Estimation for Color Engineering: Adaptive Neighborhoods and Regularized Local Linear Regression
Color management of printers
Custom color enhancements
Learning gamut expansions

regularized local linear regression over adaptive neighborhoods
what is color?
Color is an event

light source $I(w)$

reflection $\bar{I}(w)$

human perceives color

human cones respond:

\[
L = \int w \bar{I}(w)L(w)dw \\
M = \int w \bar{I}(w)M(w)dw \\
S = \int w \bar{I}(w)S(w)dw
\]

$L = \text{long wave} = \text{red}$

$M = \text{medium wave} = \text{green}$

$S = \text{short wave} = \text{blue}$
What does it mean to see black?

light source $I(w)$

reflection $\bar{I}(w)$

human perceives color

human cones respond

$L = \int_w \bar{I}(w)L(w)dw$

$M = \int_w \bar{I}(w)M(w)dw$

$S = \int_w \bar{I}(w)S(w)dw$

$L = \text{long wave} = \text{red}$

$M = \text{medium wave} = \text{green}$

$S = \text{short wave} = \text{blue}$
What does it mean to see white?

light source $I(w)$

reflection $\bar{I}(w)$

human perceives color

human cones respond

$L = \int_w \bar{I}(w)L(w)dw$

$M = \int_w \bar{I}(w)M(w)dw$

$S = \int_w \bar{I}(w)S(w)dw$

$L = \text{long wave} = \text{red}$

$M = \text{medium wave} = \text{green}$

$S = \text{short wave} = \text{blue}$
What does it mean to see white?

You can see “white” given light made up of 2-spectra.
Color management

**Problem:** How do you get a printer to print colors “correctly”?

**Motivation:** art, catalogs, product quality,
Color management

Problem: How do you get a printer to print colors “correctly”?

Motivation: art, catalogs, product quality, skin tones, preserve contrast and image quality
Color management

Problem: How do you get a printer to print colors “correctly”?

Motivation: art, catalogs, product quality, skin tones, preserve contrast and image quality, scientific and information visualization
Color management for Printers

8 bit RGB color patch → printed color patch → Human eye

device dependent color description

Measure CIEL*a*b*

device independent color description
CIELab: a device-independent colorspace

L* = lightness starting at black = 0… 100 = “reference white”

a* = red (a+) to green (a-)

b* = blue (b+) to yellow (b-)

Delta E = Euclidean distance in CIELab

Any two color pairs equal Delta E apart should be same perceptual difference

(image source: www.handprint.com)
Color management for Printers

Goal: Print a given CIEL*a*b* value.

Problem: What RGB value to input?
Step 1: characterize the printer

Print an image of RGB patches and measure the output CIEL*a*b* values
Step 2: Estimate 3D LUT that maps desired CIEL*a*b* colors to Input RGB

- Desired CIELab
- Estimated RGB input to Printer
Step 3: Interpolate 3D LUT

Desired CIEL*a*b* colors not on the grid are trilinearly interpolated to estimate best input RGB

<table>
<thead>
<tr>
<th>DESIRED CIEL<em>a</em>b*</th>
<th>RGB TO INPUT TO DEVICE</th>
</tr>
</thead>
<tbody>
<tr>
<td>(100,-50,-50)</td>
<td>(56,156,182)</td>
</tr>
<tr>
<td>(100,-50,-25)</td>
<td>(78, 174,98)</td>
</tr>
<tr>
<td>(100,-50,0)</td>
<td>(84,188,81)</td>
</tr>
<tr>
<td>(75,-50,-50)</td>
<td>(35, 104, 99)</td>
</tr>
<tr>
<td>(75,-50,-25)</td>
<td>(67,113,63)</td>
</tr>
<tr>
<td>(75,-50,0)</td>
<td>(88, 142, 24)</td>
</tr>
<tr>
<td>(50,-50,-50)</td>
<td>(14,82,85)</td>
</tr>
<tr>
<td>(50,-50,-25)</td>
<td>(53, 96, 58)</td>
</tr>
<tr>
<td>(50,-50,0)</td>
<td>(81,103, 23)</td>
</tr>
</tbody>
</table>
Printer color management summary

Step 1: Input RGB patches and measure CIEL*a*b* values

**Step 2:** Estimate RGB inputs corresponding to each color in 3D CIEL*a*b* grid

Step 3: Given a desired CIEL*a*b* color, interpolate the 3D LUT for best RGB input
Printer color management summary

Step 1: Input RGB patches and measure CIEL*a*b* values

Step 2: Estimate RGB inputs corresponding to each color in 3D CIEL*a*b* grid

Step 3: Given a desired CIEL*a*b* color, interpolate the 3D LUT for best RGB input

Bala 2003: best results with local linear regression compared to neural nets, polynomial regression, or splines.

This talk: better results by regularized local linear regression with adaptive neighborhood.
Regression

\[ f(x) \]

\[ x \]
Regression

\[ f(x) \]

\[ x_0 \]

\[ x \]
Linear Regression

\[ f(x) \]

\[ x_0 \quad x \]
Local Linear Regression

Fit a hyperplane to the $k$ nearest points

$f(x) \quad \hat{f}(x_2) \quad x_2 \quad x$

$k = 5$
Result of Local Linear Regression

$f(x)$
Local Ridge Regression

Local Linear Regression: \( \hat{f}(x) = \beta^* T x \)

\[ \beta^* = \arg \min_{\beta} \sum_{\text{neighbors } x_i} (\beta^T x_i - f(x_i))^2 \]

Local Ridge Regression: \( \hat{f}(x) = \beta^* T x \)

\[ \beta^* = \arg \min_{\beta} \sum_{\text{neighbors } x_i} (\beta^T x_i - f(x_i))^2 + \lambda \beta^T \beta \]
Regularized Regression

Fit a hyperplane to the $k$ nearest points

Ridge-fitted hyperplane is less steep
How to choose the neighborhood size?

Cross-validation
Set aside a portion of the training data to test on

Heuristics
Use what has worked in the past for your application:
k=15 (Bala 2003)

Adaptively
Geometrically adapt the neighborhood to the test point.
How do you choose near-neighbors for local learning?
How do you choose near-neighbors for local learning?

Choose a neighborhood that encloses the test point.
Enclosing Neighborhoods

Enclosing Neighborhood

A subset \( \{x_j\} \) of the training points such that the test point \( x \) can be represented as a \textbf{convex combination} of the points when possible:

\[
\sum_{i \in J} w_i x_i = x \quad \text{such that} \quad w_i \in [0, 1] \quad \text{and} \quad \sum_{i \in J} w_i = 1
\]

Using an Enclosing Neighborhood...

- Adapts the neighborhood to the local geometry
- Uses interpolation over extrapolation
Enclosing Neighborhoods Bound Estimation Variance

Let \( y = a^T x + a_0 + N_0 \) and \( y_j = a^T x_j + a_0 + N_j \) where \( N_0, \{N_j\} \) is iid with finite mean and variance. Consider the estimate \( \hat{y} = \hat{\beta}^T x + \hat{\beta}_0 \) where \( \beta, \beta_0 \) minimize the squared error on \( \{x_j, y_j\} \).

Then if \( x \) is in the convex hull of \( \{x_j\} \),

\[
\text{var}(\hat{y}) = E_N \left[ (\hat{y} - E_N[\hat{y}])^2 \right] \leq \sigma^2
\]
Enclosing Neighborhoods Bound Estimation Variance

Let $y = a^T x + a_0 + N_0$ and $y_j = a^T x_j + a_0 + N_j$ where $N_0, \{N_j\}$ is iid with finite mean and variance. Consider the estimate $\hat{y} = \hat{\beta}^T x + \hat{\beta}_0$ where $\beta, \beta_0$ minimize the squared error on $\{x_j, y_j\}$. Then if $x$ is in the convex hull of $\{x_j\}$,

$$\text{var}(\hat{y}) = E_N \left[ (\hat{y} - E_N[\hat{y}])^2 \right] \leq \sigma^2$$

Proof Sketch: Let $\tilde{\beta} = [\beta \beta_0]^T$. Let $\tilde{X}$ have $j$th column $[x_j \ 1]^T$. Using closed-form solution for $\tilde{\beta}$: $\text{cov}(\tilde{\beta}) = \sigma^2 \left( \tilde{X}^T \tilde{X} \right)^{-1}$
Enclosing Neighborhoods Bound Estimation Variance

Let \( y = a^T x + a_0 + N_0 \) and \( y_j = a^T x_j + a_0 + N_j \) where \( N_0, \{N_j\} \) is iid with finite mean and variance. Consider the estimate \( \hat{y} = \hat{\beta}^T x + \hat{\beta}_0 \) where \( \beta, \beta_0 \) minimize the squared error on \( \{x_j, y_j\} \).

Then if \( x \) is in the convex hull of \( \{x_j\} \),

\[
\text{var}(\hat{y}) = E_N \left[ (\hat{y} - E_N[\hat{y}])^2 \right] \leq \sigma^2
\]

Proof Sketch: Let \( \tilde{\beta} = [\beta \ \beta_0]^T \). Let \( \tilde{X} \) have \( j \)th column \( [x_j \ 1]^T \).

Using closed-form solution for \( \tilde{\beta} \):

\[
\text{cov}(\tilde{\beta}) = \sigma^2 \left( \tilde{X} \tilde{X}^T \right)^{-1}
\]

Then we show:

\[
\text{var}(\hat{y}) = [x \ 1] \text{cov}(\tilde{\beta}) [x \ 1]^T = [x \ 1] \sigma^2 \left( \tilde{X} \tilde{X}^T \right)^{-1} [x \ 1]^T
\]

Remains to show: \( [x \ 1] \left( \tilde{X} \tilde{X}^T \right)^{-1} [x \ 1]^T \leq 1 \) if \( x \) in convex hull
Enclosing Neighborhoods Bound Estimation Variance

Let \( y = a^T x + a_0 + N_0 \) and \( y_j = a^T x_j + a_0 + N_j \) where \( N_0, \{N_j\} \) is iid with finite mean and variance. Consider the estimate \( \hat{y} = \hat{\beta}^T x + \hat{\beta}_0 \) where \( \beta, \beta_0 \) minimize the squared error on \( \{x_j, y_j\} \).

Then if \( x \) is in the convex hull of \( \{x_j\} \),

\[
\text{var}(\hat{y}) = E_N \left[ (\hat{y} - E_N[\hat{y}])^2 \right] \leq \sigma^2
\]

Proof Sketch: Let \( \tilde{\beta} = [\beta \ \beta_0]^T \). Let \( \tilde{X} \) have \( j \)th column \([x_j \ 1]^T\).

Using closed-form solution for \( \tilde{\beta} \): \( \text{cov}(\tilde{\beta}) = \sigma^2 \left( \tilde{X} \tilde{X}^T \right)^{-1} \)

Then we show: \( \text{var}(\hat{y}) = [x \ 1] \text{cov}(\tilde{\beta}) [x \ 1]^T = [x \ 1] \sigma^2 \left( \tilde{X} \tilde{X}^T \right)^{-1} [x \ 1]^T \)

Remains to show: \( [x \ 1] \left( \tilde{X} \tilde{X}^T \right)^{-1} [x \ 1]^T \leq 1 \) if \( x \) in convex hull

Let \( w \) be minimum norm solution (pseudoinverse) to \( \tilde{X}w = [x \ 1]^T \)

Can show \( w^T w = [x \ 1] \left( \tilde{X} \tilde{X}^T \right)^{-1} [x \ 1]^T \)
Enclosing Neighborhoods Bound Estimation Variance

Let \( y = a^T x + a_0 + N_0 \) and \( y_j = a^T x_j + a_0 + N_j \) where \( N_0, \{N_j\} \) is iid with finite mean and variance. Consider the estimate \( \hat{y} = \hat{\beta}^T x + \hat{\beta}_0 \) where \( \beta, \beta_0 \) minimize the squared error on \( \{x_j, y_j\} \).

Then if \( x \) is in the convex hull of \( \{x_j\} \),

\[
\text{var}(\hat{y}) = E_N \left[ (\hat{y} - E_N[\hat{y}])^2 \right] \leq \sigma^2
\]

Proof Sketch: Let \( \tilde{\beta} = [\beta \ \beta_0]^T \). Let \( \tilde{X} \) have \( j \)th column \([x_j \ 1]^T \).

Using closed-form solution for \( \tilde{\beta} \): \( \text{cov}(\tilde{\beta}) = \sigma^2 \left( \tilde{X} \tilde{X}^T \right)^{-1} \)

Then we show: \( \text{var}(\hat{y}) = [x \ 1] \text{cov}(\tilde{\beta})[x \ 1]^T = [x \ 1] \sigma^2 \left( \tilde{X} \tilde{X}^T \right)^{-1} [x \ 1]^T \)

Remains to show: \( [x \ 1] \left( \tilde{X} \tilde{X}^T \right)^{-1} [x \ 1]^T \leq 1 \) if \( x \) in convex hull

Let \( w \) be minimum norm solution (pseudoinverse) to \( \tilde{X}w = [x \ 1]^T \)

Can show \( w^Tw = [x \ 1] \left( \tilde{X} \tilde{X}^T \right)^{-1} [x \ 1]^T \)

If \( x \) is in the convex hull of \( \{x_j\} \), there exists weights \( v \in [0, 1]^k \) s.t. \( \tilde{X}v = [x \ 1]^T \).

Then \( 0 \leq w^Tw \leq v^Tv = \sum_j v_j^2 \leq \sum_j v_j = 1 \)
Natural neighbors (Sibson '81) is an enclosing neighborhood

Natural Neighbors of g is the set of all points whose Voronoi cells are adjacent to the cell of g.

Sibson proposed 1981 as neighborhood for linear interpolation.
Natural neighbors (Sibson ’81) is an enclosing neighborhood

Natural Neighbors of \( g \) is the set of all points whose Voronoi cells are adjacent to the cell of \( g \).

Sibson proposed 1981 as neighborhood for linear interpolation.

(natural neighbors inclusive (Gupta ’07): add all nearer neighbors to the natural neighbors.)
Enclosing k-NN neighborhood (Gupta '05)

The neighborhood about $g$ with the fewest number of neighbors $k$ that achieves the minimum distance to their convex hull.

(The neighborhood about $g$ with the fewest number of neighbors $k$ that enclose $g$ in their convex hull.)
Natural neighbors

Enclosing k-NN
Drawbacks to enclosing neighborhoods

**Computationally Expensive**

**Natural Neighbors**
Voronoi tessellation of entire training set and test point.
\[
\text{Worst Case: } O(n^{\left\lfloor \frac{d}{2} \right\rfloor})
\]

**Enclosing k-NN**
QP at each step to find distance to neighbors.
\[
\text{Worst Case: } O(n^4)
\]

**Test point outside convex hull of data**
Sometimes in 3D, common in higher dimensions
Expected number of neighbors in enclosing k-NN neighborhood

1-d case: assume training points \( \sim \) uniformly in \([0,1]\)

\(\text{Need at least 1 neighbor. Then need a neighbor on the other side}\)

\[P(\text{neighbor on other side}) = \frac{1}{2}\]

\[E[\text{draws until neighbor on other side}] = 2\]

Expected Total Neighbors = 1 + 2 = 3
Expected number of neighbors in enclosing k-NN neighborhood

2-dimensional Case:

x1

x2
Expected number of neighbors in enclosing k-NN neighborhood

2-dimensional Case:

- $x_1$
- $x_2$
- $x_3$
Expected number of neighbors in enclosing k-NN neighborhood

2-dimensional Case:
Expected size of enclosing k-NN

Given neighbors drawn uniformly around test point in \( d \)-dimensional feature space,

\[
E[\# \text{ Neighbors}] = 2d + 1
\]
as number of training samples \( n \) goes to infinity.
**Expected size of Enclosing k-NN = 2d+1**

**Thm:** If d-dimensional training points drawn uniformly randomly around test point, the enclosing k-NN neighborhood will have an average of $2d + 1$ neighbors.

**Proof:**
Combinatorial geometry (Wendel 1962) gives $P(n \text{ pts uniformly drawn on a hypersphere are from same half-space})$.

We need:
$E[ N \text{ pts to first not be on same half-space}]$

Used:
Wendel’s result + bunch of discrete math tricks
Experimental neighborhood sizes

![Graphs showing frequency of different neighborhood sizes](image-url)
Adaptive LLR and Color Management

Step 1: Input RGB patches and measure CIEL*a*b* values

Step 2: *Estimate RGB inputs corresponding to a 3D CIEL*a*b* grid*

Step 3: Given a desired CIEL*a*b* color, interpolate the 3D grid for best RGB input

We compared local linear and local ridge regression for $k=15$, enclosing k-NN, natural neighbors.
## Example Color Management Results

Ricoh Laser Printer, 918 training patches, 729 in-gamut test patches, regularization parameter fixed at .1

<table>
<thead>
<tr>
<th>Neighborhood</th>
<th>Method</th>
<th>Mean $\Delta E_{94}$ Error</th>
<th>95th %ile Error</th>
</tr>
</thead>
<tbody>
<tr>
<td>Enclosing k-NN</td>
<td>Linear</td>
<td>4.3</td>
<td>8.5</td>
</tr>
<tr>
<td></td>
<td>Ridge</td>
<td>3.7</td>
<td>7.4</td>
</tr>
<tr>
<td>Enclosing k-NN</td>
<td>Linear</td>
<td>4.0</td>
<td>8.3</td>
</tr>
<tr>
<td>Minimum 15</td>
<td>Ridge</td>
<td>3.5</td>
<td>6.8</td>
</tr>
<tr>
<td>Natural Neighbors</td>
<td>Linear</td>
<td>3.7</td>
<td>7.6</td>
</tr>
<tr>
<td></td>
<td>Ridge</td>
<td>3.7</td>
<td>7.1</td>
</tr>
<tr>
<td>15 Neighbors</td>
<td>Linear</td>
<td>4.4</td>
<td>9.8</td>
</tr>
<tr>
<td></td>
<td>Ridge</td>
<td>4.2</td>
<td>8.6</td>
</tr>
</tbody>
</table>
Example Color Management Results

Ricoh Laser Printer, 918 training patches, 729 in-gamut test patches, regularization parameter fixed at .1

<table>
<thead>
<tr>
<th>Neighborhood</th>
<th>Method</th>
<th>Mean $\Delta E_{94}$ Error</th>
<th>95th %ile Error</th>
</tr>
</thead>
<tbody>
<tr>
<td>Enclosing k-NN</td>
<td>Ridge</td>
<td>3.7</td>
<td>7.4</td>
</tr>
<tr>
<td>Enclosing k-NN</td>
<td>Ridge</td>
<td>3.5</td>
<td>6.8</td>
</tr>
<tr>
<td>Minimum 15</td>
<td>Ridge</td>
<td>3.7</td>
<td>7.1</td>
</tr>
<tr>
<td>Natural Neighbors</td>
<td>Ridge</td>
<td>4.2</td>
<td>8.6</td>
</tr>
<tr>
<td>15 Neighbors</td>
<td>Ridge</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Related problem:

Learning custom color transformations
Goal: give digital designers tools to create custom color enhancements that are easy to use, share, and modify

*Ex: simulating illumination effects*

![original](image1)

![transformed](image2)

original

transformed by artist to “sunset”

2 hrs. work in Photoshop
Proposed custom color enhancement architecture

- Input image
- Standard color management module
- ICC profile (3D look-up table)
- Statistical learning algorithm
- Enhanced image

Reference set of sample color pairs
Proposed custom color enhancement architecture

user controls
requires little
very flexible

reference
set of sample
color pairs

standard color
management module

enhanced
image

input
image

ICC profile
(3D look-up table)

statistical
learning algorithm
Proposed custom color enhancement architecture

- Input image
- Reference set of sample color pairs
- User controls require little very flexible
- Very flexible non-proprietary CMM’s common easy to use, modify, share
- Standard color management module
- ICC profile (3D look-up table)
- Statistical learning algorithm
- Enhanced image
Example
Convert an image to how it would look in Cinecolor based on 16 sample color pairs

www.widescreenmuseum.org
Convert original to Cinecolor

Local linear regression on 4 nearest neighbors

Local ridge regression on 4 nearest neighbors
Convert original to Cinecolor

Local linear regression on smallest enclosing neighborhood

Local ridge regression on smallest enclosing neighborhood
Convert original to Husky Colors

Local ridge regression on smallest enclosing neighborhood

Local ridge regression on n-1 nearest-neighbors
Smarter Regularization

Model: \( f(x) = \beta^T x \)

Ridge:
\[
\arg \min_{\beta} \sum_{\text{neighbors } x_i} (\beta^T x_i - f(x_i))^2 + \lambda \beta^T \beta
\]

Tikhonov:
\[
\arg \min_{\beta} \sum_{\text{neighbors } x_i} (\beta^T x_i - f(x_i))^2 + \lambda (\beta - \beta_g)^T (\beta - \beta_g)
\]

Local-Global:
\[
\arg \min_{\beta} \frac{1 - \lambda}{k} \sum_{\text{neighbors } x_i} (\beta^T x_i - f(x_i))^2 + \frac{\lambda}{n} \sum_{\text{all } X_j} (\beta^T X_j - f(X_j))^2
\]
Bayesian perspective for Ridge

Model: \( f(x_i) = \beta^T x_i + N_i \) where \( N_i \) is iid \( \mathcal{N}(0, I) \)

\[
\arg\max_{\beta} p(\beta | \{x_i, f(x_i)\}) \\
\equiv \arg\max_{\beta} p(\{x_i, f(x_i)\} | \beta)p(\beta) \\
\equiv \arg\max_{\beta} p(\{N_i = f(x_i) - \beta^T x_i\} | \beta)p(\beta) \\
\equiv \arg\min_{\beta} \sum_{i} (f(x_i) - \beta^T x_i)^2 - \ln p(\beta)
\]
Bayesian perspective for Ridge

Model: \( f(x_i) = \beta^T x_i + N_i \) where \( N_i \) is iid \( \mathcal{N}(0, I) \)

\[
\arg \max_{\beta} p(\beta \mid \{x_i, f(x_i)\})
\]
\[
\equiv \arg \max_{\beta} p(\{x_i, f(x_i)\} \mid \beta) p(\beta)
\]
\[
\equiv \arg \max_{\beta} p(\{N_i = f(x_i) - \beta^T x_i\} \mid \beta) p(\beta)
\]
\[
\equiv \arg \min_{\beta} \sum_i (f(x_i) - \beta^T x_i)^2 - \ln p(\beta)
\]

Prior model on \( \beta \): \( p(\beta) \) is \( \mathcal{N}(0, \frac{I}{\lambda}) \)

\[
\equiv \arg \min_{\beta} \sum_{\text{neighbors } x_i} (\beta^T x_i - f(x_i))^2 + \lambda \beta^T \beta \quad \text{(Ridge)}
\]
Bayesian perspective for Tikhonov

Model: \( f(x_i) = \beta^T x_i + N_i \) where \( N_i \) is iid \( \mathcal{N}(0, I) \)

\[
\arg \max_{\beta} p(\beta | \{x_i, f(x_i)\})
\]

\[\equiv \arg \max_{\beta} p(\{x_i, f(x_i)\} | \beta) p(\beta)\]

\[\equiv \arg \max_{\beta} p(\{N_i = f(x_i) - \beta^T x_i\} | \beta) p(\beta)\]

\[\equiv \arg \min_{\beta} \sum_i (f(x_i) - \beta^T x_i)^2 - \ln p(\beta)\]

Prior model on \( \beta \): \( p(\beta) \) is \( \mathcal{N}(\beta_g, \frac{I}{\lambda}) \) yields Tikhonov:

\[\equiv \arg \min_{\beta} \sum_{\text{neighbors } x_i} (\beta^T x_i - f(x_i))^2 + \lambda (\beta - \beta_g)^T (\beta - \beta_g)\]
Bayesian perspective for Local-global

Model: \( f(x_i) = \beta^T x_i + N_i \) where \( N_i \) is iid \( \mathcal{N}(0, \frac{I}{1-\lambda}) \)

No prior on \( \beta \) (flat prior),
and \( f(X_j) = \beta^T X_j + E_j \) where \( E_j \) is iid \( \mathcal{N}(0, \frac{I}{\lambda}) \)

Then,

\[
\arg\max_\beta p(\beta | \{x_i, f(x_i)\}, \{X_j, f(X_j)\})
\]

Produces local-global regularization:

\[
\arg\min_\beta \frac{1 - \lambda}{k} \sum_{\text{neighbors } x_i} (\beta^T x_i - f(x_i))^2 + \frac{\lambda}{n} \sum_{\text{all } X_j} (\beta^T X_j - f(X_j))^2
\]
Example Color Management Results

HP InkJet Printer, 918 training patches, 918 in-gamut test patches, Regularization parameter smallest that did not cause false contours in a set of 4 Kodak images

<table>
<thead>
<tr>
<th>Neighborhood</th>
<th>Regularization</th>
<th>Mean Error</th>
<th>95th % Error</th>
<th>Max Error</th>
</tr>
</thead>
<tbody>
<tr>
<td>Enclosing k-NN</td>
<td>Ridge</td>
<td>2.9</td>
<td>5.7</td>
<td>14.1</td>
</tr>
<tr>
<td></td>
<td>Tikhonov</td>
<td>2.5</td>
<td>4.7</td>
<td>7.8</td>
</tr>
<tr>
<td></td>
<td>Local-Global</td>
<td>2.0</td>
<td>4.0</td>
<td>6.9</td>
</tr>
<tr>
<td>Natural Neighbors</td>
<td>Ridge</td>
<td>2.6</td>
<td>4.9</td>
<td>6.7</td>
</tr>
<tr>
<td></td>
<td>Tikhonov</td>
<td>2.3</td>
<td>4.6</td>
<td>6.5</td>
</tr>
<tr>
<td></td>
<td>Local-Global</td>
<td>2.3</td>
<td>4.5</td>
<td>6.7</td>
</tr>
</tbody>
</table>
Related problem:

Gamut expansion
Problem: extend an image-gamut to take advantage of the display

- Extended gamut monitors provide **new colors**
- Favorite HDTV movies encoded in sRGB
- Enhance gamut appropriately for a given display and image set/video
Extended Gamut Displays are here

**Sharp**: 5 wavelengths displays:
Richer reds by using multi-spike backlight LEDs

**Brightside**: spatially modulated backlight

*Genoa*: RGBCY primaries
Gamut expansion: extrapolation/estimation problem

Standard: Linear transformation shifts skin tones, pastels and neutrals to be too saturated.
Proposed Gamut Mapping Architecture

• Expert provides custom enhancement for a single image

• Nonlinear transformation is learned and applied to each subsequent image in the scene
Proposed Gamut Expansion Architecture

Expert creates sample pairs of original to enhanced colors

Local linear fit to sample pairs

3D LUT forms core of standard ICC profile
Proposed Gamut Expansion Architecture

Single frame → Enhanced

Local linear fit to sample pairs

3D LUT

image sequence

Enhanced sequence

Color Management Module (CMM)
General/Specific Gamut Expansion

Flexible: Create ICC profile to specify gamut mapping

General: Gamut map for a given display type

Specific: Gamut map for a given video/image set for a given standardized extended gamut space or display
Experimental Setup

Original video sequence

Simulated “small gamut” sequence

Color selective desaturation

Enhanced video sequence

Gamut expansion via LUT

Quantitative comparison
Compressed gamut

Original gamut

Compressed gamut
Training image
Gamut expansion results

Truth

Global linear

K=100 LLR

“Reduced Gamut”
Gamut expansion results: a closer look
Gamut expansion results:
a closer look

Truth

Global linear

K=100 LLR

"Original"
Gamut expansion results: a closer look
Gamut expansion results: a closer look
Comparison: Night Sequence

- Global linear 95% worst error
- kNN 95% worst error
- Global linear median error
- kNN median error
take-away’s

• Color is hard but there’s a lot we can do

• Mixtures of simple algorithms (e.g. locally linear models) can capture complex behavior

• Learning: what’s asymptotically good and what’s helpful in a finite case may not be the same

• Defining neighbors for local learning an open question

• Ask: what else can I do with this tool/framework/standard?
Related publications ( available at idl.ee.washington.edu )


“Learning ICC profiles for custom color transformations with adaptive neighborhoods,” Gupta Garcia, (In review for journal publication)