\|_2.$$

Figure 4 shows a comparison of the different neighborhoods.

For the experiments, the following additional neighborhood definitions are compared:

Four nearest neighbors: This neighborhood consists of the four sample points closest to the gridpoint \mathbf{g} , which is the minimum needed to solve for linear regression coefficients (assuming all the sample points are in general position).

All-but-one neighbors. Smoother interpolations should be achieved by regressing over larger neighborhoods. The all-but-one neighbors include all the training samples except the furthest training sample from the gridpoint \mathbf{g} .

6 Experiments

Neighborhoods are formed as per the descriptions in Section 5, where “nearest neighbors” are computed with respect to Euclidean distance in CIELab, and any distance or norm is computed in CIELab color space. If a neighborhood consisted of less than four neighbors, then nearest neighbors were added to make a minimum of four neighbors (to ensure stable regression). For each neighborhood method, a 3-D LUT is formed using local ridge regression with smoothing parameters $\lambda \in \{0, .5, 1\}$. Ridge regression with $\lambda=0$ is equivalent to linear regression. The set of gridpoints \mathcal{G} formed a $21 \times 21 \times 21$ LUT in CIELab color space with gridpoints spaced five units apart for the L channel and spaced 10 units apart for the a and b channels. All colors were either originally described in CIELab or originally described as RGB samples and transformed to CIELab using the standard sRGB-to-CIELab transformation with the default white point of D65.

For each of the different neighborhood definitions, a LUT was generated by estimating the corresponding output color $\hat{\mathbf{y}}$ using local ridge regression for each gridpoint

$\mathbf{g} \in \mathcal{G}$. Then, each of the different LUTs was used to enhance the test set of images. The ICC standard does not constrain which estimation method is used for the LUT interpolation; the experiments used trilinear interpolation. Trilinear interpolation is a standard method for interpolating profiles³¹ and is a 3-D version of the common bilinear interpolation. Code for trilinear interpolation is available in Ref. 32 and is implemented by the Matlab function *interp3*.³³ Recently, it has been shown that trilinear interpolation weights the vertices of the LUT cell with weights that have the maximum entropy out of all solutions that satisfy the linear interpolation equations.¹²

The test set of images included 24 Kodak images from the Kodak Photo CD PCD0992 and the 918 sample Chromix color chart (available at www.chromix.com). The Kodak images have been released by Kodak to the public domain and are 24 bit RGB color natural images of a variety of scenes with 768×512 pixels. We converted the images to CIELab using the default sRGB-to-CIELab formula. For each estimation method, the image quality of enhanced test images was compared and the error on the training sample pairs was calculated to measure the fidelity of the enhancement to the given color sample pairs. The experimental data and images are available at idl.ee.washington.edu/projects.php.

For each neighborhood definition, the estimated LUT are compared on their ability to accurately recreate the original sample pairs $\{\mathbf{x}_i, \mathbf{y}_i\}$. Each estimated LUT is used with each \mathbf{x}_i to obtain the estimated $\hat{\mathbf{y}}_i$, which is then compared to the true training sample output color \mathbf{y}_i . The mean and median ΔE CIELab errors for each estimated LUT for the given training samples are reported in Table 1. The ΔE numbers are calculated as per the original CIE formula (Ref. 5, pg. 80), where ΔE is Euclidean distance in the CIELab space. These errors are based on testing on the training samples and thus reward overfit solutions. However, this still is a useful metric of the fidelity of the enhancement to the given color sample pairs. Furthermore, the overfitting potential is limited, as some smoothing will have taken place when estimating the LUT gridpoints, and smoothing occurs again when the original sample points are

interpolated based on the cell vertices of the LUT. Furthermore, any overfitting that does occur is likely to lead to false contours in the enhanced test images and would be counted negatively under image quality.

6.1 Experimental Transforms

Twenty custom enhancements were used for testing. Twelve transforms simulated illuminant changes as a digital designer (without access to a spectrophotometer) might do. For these, the standard 24 sample Gretag Macbeth color chart was photographed under four different illumination conditions: D65, Seattle cloudy, Seattle sunset, and a soft incandescent lightbulb. Twelve different pairings of input and output illumination samples defined 12 of the transforms. Photographs were taken with a Sony DSC-F828 8 MegaPixel camera. For each picture, the color chart and a standard photographer's gray card were set up on an easel. The camera was set to manual mode with the white balance set to "daylight." The metering mode was set to "spot," and the angle of the board was adjusted until all the readings on the gray card were equal. Then the photo was taken. The D65 illuminant photograph was taken under a Gretag Macbeth SolSource D65 filtered lamp. For each of the four illuminant conditions, the pixels of each of the 24 color squares was averaged, forming a data set of 24 sRGB values for each illumination condition. The sRGB values were converted to CIELab values by the standard formula, using the default D65 white point to match the camera's "daylight" white point.

The other eight transforms were designed in consultation with or by professional digital designers. The thirteenth transform maps 16 colors to approximately how they would appear if rendered by the two-color Cinecolor film process, as shown in Fig. 3. The fourteenth transform maps input sample colors to the closest color in a forest palette (14 sample pairs); the fifteenth maps sample colors to the gamut of a product line of ceramic tiles (27 sample pairs). The sixteenth transform only saturates midtones (15 sample pairs). The seventeenth transform only brightens yellows (28 samples). The eighteenth transform tints all the high-lights rose-colored (22 samples). The nineteenth transforms various small parts of the color space to bright red (42 samples). The twentieth transform maps nongrays to the purple and gold school colors of the University of Washington (111 samples).

In practice, a user would apply a transform and then might change, add, or delete sample pairs to edit the enhancement. To compare the quality of the different estimations, the enhancements were not edited in these experiments.

7 Results

The results show that the proposed architecture can be used to effectively learn the tested set of custom enhancements. Example results are shown in Figs. 5 and 6. Using ridge regression over any of the enclosing neighborhoods produced consistently good results. In Fig. 5, the Cinecolor transformation is shown for different neighborhood choices (rows) and with linear or ridge regression (columns). The images in Row (2) of Fig. 5 use four nearest neighbors. The ridge regression on the right (c) successfully dampens the wild extrapolations seen on the left (b), but there are still

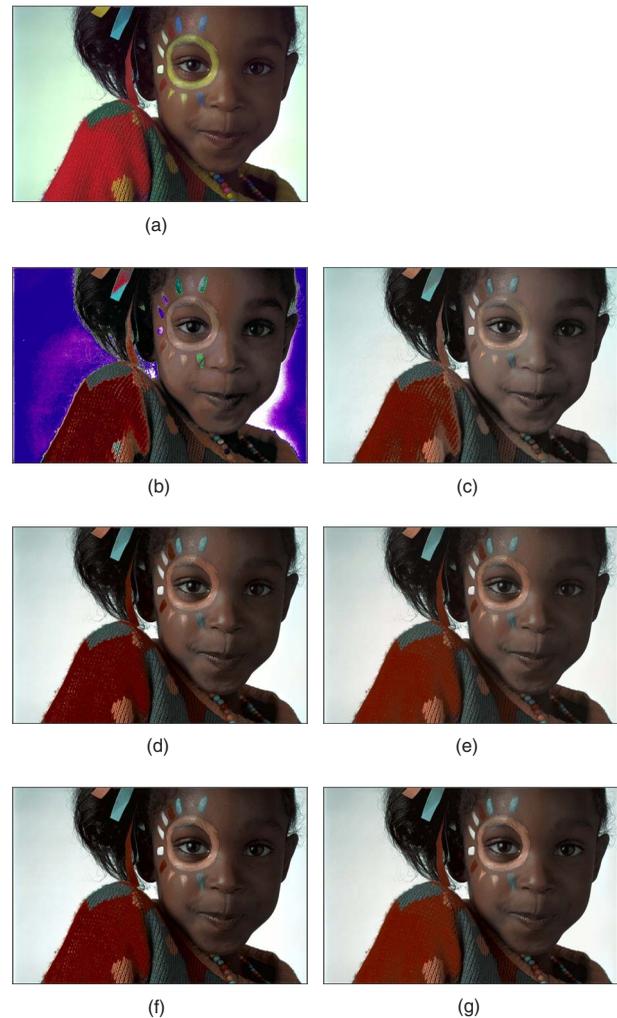


Fig. 5 The original image [shown in Row (1)] and the image transformed by the different Cinecolor LUT's. Row (2): four nearest neighbors: (b) local linear regression, and (c) local ridge regression $\lambda=0.5$. Row (3): smallest enclosing neighborhood: (d) local linear regression, and (e) local ridge regression $\lambda=0.5$. Row (4): smallest enclosing inclusive neighborhood: (f) local linear regression, and (g) local ridge regression $\lambda=0.5$. (Color online only).

unacceptable image quality problems on the red sweater. Objectionable artifacts occur often in the test set when the four nearest neighbors are used with any of the ridge regression parameter settings.

In Rows (3) and (4) of Fig. 5, images are shown for the smallest enclosing neighborhood and the smallest enclosing inclusive neighborhood, respectively. These two neighborhoods yield small estimation error on the original color sample pairs, as reported in Table 2. However, they occasionally suffer image quality problems when used with linear regression. The objectionable artifacts disappear when ridge regression is used. There were only two instances of objectionable false contouring in the test set that were not removed by using ridge regression; both instances occurred with the Cinecolor transform. Additionally, ridge regression did not remove mildly objectionable color distortion on the Kodak hat test image when transformed with D65-to-Soft-Incandescent.

The natural neighbors neighborhood generally achieved

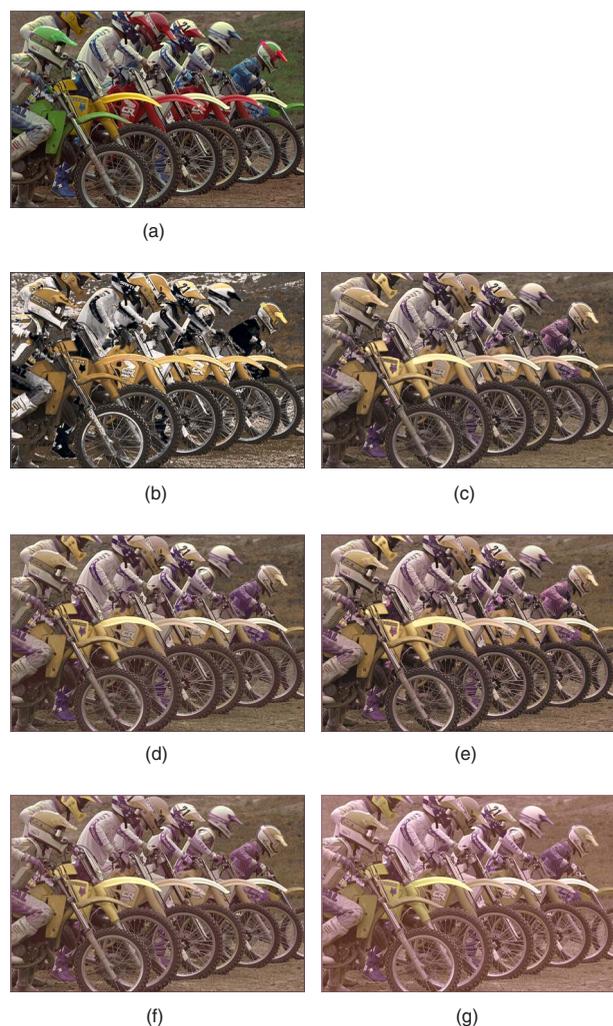


Fig. 6 The original image [shown in Row (1)] and the image transformed by the different purple-and-gold LUTs that were estimated with local ridge regression with $\lambda=0.5$. Row (2): (b) four nearest neighbors, and (c) smallest enclosing neighborhood. Row (3): (d) smallest enclosing inclusive neighborhood, and (e) natural neighbors. Row (4): (f) natural neighbors inclusive, and (g) all-but-one neighbors.

high image quality, but on occasion led to objectionable false contouring. When used with ridge regression ($\lambda=0.5$), no objectionable false contouring was seen with the natural neighbors enhanced images. However, the Kodak test image of brightly colored hats transformed with the D65-to-Soft-Incandescent enhancement using natural neighbors showed mildly objectionable color distortion on one of the hats, even with ridge regression. The larger neighborhood size of the natural neighbors inclusive neighborhood did not exhibit objectionable artifacts in the test set, even when used with linear regression.

Using the all-but-one neighborhood never resulted in objectionable image artifacts, but failed to accurately capture some of the more nonlinear color-space transformations. For example, the results of the purple-and-gold enhancement for each neighborhood with ridge regression ($\lambda=0.5$) is shown in Fig. 6. The image (g) is the all-but-one neighborhood, and the colors appear oversmoothed compared to the enclosing neighborhoods. In particular, because the neutral color axis was mapped to itself in the reference samples, the bikers' shirts should appear white. The larger neighborhoods smooth the colors such that the shirts do not appear white. The oversmoothing of the colors is reflected in Table 3, which gives the average error on the original color sample pairs for the purple-and-gold transform. For $\lambda=0.5$, the all-but-one neighbors has twice as much error as the smallest enclosing neighborhood for reproducing the original sample color pairs. Four neighbors achieves the lowest error on the original sample pairs, but the image shows that four nearest neighbors fails to produce reasonable image quality when faced with the larger test set of colors that appear in the image.

Similarly, the four nearest neighbors performs well in terms of fidelity to the original color sample pairs for the Cinecolor transform for $\lambda=0.5$, as shown in Table 1, but the four neighbor images in Row (2) of Fig. 5 contain objectionable artifacts (here, the appearance of gray mold on the red sweater). Table 2 shows fidelity to the given enhancement color sample pairs averaged over the 20 different transforms. At $\lambda=0.5$, the smallest enclosing neighborhood

Table 2 The mean and median errors averaged over all 20 transforms. Errors have been rounded for display.

	Mean ΔE			Median ΔE		
	$\lambda=0$	$\lambda=0.5$	$\lambda=1$	$\lambda=0$	$\lambda=0.5$	$\lambda=1$
Four neighbors	14	8	9	8	6	7
All-but-one neighbors	12	12	12	10	10	11
Natural neighbors	10	10	11	8	9	9
Natural neighbors inclusive	10	10	11	9	9	9
Smallest enclosing neighborhood	10	7	8	7	6	7
Smallest enclosing inclusive neighborhood	9	8	8	7	7	7

Table 3 Errors for the purple-and-gold transform. Errors have been rounded for display.

	Mean ΔE			Median ΔE		
	$\lambda=0$	$\lambda=0.5$	$\lambda=1$	$\lambda=0$	$\lambda=0.5$	$\lambda=1$
Four neighbors	36	6	6	31	3	3
All-but-one neighbors	16	16	16	16	16	16
Natural neighbors	11	10	10	9	8	9
Natural neighbors inclusive	14	14	14	13	13	14
Smallest enclosing neighborhood	15	7	7	14	5	5
Smallest enclosing inclusive neighborhood	15	11	11	13	10	10

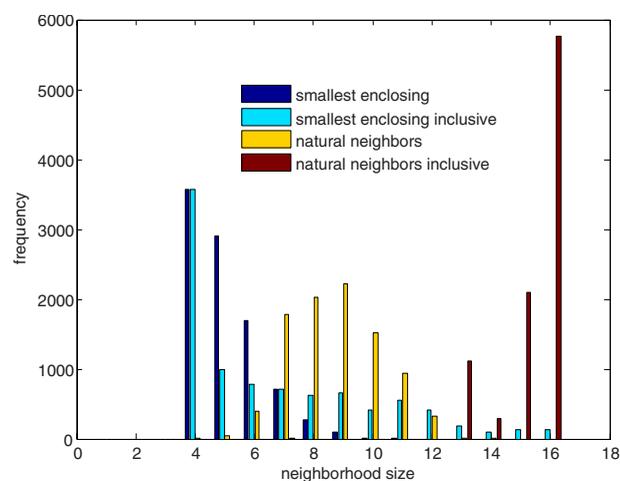
performs best by the total mean error metric and is tied for best by the total median error metric. Overall, the combined goals of fidelity and image quality are best satisfied for the test set by the smallest enclosing neighborhood with ridge regression at $\lambda=0.5$. Not shown in the figures or tables are the ridge regression results for smoothing parameter $\lambda=1$. Those results are generally very similar to the results with $\lambda=0.5$. CIELab errors are difficult to interpret quantitatively, but larger errors will generally correspond to greater perceptual errors.

Histograms of neighborhood size for two example enhancements are shown in Fig. 7. On average, the smallest enclosing inclusive is smaller than the natural neighbors neighborhood. The natural neighbors inclusive neighborhood is often quite large relative to the other enclosing neighborhoods and that explains the relatively color-smoothed images produced with this neighborhood.

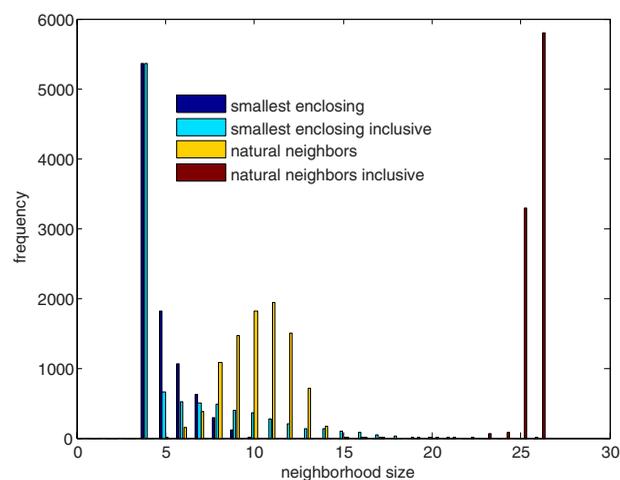
8 Discussion

In earlier work, we proposed the use of ICC profiles to capture custom color enhancements based on a small set of color sample pairs.¹ In this work, we have established that ridge regression over enclosing neighborhoods will produce reasonably accurate transforms with consistently good image quality and without the need for training (or cross-validation). In particular, new definitions of enclosing neighborhoods provide the most faithful color enhancements while rarely producing objectionable artifacts for enhancements where a smooth enhanced image could be expected.

ICC profiles are a flexible standard that has penetrated design software and hardware. This makes them an ideal vehicle for color processing far beyond the original intent of color management. In this work, we focused on how to learn an enhancement from an arbitrary small set of color sample pairs. As suggested in work by Gatta *et al.*⁷ and Rizzi *et al.*⁸ complex color transforming functions or programs can be approximated and implemented as ICC profiles. An open problem with color-space transformations of colors described as 3-D vectors is that they do not take into account semantic information about image content and thus



(a)



(b)

Fig. 7 Histograms show the frequency of each size neighborhood over the 9261 gridpoints for (a) the Cinecolor transform, and (b) the cloudy-to-sunset transform.

face the problem of metamers. For example, regions of blue sky, blue jeans, and blue water may all have the same color value, but their physical nature is different and how they reflect incoming light spectra is different.

Another limitation of ICC profiles is the lack of spatial dependence. An ICC profile can be applied to segmented parts of an image, but the color transform itself ignores spatial information. This limits the ability to implement textural or spatial color enhancements. Color management scientists know that spatial effects on color perception are important and have proposed color appearance models.⁵ We ponder what a general spatial color transforming architecture would be that would have the flexibility and simplicity of the ICC profiles.

Acknowledgments

The authors thank Steve Upton, Lian Chang, Al Luckow, Matthew Cassarino, and Jayson Bowen for helpful discussions.

References

- M. R. Gupta, S. Upton, and J. Bowen, "Simulating the effect of illumination using color transformation," *Proc. SPIE* **5674**, 248–258 (2005).
- M. R. Gupta, "Custom color enhancements," *Proc. of IEEE Intl. Conf. on Image Processing*, pp. 968–971 (2005).
- ICC webpage, (2005), www.color.org/profile.html
- B. Fraser, C. Murphy, and F. Bunting, *Real World Color Management*, Peachpit Press, Berkeley, CA (2003).
- M. D. Fairchild, *Color Appearance Models*, 2nd ed., Addison Wesley, Reading, MA (2005).
- H. J. Trussell, H. Zeng, and M. J. Vrhel, "Color correction and interpolation with few samples," *Proc. of IEEE Intl. Conf. on Image Processing* (2003).
- C. Gatta, S. Vacchi, D. Marini, and A. Rizzi, "Proposal for a new method to speed up local color correction algorithms," *Proc. SPIE* **5293**, 203–212 (2004).
- A. Rizzi, D. Marini, and L. D. Carli, "LUT and multilevel Brownian Retinex colour correction," *Mach. Graphics Vision* **11**, 153–168 (2002).
- A. Hertzmann, C. E. Jacobs, N. Oliver, B. Curless, and D. H. Salesin, "Image analogies," *SIGGRAPH*, pp. 327–340 (2001).
- E. Reinhard, M. Ashikhmin, B. Gooch, and P. Shirley, "Color transfer between images," *IEEE Comput. Graphics Appl.* **21**, 34–41 (2001).
- Y. Chang and S. Saito, "Example-based color stylization based on categorical perception," *Proc. of 1 Symp. on Applied Perception in Graphics and Visualization*, pp. 91–98 (2004).
- M. R. Gupta, R. M. Gray, and R. A. Olshen, "Nonparametric supervised learning with linear interpolation and maximum entropy," *IEEE Trans. Pattern Anal. Mach. Intell.* **28**, 766–781 (2006).
- T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning*, Springer-Verlag, New York (2001).
- T. Hastie and C. Loader, "Local regression: automatic kernel carpentry," *Stat. Sci.* **8**, 120–143 (1993).
- R. Bala, "Device characterization," Chapter 5 in *Digital Color Handbook*, G. Sharma, Ed., pp. 269–384, CRC Press, Boca Raton (2003).
- A. E. Hoerl and R. Kennard, "Ridge regression: biased estimation for nonorthogonal problems," *Technometrics* **12**, 55–67 (1970).
- K. Fukunaga and L. Hostetler, "Optimization of k -nearest neighbor density estimates," *IEEE Trans. Inf. Theory* **19**, 320–326 (1973).
- R. Short and K. Fukunaga, "The optimal distance measure for nearest neighbor classification," *IEEE Trans. Inf. Theory* **27**, 622–627 (1981).
- K. Fukunaga and T. Flick, "An optimal global nearest neighbor metric," *IEEE Trans. Pattern Anal. Mach. Intell.* **6**, 314–318 (1984).
- J. Myles and D. Hand, "An optimal global nearest neighbor metric," *Pattern Recogn.* **23**, 1291–1297 (1990).
- J. Friedman, "Flexible metric nearest neighbor classification," Technical Report, Stanford University, Stanford, CA (1994).
- C. Domeniconi, D. Gunopulos, and J. Peng, "Adaptive metric nearest neighbor classification," *IEEE Conf. on Computer Vision and Pattern Recognition*, 517–522 (2000).
- T. Hastie and R. Tibshirani, "Discriminative adaptive nearest neighbour classification," *IEEE Trans. Pattern Anal. Mach. Intell.* **18**, 607–615 (1996).
- L. Devroye, L. Györfi, and G. Lugosi, *A Probabilistic Theory of Pattern Recognition*, Springer-Verlag, New York (1996).
- R. Sibson, "A brief description of natural neighbour interpolation," Chapter 2 in *Interpreting Multivariate Data*, V. Barnett, Ed., pp. 21–36 Wiley, Hoboken, NJ (1981).
- B. Bhattacharya, K. Mukherjee, and G. Toussaint, "Geometric decision rules for high dimensions," *Proc. of 55th Session of Intl. Statistics Institute*, pp. 1–4 (2005).
- A. Okabe, B. Boots, K. Sugihara, and S. N. Chiu, *Spatial Tessellations—Concepts and Applications of Voronoi Diagrams*, 2nd ed. Wiley, Hoboken, NJ (2000).
- L. Devroye, "The expected size of some graphs in computational geometry," *Comput. Math. Appl.* **15**, 53–64 (1988).
- G. Toussaint, "Proximity graphs for nearest neighbor decision rules: recent progress," *Proc. of 34th Symp. on Comput. and Stat.* pp. 1–20 (2002).
- S. Buttrey and C. Karo, "Using k -nearest neighbor classification in the leaves of a tree," *Comput. Stat. Data Anal.* **40**, 27–37 (2002).
- H. Kang, *Color Technology for Electronic Imaging Devices*, SPIE, Bellingham, WA (1997).
- W. H. Press, W. T. Vetterling, S. A. Teukolsky, and B. P. Flannery, *Numerical Recipes in C*, 2nd ed. Cambridge University Press, Cambridge, England (1999).
- Matlab Version 7.0, Mathworks, (www.matlab.com) (2005).
- Cinecolor Web site by the American Widescreen Museum, (<http://www.widescreenmuseum.com/oldcolor/cinecolor2.html>) (2005).



Maya Gupta completed her PhD in electrical engineering in 2003 at Stanford University as a National Science Foundation Graduate Fellow. She completed her BS in electrical engineering and a BA in economics at Rice University, 1994 to 1997. From 1999 to 2003, she worked for Ricoh's California Research Center as a color image processing research engineer. In the fall of 2003, she joined the EE faculty of the University of Washington as an assistant professor. She was awarded the 2007 Office of Naval Research Young Investigator Award and the 2007 University of Washington, Department of Electrical Engineering Outstanding Teaching Award. More information about her research is available at her group's webpage: idl.ee.washington.edu.



Eric Garcia is an Intel/GEM fellow studying for his PhD in electrical engineering at the University of Washington, where he completed his MSEE in 2006. Before that he was a Gates Millennium Scholar at Oregon State University, where he finished the BS in computer engineering in 2004.

Andrey Strojlov completed his BA in computer science in 2005 at the University of Washington and currently works at Google.