EECE/CS 4353 Image Processing

Lecture Notes: Color Correction

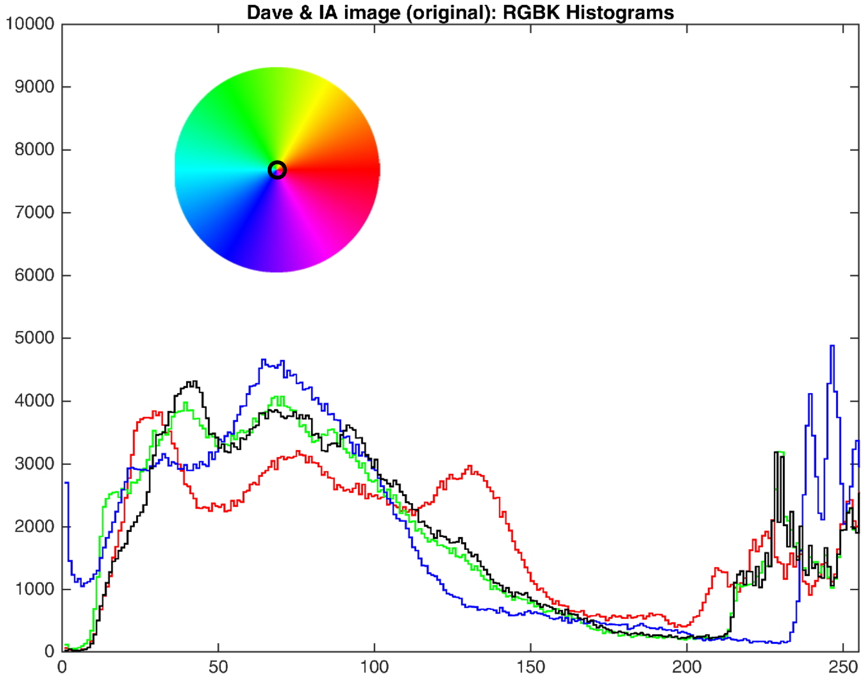
Richard Alan Peters II
Department of Electrical and Computer Engineering
Fall Semester 2021

Color Correction

is a global change in the coloration of an image to alter its tint, its hues, or the saturation of its colors with minimal changes to its luminant features.

Gamma Adjustment of Color Bands

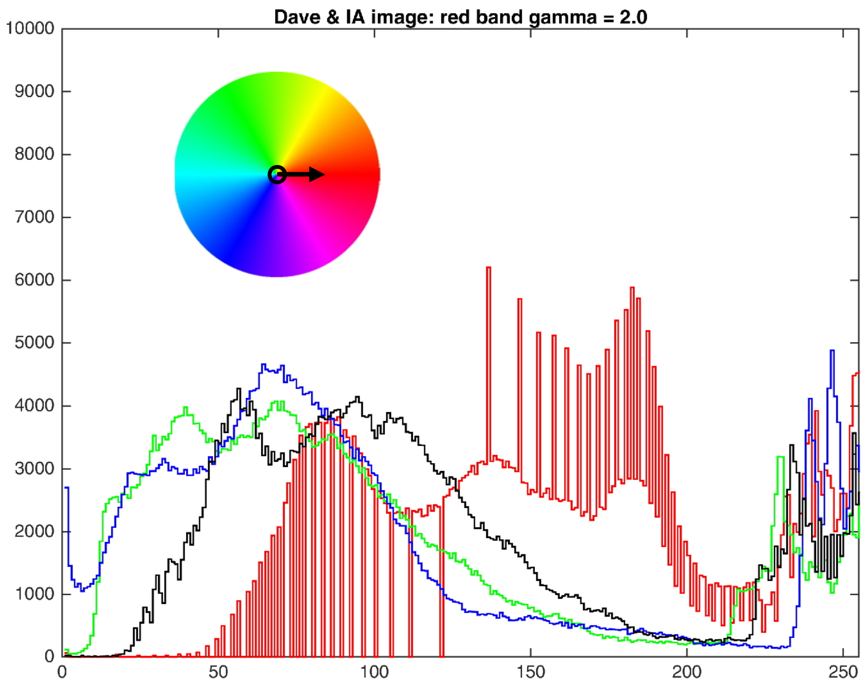
original



David Peters, producer, and representatives of the IA, The International Alliance of Theatrical Stage Employees, Moving Picture Technicians, Artists and Allied Crafts, on the set of *Frozen Impact* (PorchLight Entertainment, 2003).

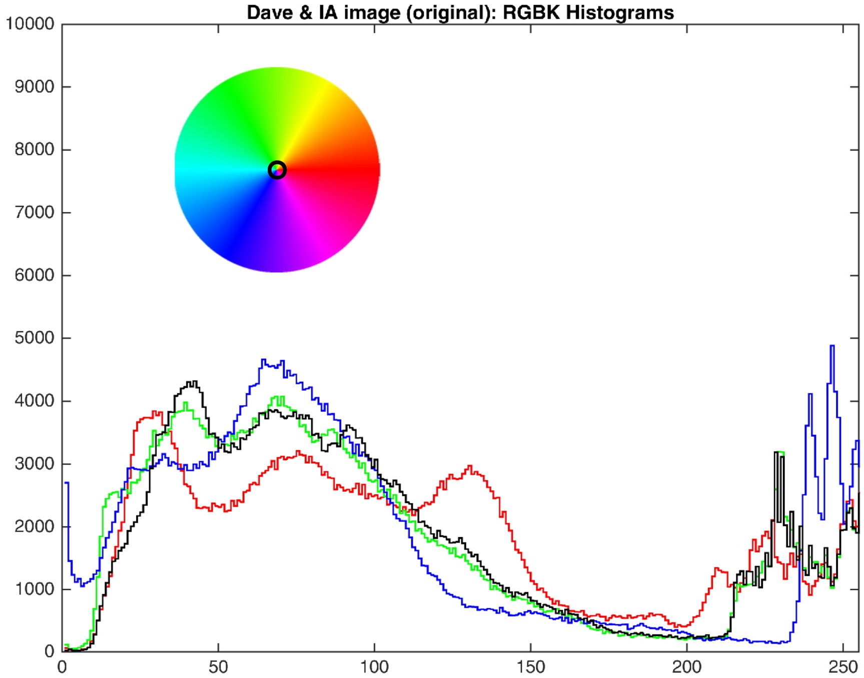
Gamma Adjustment of Color Bands

red γ=2



Gamma Adjustment of Color Bands

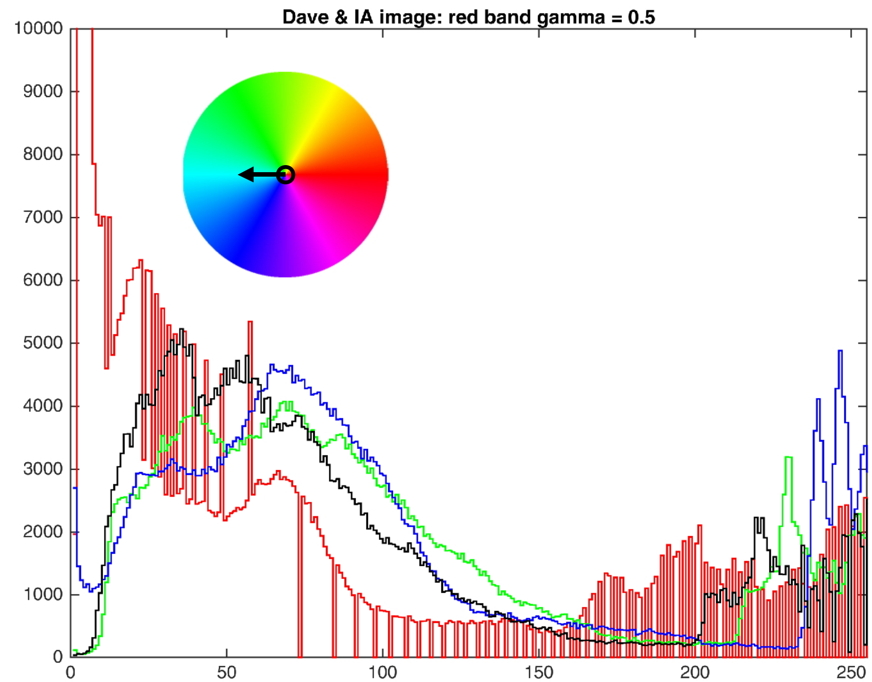
original



David Peters, producer, and representatives of the IA, The International Alliance of Theatrical Stage Employees, Moving Picture Technicians, Artists and Allied Crafts, on the set of *Frozen Impact* (PorchLight Entertainment, 2003).

Gamma Adjustment of Color Bands

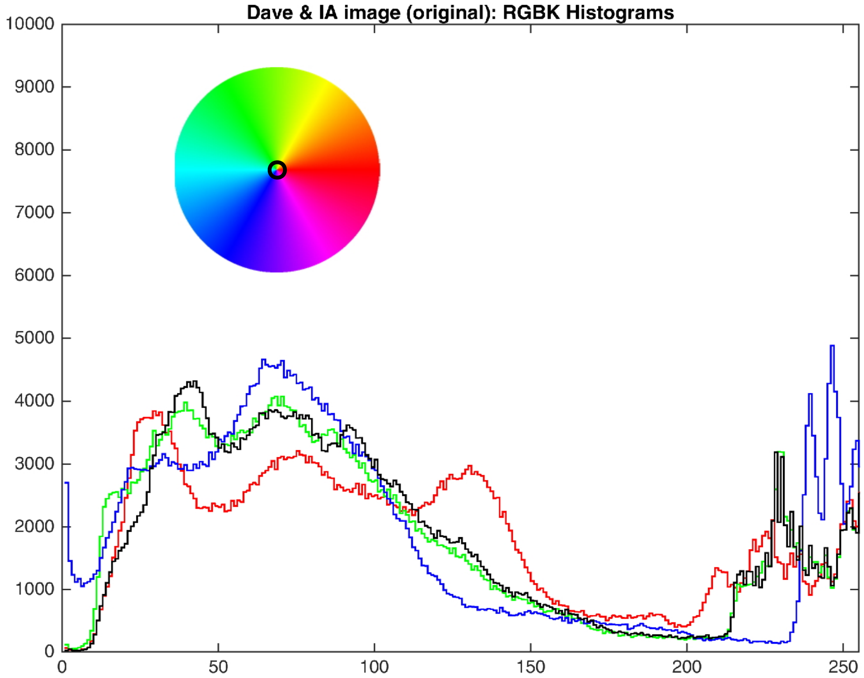
red γ=0.5



reduced red = increased cyan

Gamma Adjustment of Color Bands

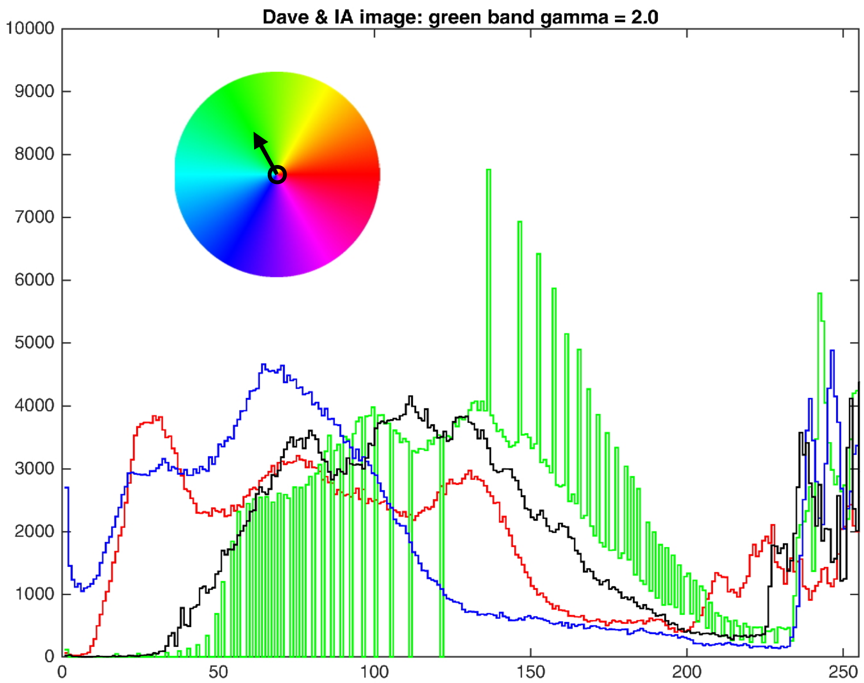
original



David Peters, producer, and representatives of the IA, The International Alliance of Theatrical Stage Employees, Moving Picture Technicians, Artists and Allied Crafts, on the set of *Frozen Impact* (PorchLight Entertainment, 2003).

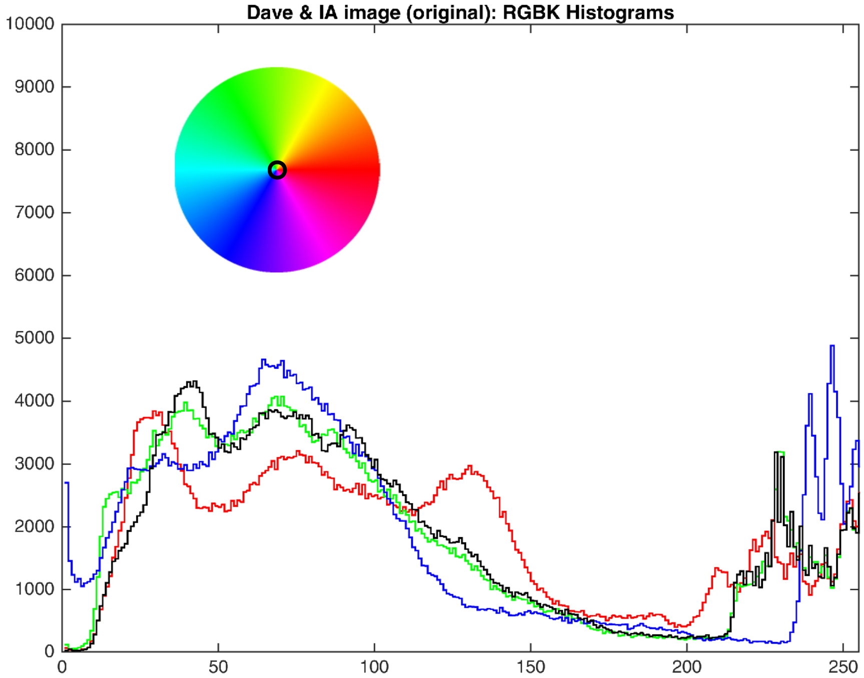
Gamma Adjustment of Color Bands

green γ=2



Gamma Adjustment of Color Bands

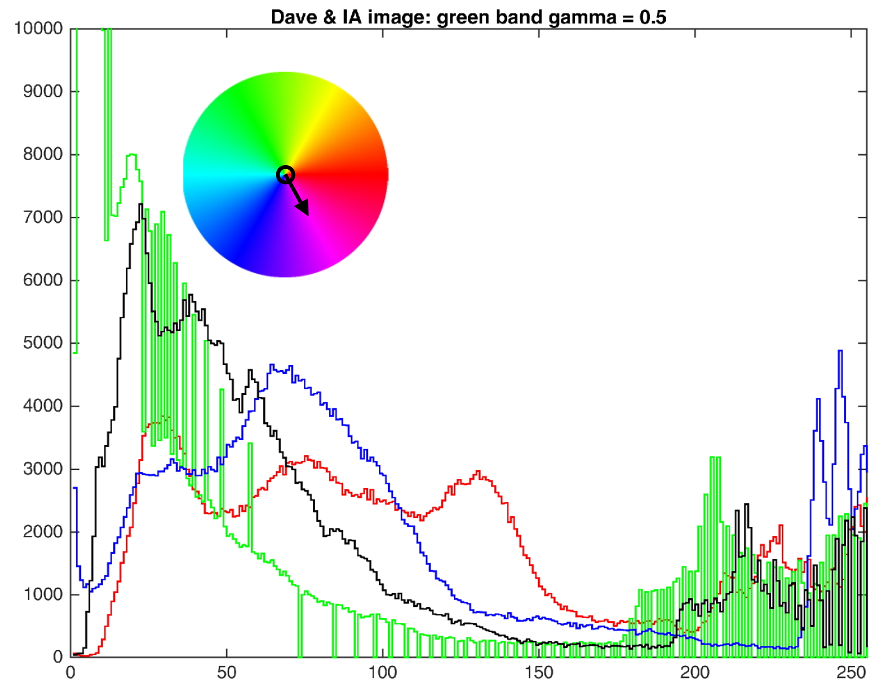
original



David Peters, producer, and representatives of the IA, The International Alliance of Theatrical Stage Employees, Moving Picture Technicians, Artists and Allied Crafts, on the set of *Frozen Impact* (PorchLight Entertainment, 2003).

Gamma Adjustment of Color Bands

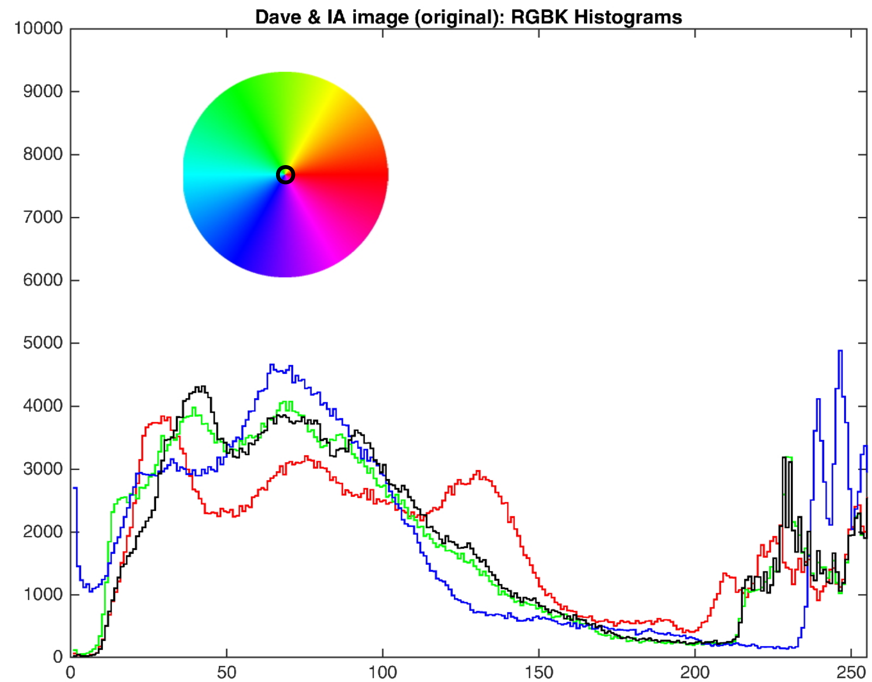
green γ=0.5



reduced green = incr. magenta

Gamma Adjustment of Color Bands

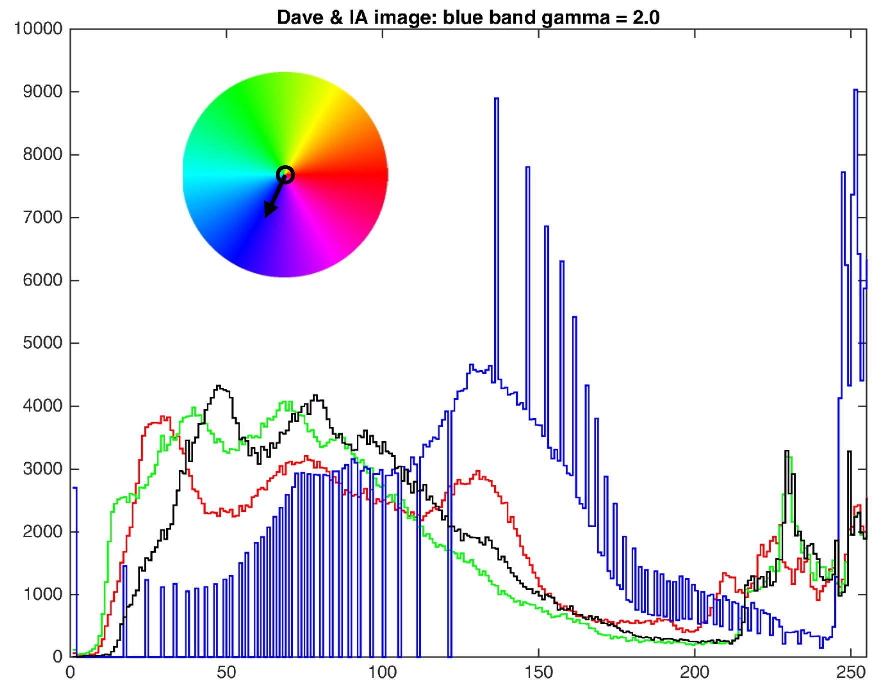
original



David Peters, producer, and representatives of the IA, The International Alliance of Theatrical Stage Employees, Moving Picture Technicians, Artists and Allied Crafts, on the set of *Frozen Impact* (PorchLight Entertainment, 2003).

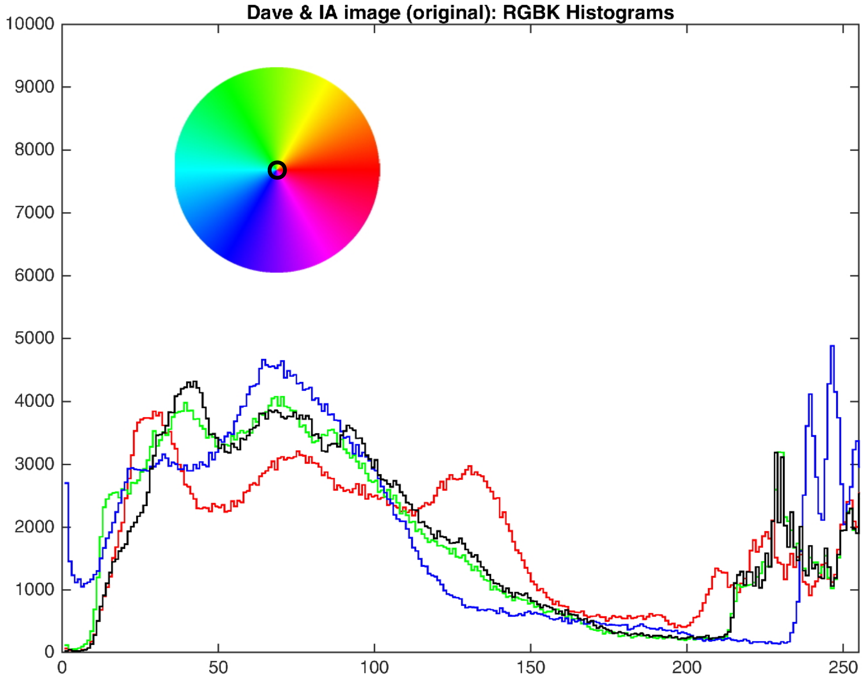
Gamma Adjustment of Color Bands

blue γ=2



Gamma Adjustment of Color Bands

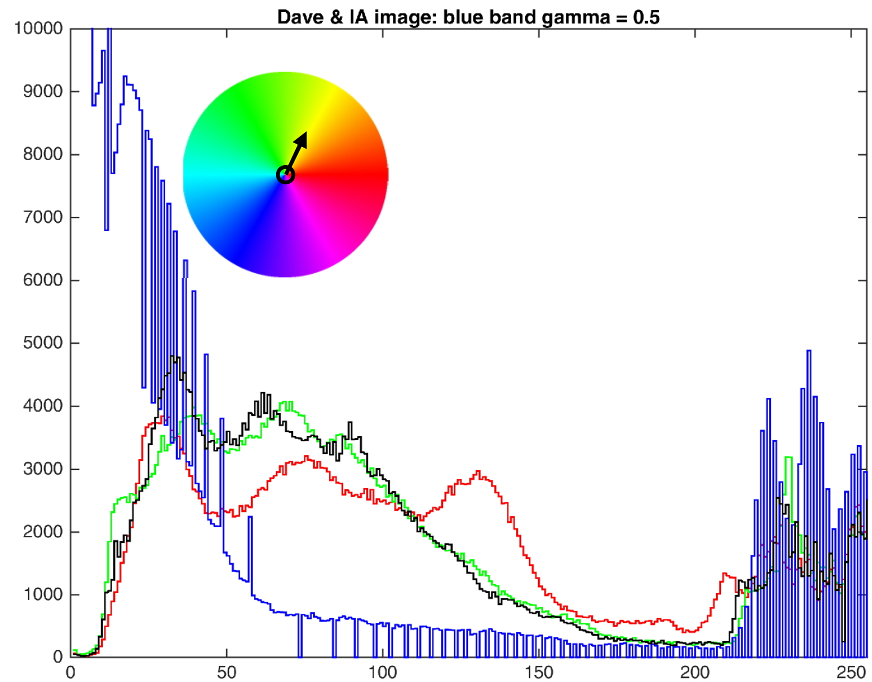
original



David Peters, producer, and representatives of the IA, The International Alliance of Theatrical Stage Employees, Moving Picture Technicians, Artists and Allied Crafts, on the set of *Frozen Impact* (PorchLight Entertainment, 2003).

Gamma Adjustment of Color Bands

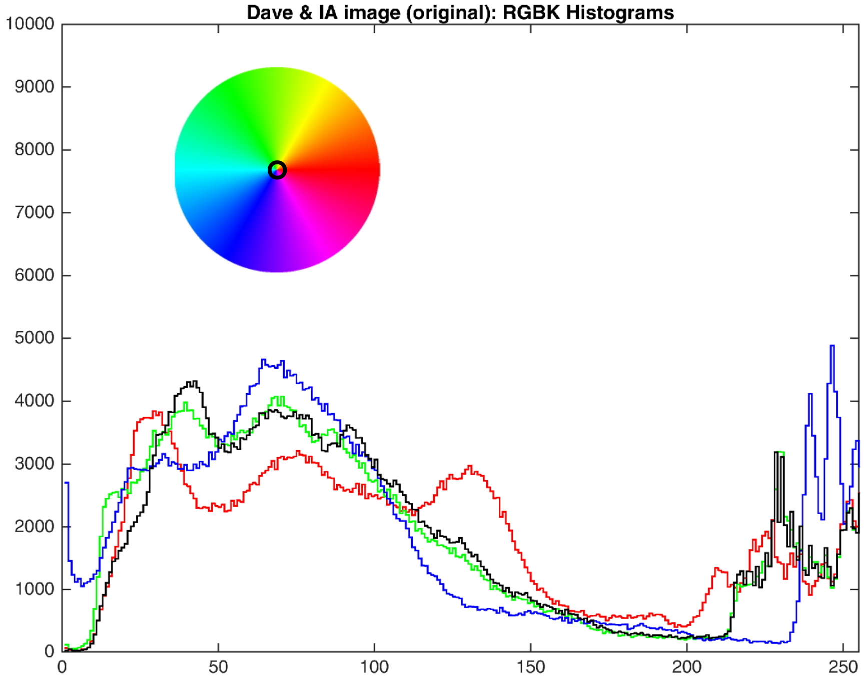
blue γ=0.5



reduced blue = incr. yellow

Gamma Adjustment of Color Bands

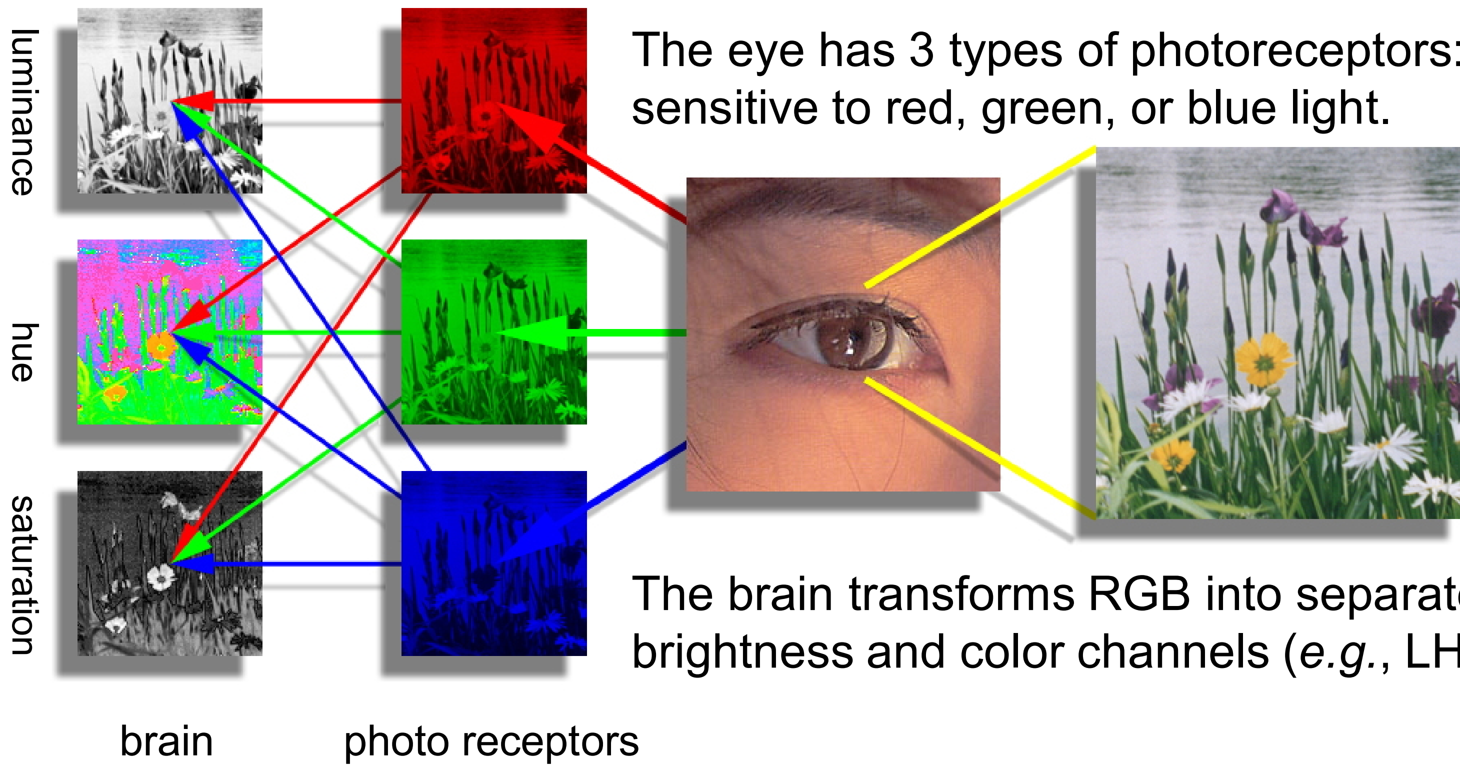
original



David Peters, producer, and representatives of the IA, The International Alliance of Theatrical Stage Employees, Moving Picture Technicians, Artists and Allied Crafts, on the set of *Frozen Impact* (PorchLight Entertainment, 2003).

RGB to LHS: A Perceptual Transformation

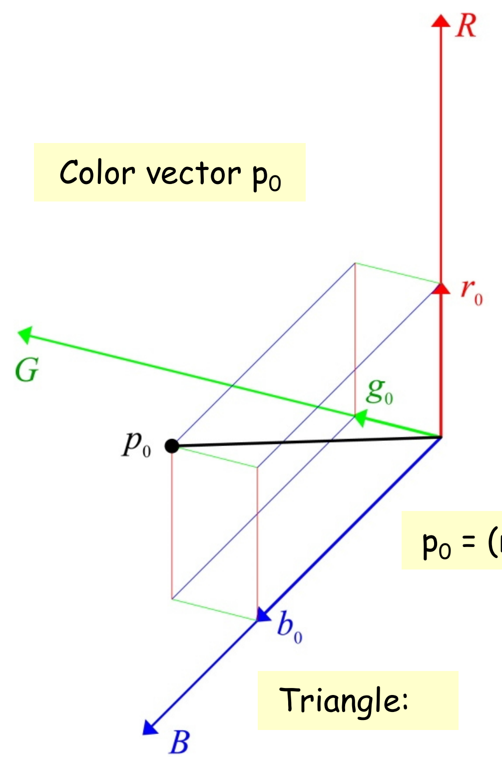
The eye has 3 types of photoreceptors: sensitive to red, green, or blue light.



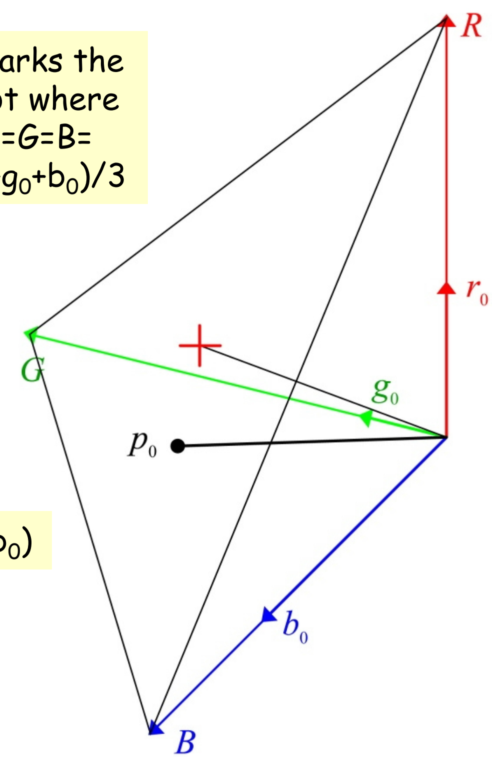
The brain transforms RGB into separate brightness and color channels (*e.g.*, LHS).

HSV Color Representation

Color vector p_0



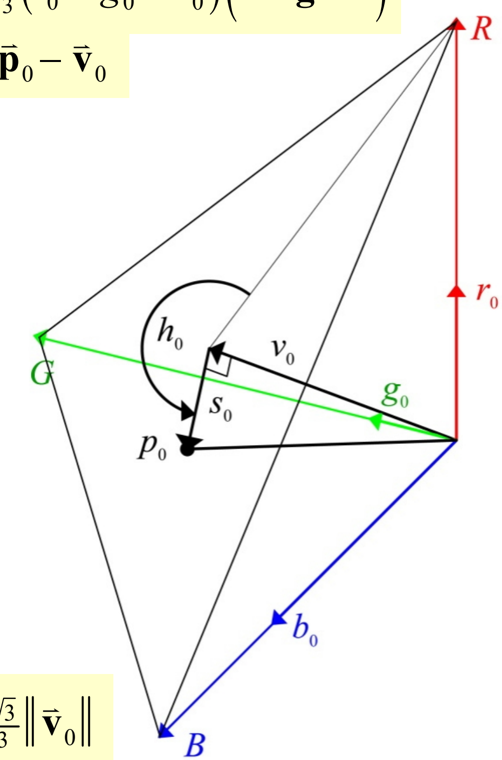
+ marks the spot where R=G=B= $(r_0+g_0+b_0)/3$



$p_0 = (r_0, g_0, b_0)$

$$\vec{\mathbf{v}}_0 = \tfrac{1}{3}\left(r_0 + g_0 + b_0\right)\left(\hat{\mathbf{r}} + \hat{\mathbf{g}} + \hat{\mathbf{b}}\right)$$

$$\vec{\mathbf{s}}_0 = \vec{\mathbf{p}}_0 - \vec{\mathbf{v}}_0$$



Triangle:

lies in plane R+G+B= $(r_0+g_0+b_0)$,

intersects R, G, & B axes at $r_0+g_0+b_0$, and contains p_0.

$$v_0 = \tfrac{\sqrt{3}}{3}\left\|\vec{\mathbf{v}}_0\right\|$$

$$s_0 = \left\|\vec{\mathbf{s}}_0\right\|$$

$$h_0 = \angle\left\{\left[\left(r_0 + g_0 + b_0\right)\hat{\mathbf{r}} - \vec{\mathbf{v}}_0\right], \vec{\mathbf{s}}_0\right\}$$

A Fast RGB to HSV Algorithm

Given color $\mathbf{p} = [R\ G\ B]^T$ where $R, G, B \in \{0,\ldots,255\}$, to compute $[h\ s\ v]^T$ where s, v $\in$[0,1] and h $\in$[0,360) the algorithm proceeds as follows:

```
1. Compute [r g b] = [R G B]/255.
2. Set m = min(r,g,b) , M = max(r,g,b).
3. Set v = M.
4. Compute C = M – m.
5. If C == 0 then s=0, h=0. Return [h s v]^T.
6. s = C/M.
7. If M==r then h = ((g-b)/c) modulo 6.
8. else if M==g then h = 2 + (b-r)/c.
9. else h = 4 + (r-g)/c.
10.h = 60h.
```

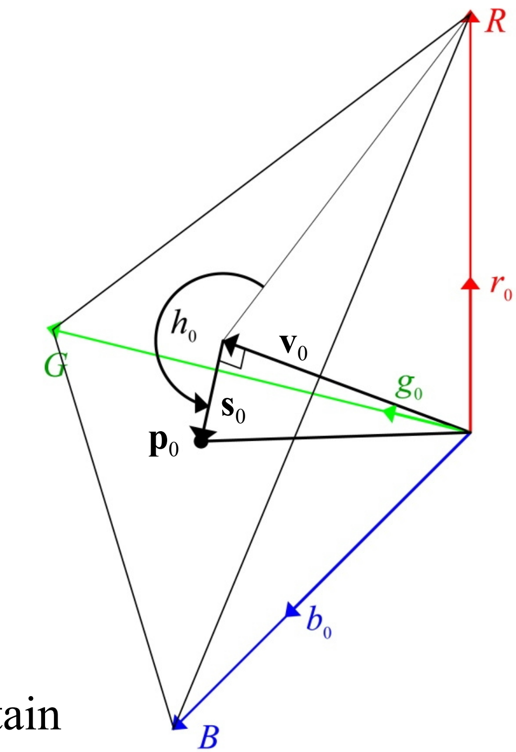
R,G,B are numbers here not images.

Experiments with Matlab show this algorithm to be 3 times faster than Algorithm 1 and 1.13 faster than Algorithm 2 (EECE_4353_06_RGBandHSVColor). The numbers output by this one differ from the other two.

Reference: HSL and HSV - Wikipedia, the free encyclopedia

HSV to RGB Conversion

The x, y, & z unit vectors in r, g, & b coordinates are the columns of the rotation matrix:



Therefore, the rotation matrix is

$$A = \frac{\sqrt{6}}{6}\begin{bmatrix} 2 & 0 & \sqrt{2} \\ -1 & \sqrt{3} & \sqrt{2} \\ -1 & -\sqrt{3} & \sqrt{2} \end{bmatrix}.$$

Substitute that into the 2nd equation on slide 94 to get:

$$\begin{aligned}[\mathbf{s}]_{\mathbf{rgb}} &= s\tfrac{\sqrt{6}}{6}\cos(h)\begin{bmatrix} 2 \\ -1 \\ -1 \end{bmatrix} + s\tfrac{\sqrt{2}}{2}\sin(h)\begin{bmatrix} 0 \\ 1 \\ -1 \end{bmatrix} + 0\tfrac{\sqrt{3}}{3}\begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} \\ &= s\tfrac{\sqrt{6}}{6}\cos(h)\begin{bmatrix} 2 \\ -1 \\ -1 \end{bmatrix} + s\tfrac{\sqrt{2}}{2}\sin(h)\begin{bmatrix} 0 \\ 1 \\ -1 \end{bmatrix}.\end{aligned}$$

Finally, $[\mathbf{s}]_{\mathbf{rgb}}$ must be translated to the value vector to obtain the rgb color of $\mathbf{p}_0$:

$$\mathbf{p}_0 = [\mathbf{p}]_{\mathbf{rgb}} = [\mathbf{s}]_{\mathbf{rgb}} + [\mathbf{v}]_{\mathbf{rgb}},$$ where $\mathbf{s}_0 = [\mathbf{s}]_{\mathbf{rgb}}$ and $[\mathbf{v}]_{\mathbf{rgb}} = \mathbf{v}_0$ as def'd. on slide 81.

This algorithm is the inverse of the hex projection.

A Fast HSV to RGB Algorithm

Given vector $\mathbf{h}^T = [h\ s\ v]$ where $h \in [0, 360)$, $s \in [0, 1]$, and $v \in [0, 1]$, to compute $\mathbf{p}^T = [r\ g\ b]$ where $r, g, b \in \{0, \ldots, 255\}$:

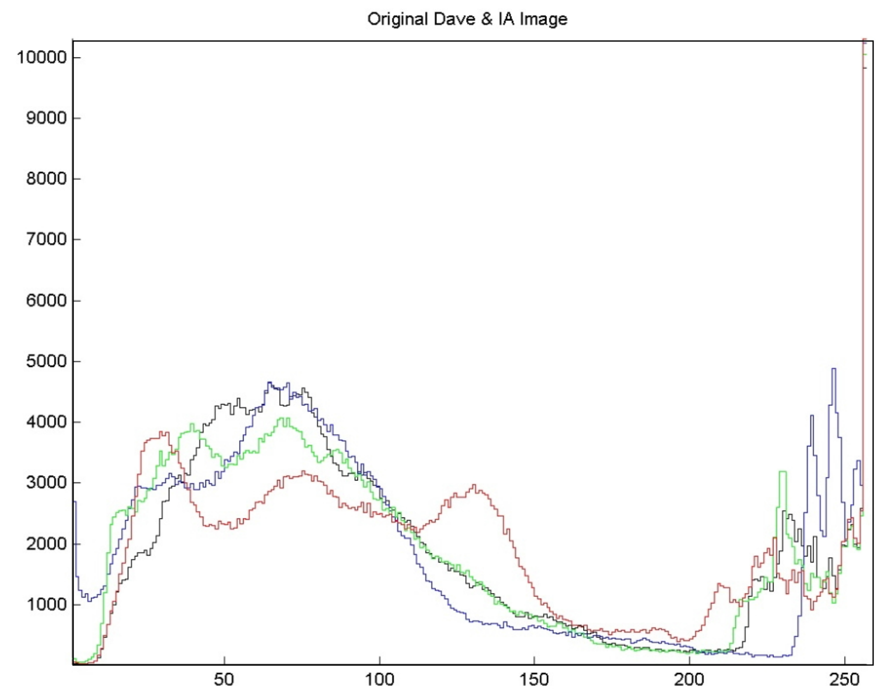
```
1. H = h/60.
2. C = v·s.
3. D = v-C.
4. X = C·(1 - |(H mod 2)-1|).
5. if      0 ≤ H < 1 then [r g b] = [C X 0]
   else if 1 ≤ H < 2 then [r g b] = [X C 0]
   else if 2 ≤ H < 3 then [r g b] = [0 C X]
   else if 3 ≤ H < 4 then [r g b] = [0 X C]
   else if 4 ≤ H < 5 then [r g b] = [X 0 C]
   else if 5 ≤ H < 6 then [r g b] = [C 0 X]
   else [r g b] = [0 0 0]
6. [r g b] = 255*[r+D g+D b+D]
```

h, s, & v are numbers here not images.

Reference: HSL and HSV - Wikipedia, the free encyclopedia

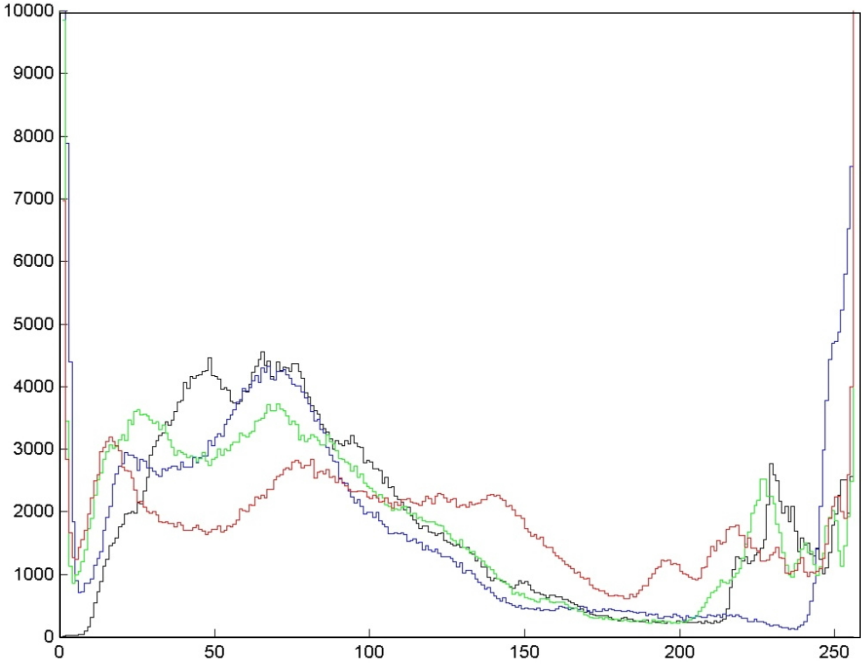
Saturation Adjustment

original



Saturation Adjustment

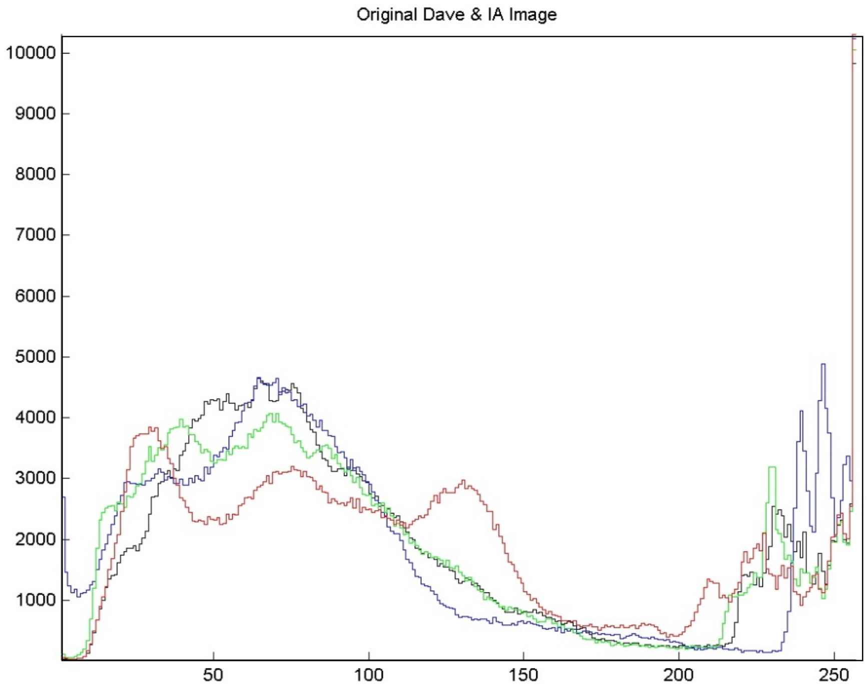
saturation + 50%



All the colors become closer to pure primaries.

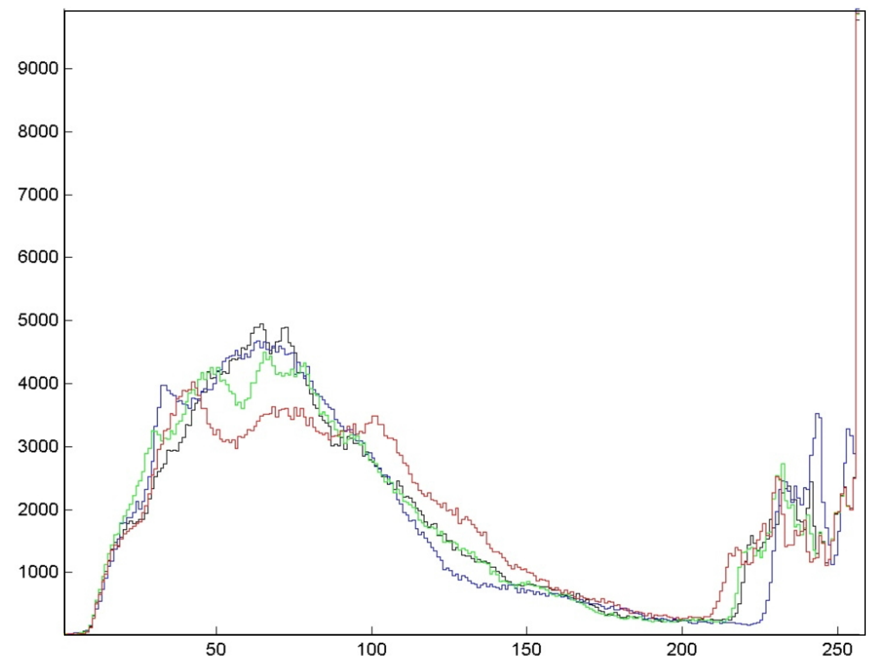
Saturation Adjustment

original



Saturation Adjustment

saturation - 50%

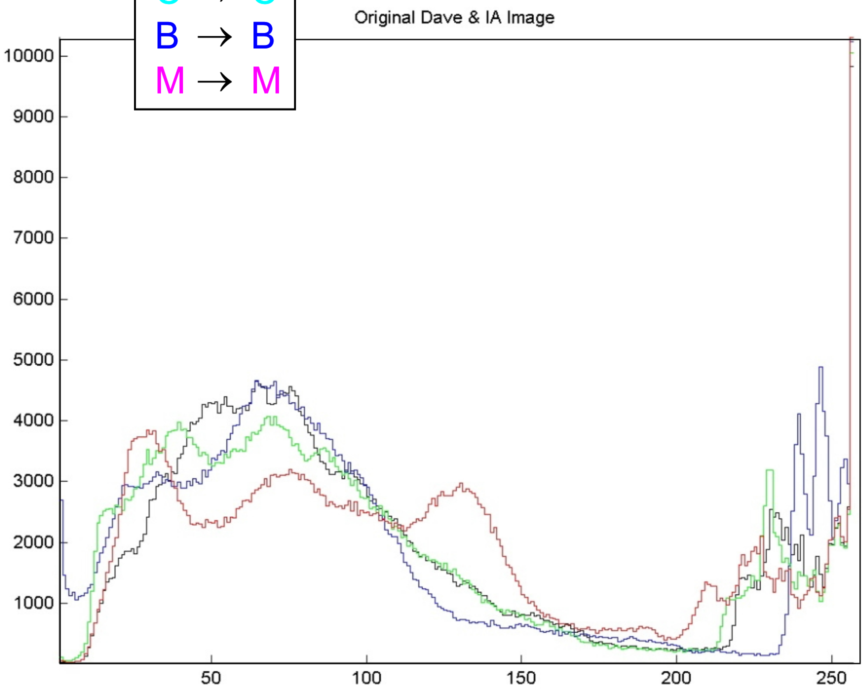


The r, g, & b histograms approach the value histogram as the color fades to grayscale.

Hue Shifting

original

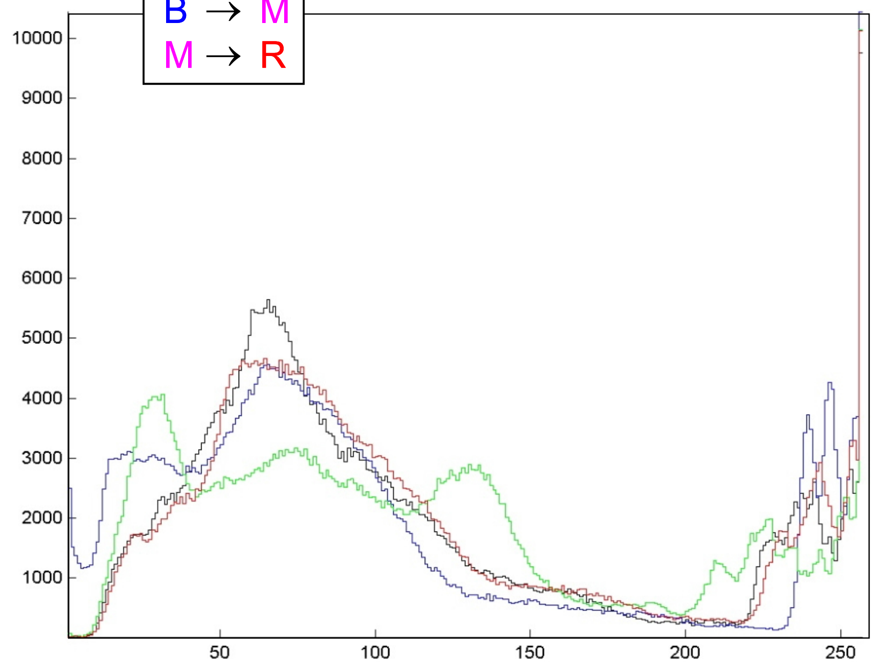
R → R
Y → Y
G → G
C → C
B → B
M → M



Hue Shifting

R → Y
Y → G
G → C
C → B
B → M
M → R

hue + 60°

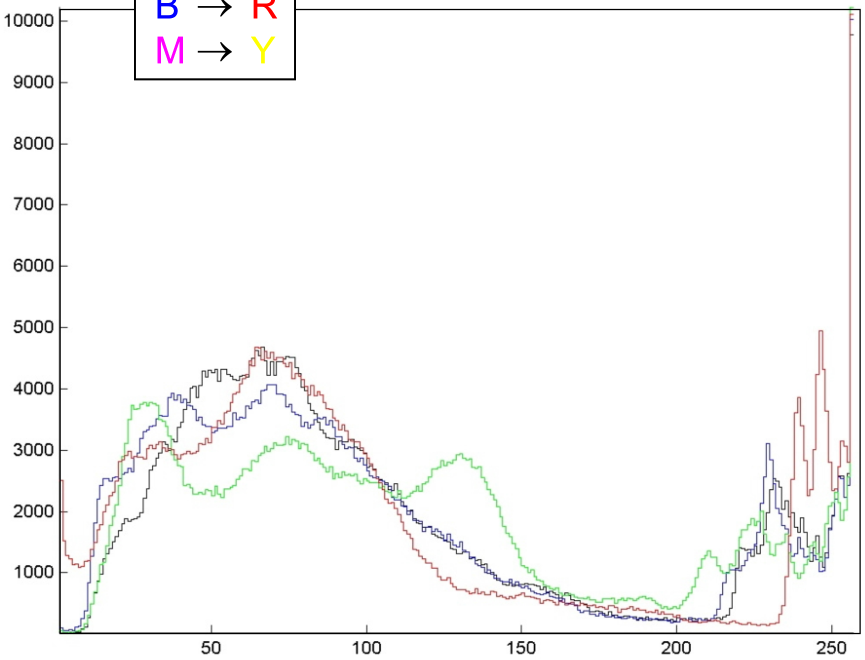


The effects of a hue shift are nonlinear. They difficult to characterize on the r, g, & b histograms

Hue Shifting

R → G
Y → C
G → B
C → M
B → R
M → Y

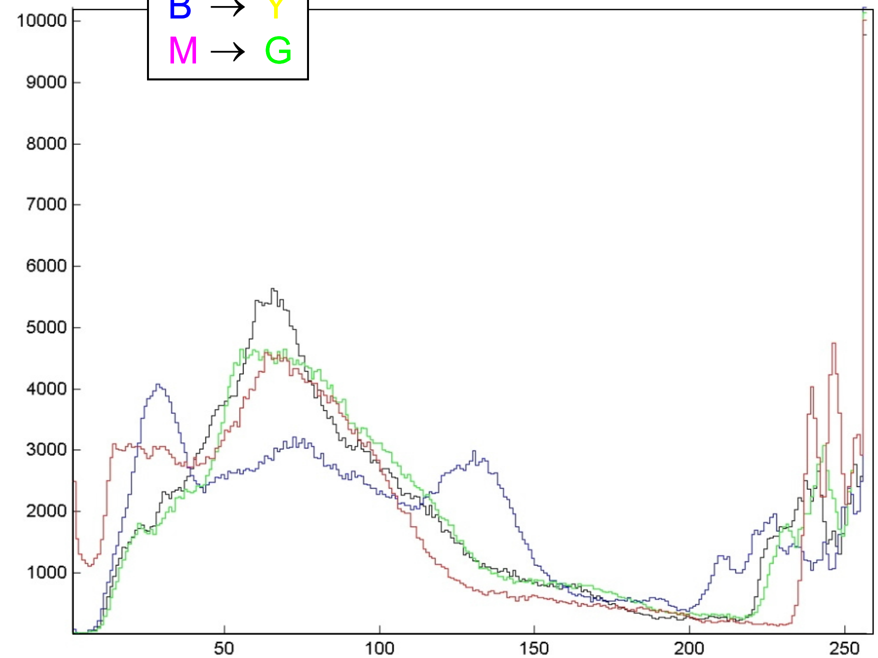
hue + 120°



Hue Shifting

R → C
Y → B
G → M
C → R
B → Y
M → G

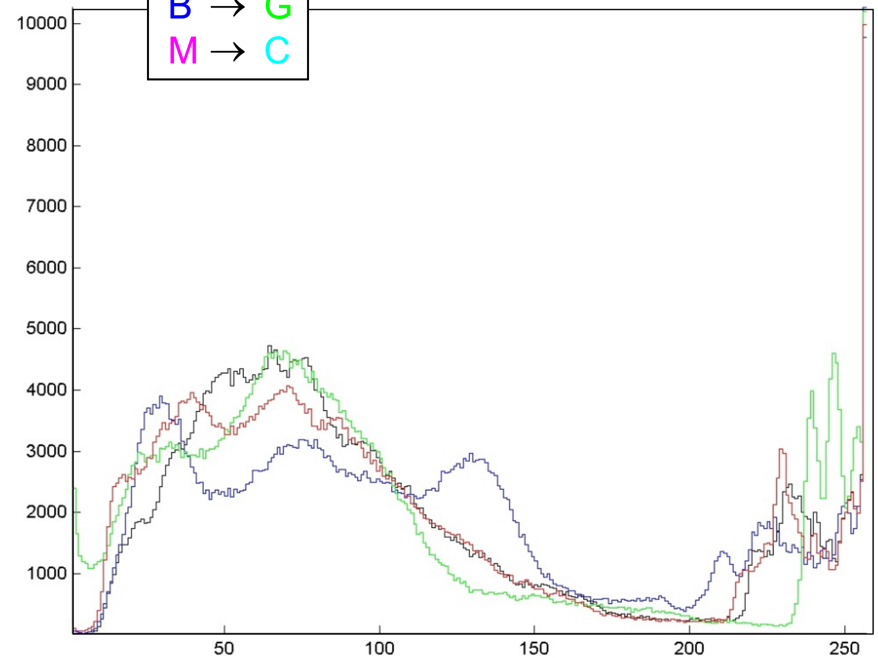
hue + 180°



Hue Shifting

R → B
Y → M
G → R
C → Y
B → G
M → C

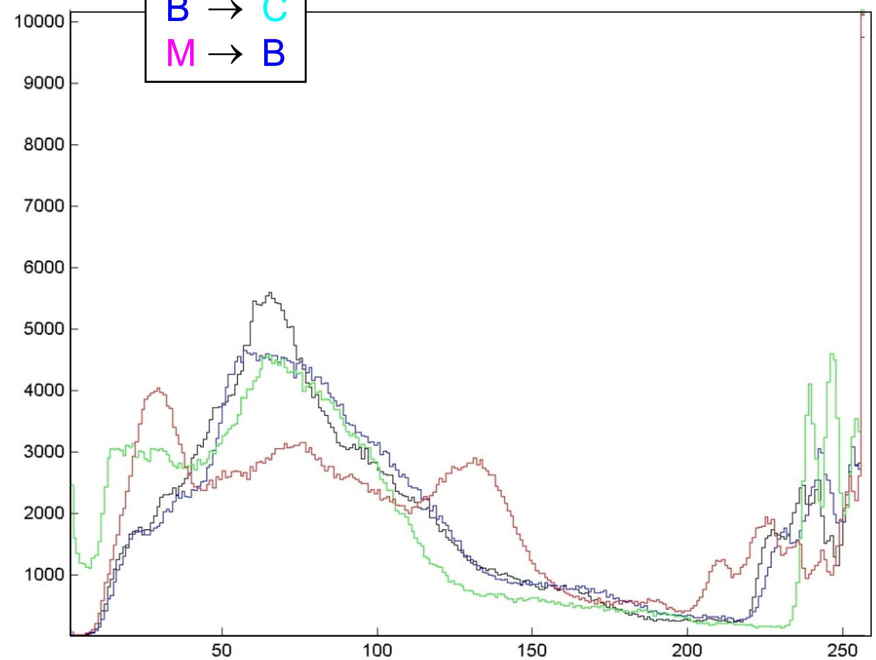
hue + 240°



Hue Shifting

R → M
Y → R
G → Y
C → G
B → C
M → B

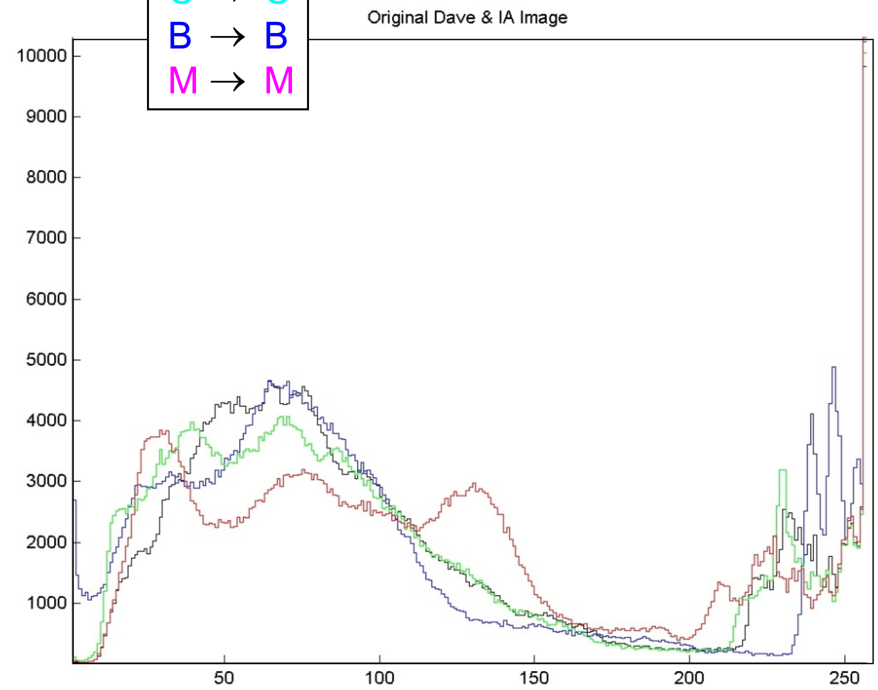
hue + 300°



Hue Shifting

R	→	R
Y	→	Y
G	→	G
C	→	C
B	→	B
M	→	M

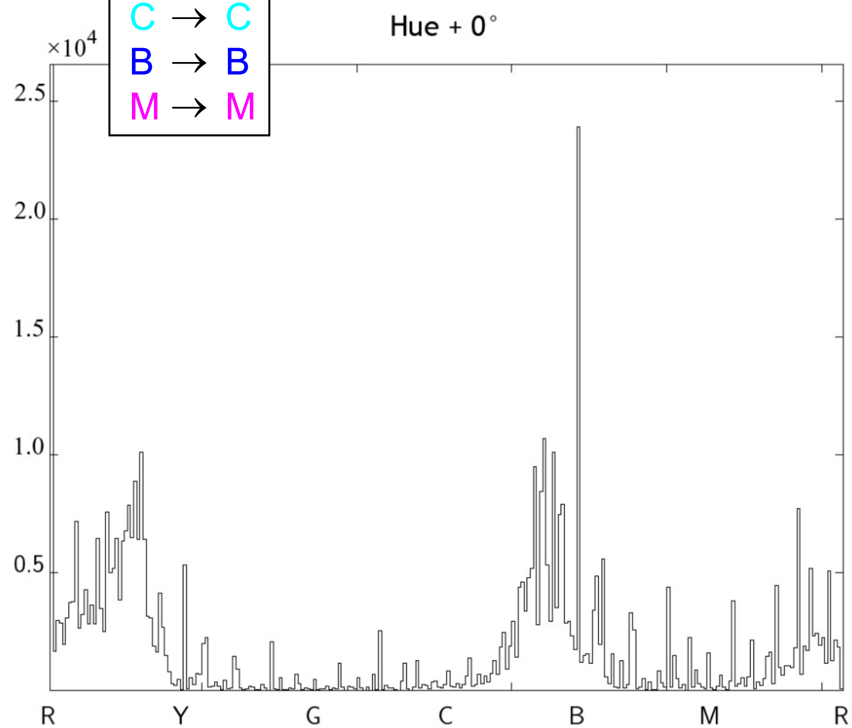
hue + 360° = original



Hue Shifting

R → R
Y → Y
G → G
C → C
B → B
M → M

original

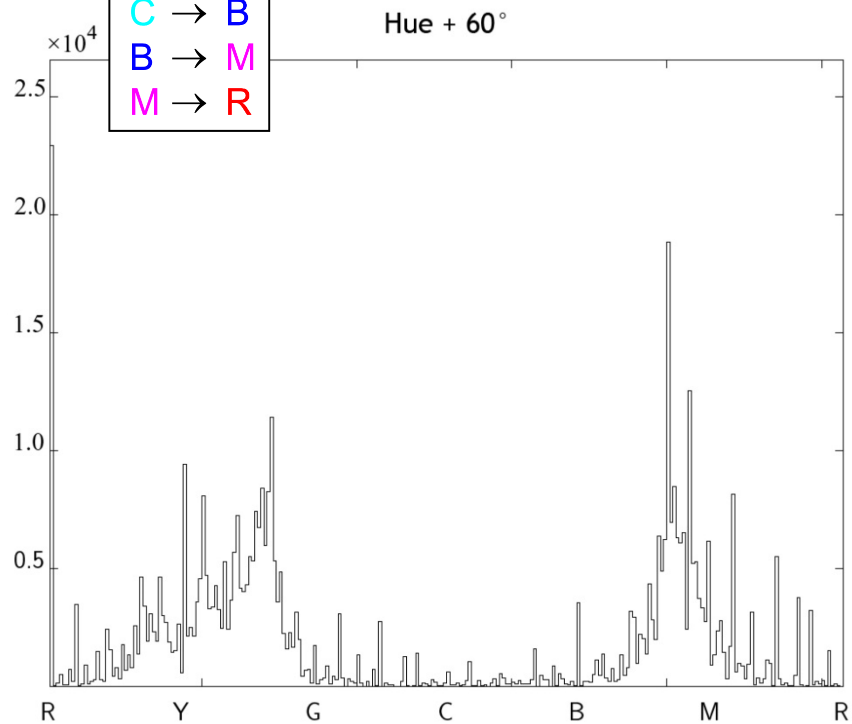


The effect of a hue shift on the hue histogram is quite obvious…

Hue Shifting

R → Y
Y → G
G → C
C → B
B → M
M → R

hue + 60°

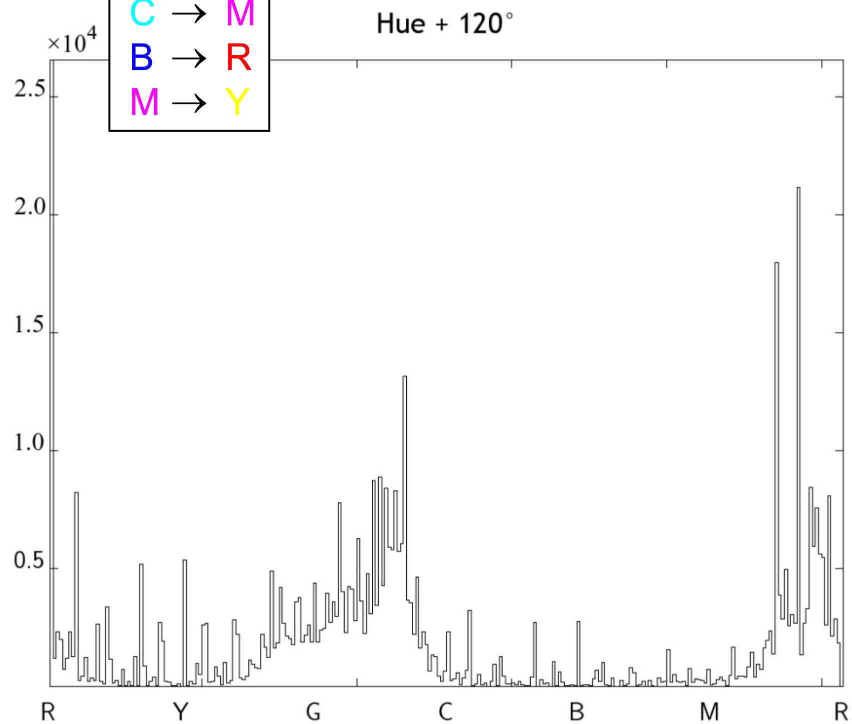


… the entire histogram is shifting…

Hue Shifting

R → G
Y → C
G → B
C → M
B → R
M → Y

hue + 120°

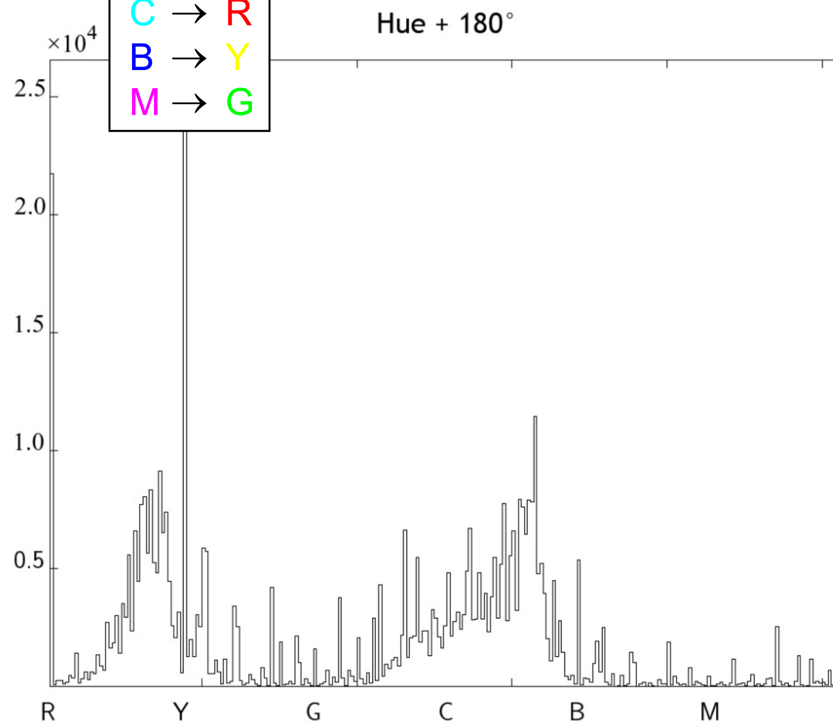


… and the shift is circular since the hue is a circular function – it is defined on a circle.

Hue Shifting

R → C
Y → B
G → M
C → R
B → Y
M → G

hue + 180°

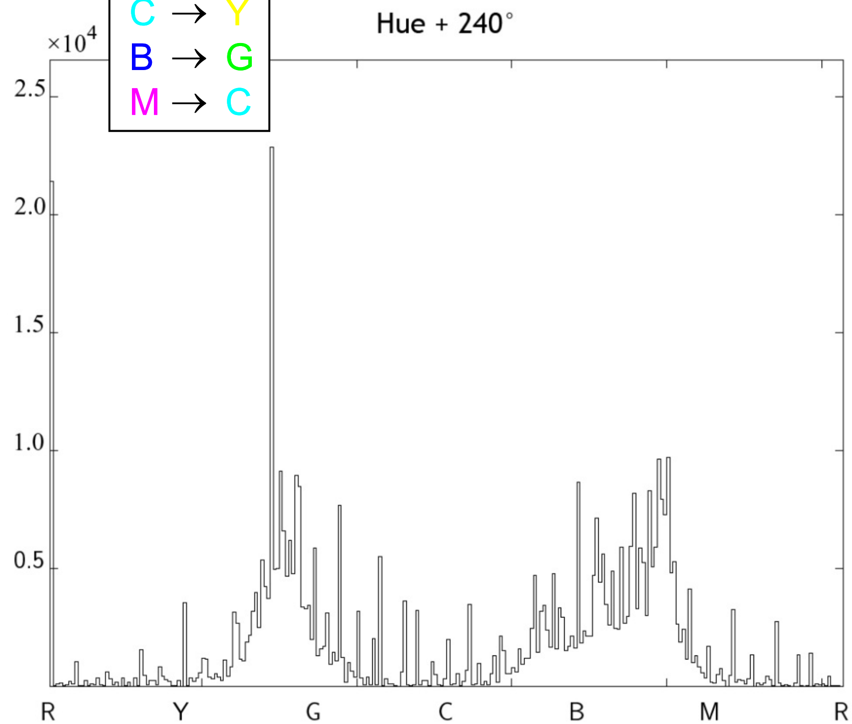


The part of the histogram that leaves one side appears on the other.

Hue Shifting

R → B
Y → M
G → R
C → Y
B → G
M → C

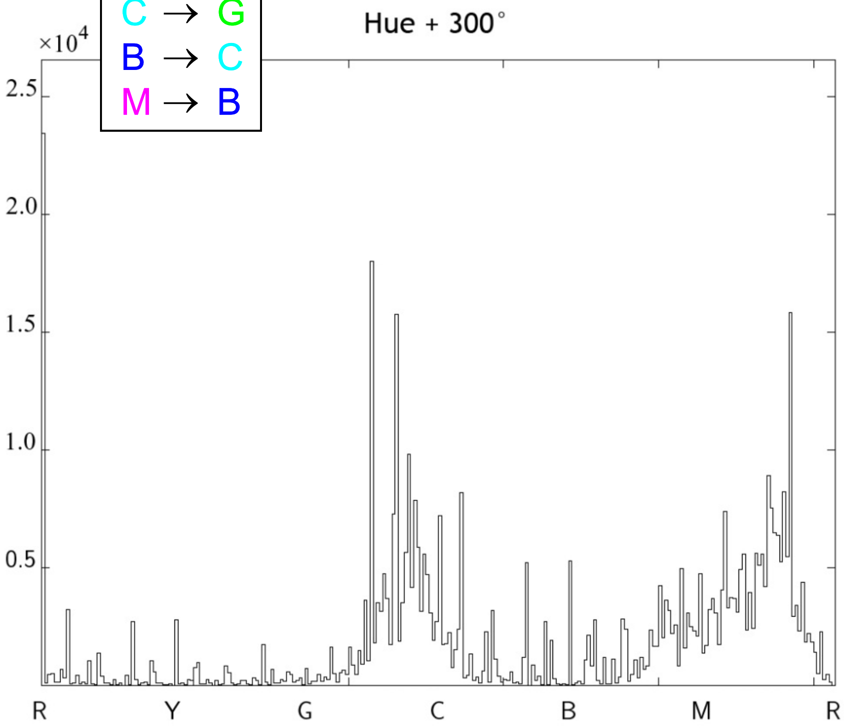
hue + 240°



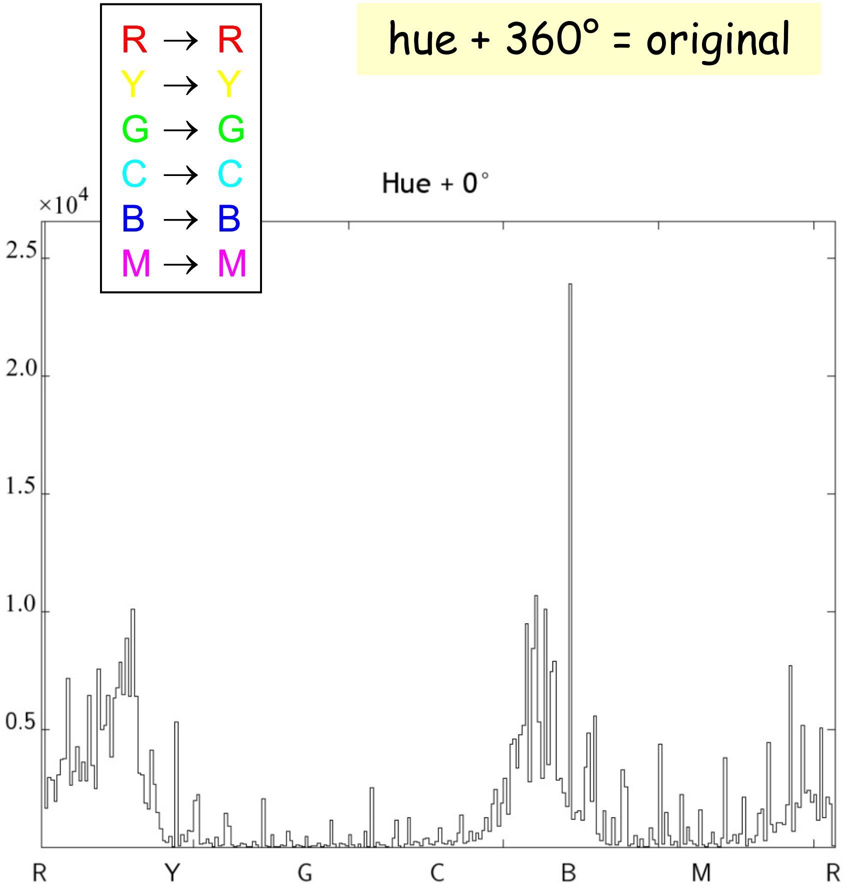
Hue Shifting

R → M
Y → R
G → Y
C → G
B → C
M → B

hue + 300°



Hue Shifting



Linear Transformation of Color

Color Correction via Linear Transformation

is a point process; the transformation is applied to each pixel as a function of its color alone.

$$\mathbf{J}(r,c) = \Phi\left[\mathbf{I}(r,c)\right], \quad \forall (r,c) \in \operatorname{supp}(\mathbf{I}).$$

Each pixel is vector valued, therefore the transformation is a vector space operator.

$$\mathbf{I}(r,c) = \begin{bmatrix} \mathbf{R}_{\mathbf{I}}(r,c) \\ \mathbf{G}_{\mathbf{I}}(r,c) \\ \mathbf{B}_{\mathbf{I}}(r,c) \end{bmatrix}, \ \mathbf{J}(r,c) = \begin{bmatrix} \mathbf{R}_{\mathbf{J}}(r,c) \\ \mathbf{G}_{\mathbf{J}}(r,c) \\ \mathbf{B}_{\mathbf{J}}(r,c) \end{bmatrix} = \Phi\left\{\mathbf{I}(r,c)\right\} = \Phi\left\{\begin{bmatrix} \mathbf{R}_{\mathbf{I}}(r,c) \\ \mathbf{G}_{\mathbf{I}}(r,c) \\ \mathbf{B}_{\mathbf{I}}(r,c) \end{bmatrix}\right\}.$$

Color Vector Space Operators

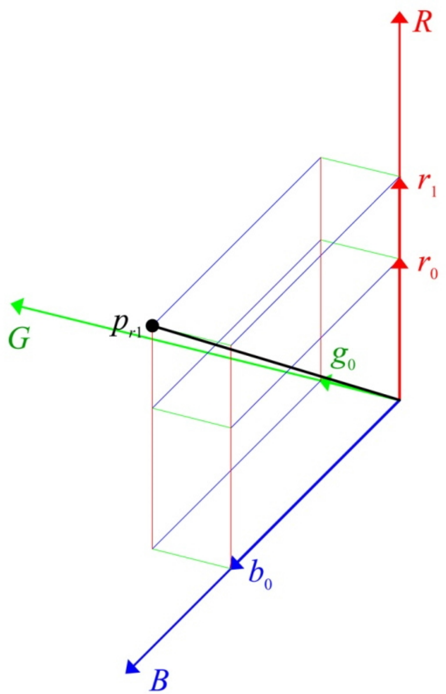
Linear operators are matrix multiplications

$$\begin{bmatrix} r_1 \\ g_1 \\ b_1 \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} \begin{bmatrix} r_0 \\ g_0 \\ b_0 \end{bmatrix}$$

$$\begin{bmatrix} r_1 \\ g_1 \\ b_1 \end{bmatrix} = 255 \cdot \begin{bmatrix} \left(r_0 / 255\right)^{1/\gamma_r} \\ \left(g_0 / 255\right)^{1/\gamma_g} \\ \left(b_0 / 255\right)^{1/\gamma_b} \end{bmatrix}$$

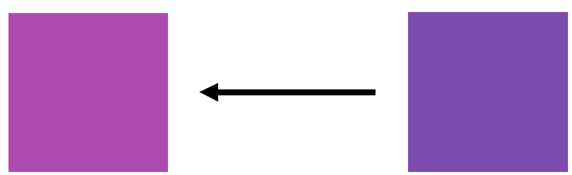
Example of a nonlinear operator: gamma correction

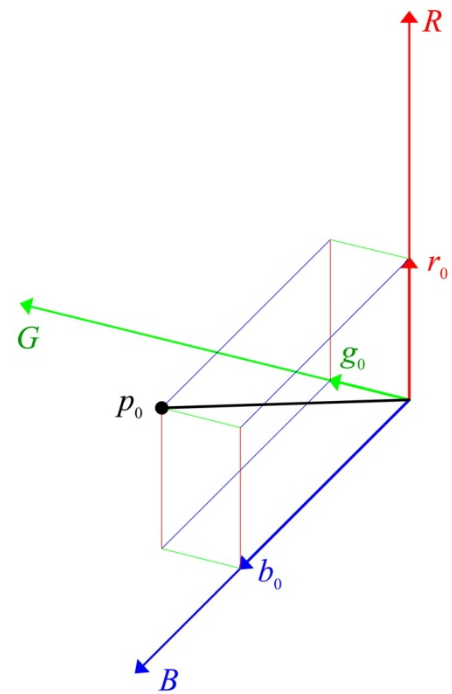
Linear Transformation of Color



$$\begin{bmatrix} r_1 \\ g_0 \\ b_0 \end{bmatrix} = \begin{bmatrix} r_1/r_0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} r_0 \\ g_0 \\ b_0 \end{bmatrix}$$

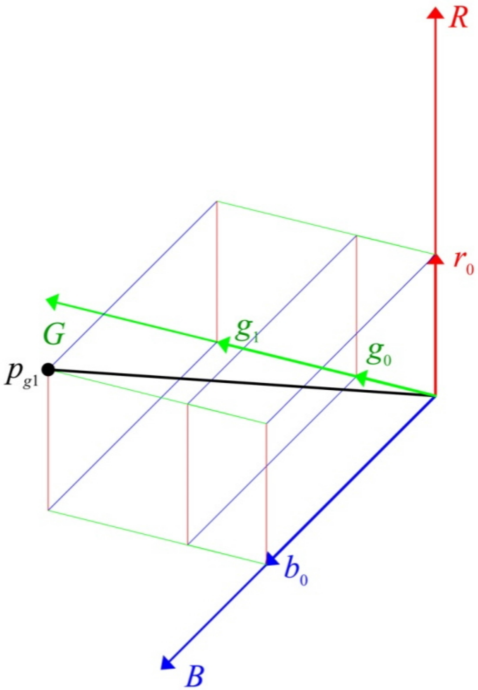
$$\begin{bmatrix} 175 \\ 75 \\ 175 \end{bmatrix} \longleftarrow \begin{bmatrix} 125 \\ 75 \\ 175 \end{bmatrix}$$



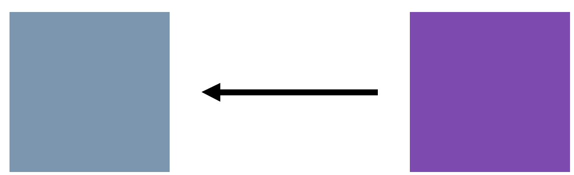


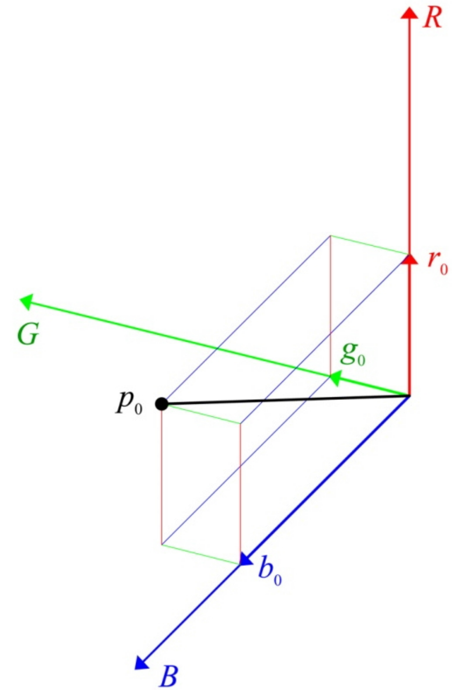
Linear Transformation of Color

$$\begin{bmatrix} r_0 \\ g_1 \\ b_0 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & g_1/g_0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} r_0 \\ g_0 \\ b_0 \end{bmatrix}$$

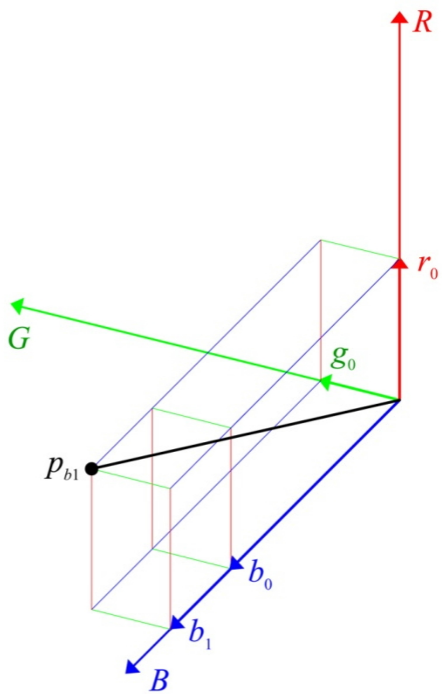


$$\begin{bmatrix} 125 \\ 150 \\ 175 \end{bmatrix} \leftarrow \begin{bmatrix} 125 \\ 75 \\ 175 \end{bmatrix}$$



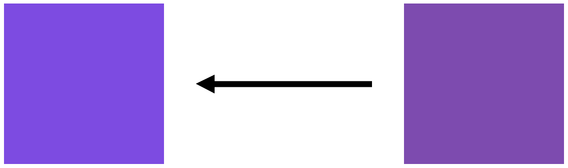


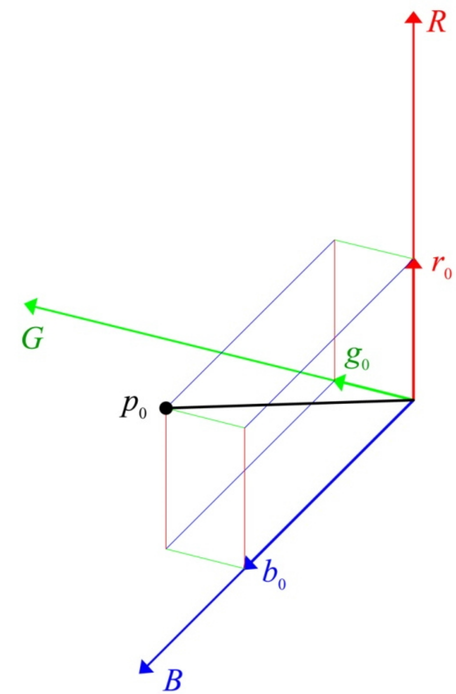
Linear Transformation of Color



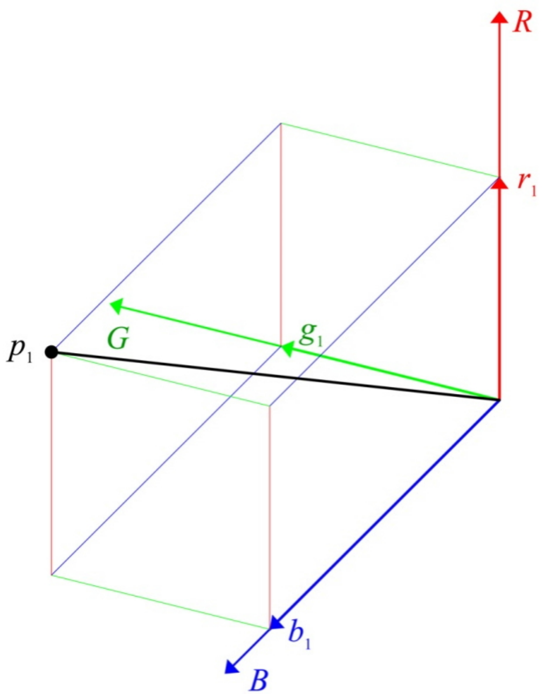
$$\begin{bmatrix} r_0 \\ g_0 \\ b_1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & b_1/b_0 \end{bmatrix} \begin{bmatrix} r_0 \\ g_0 \\ b_0 \end{bmatrix}$$

$$\begin{bmatrix} 125 \\ 75 \\ 225 \end{bmatrix} \longleftarrow \begin{bmatrix} 125 \\ 75 \\ 175 \end{bmatrix}$$



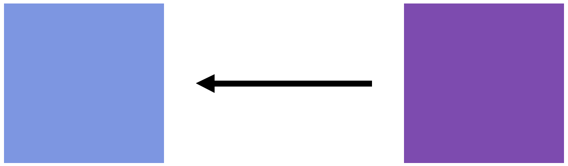


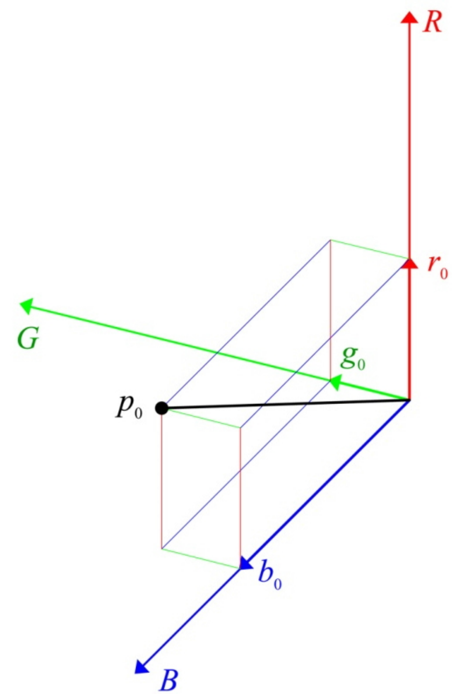
Linear Transformation of Color



$$\begin{bmatrix} r_1 \\ g_1 \\ b_1 \end{bmatrix} = \begin{bmatrix} r_1/r_0 & 0 & 0 \\ 0 & g_1/g_0 & 0 \\ 0 & 0 & b_1/b_0 \end{bmatrix} \begin{bmatrix} r_0 \\ g_0 \\ b_0 \end{bmatrix}$$

$$\begin{bmatrix} 175 \\ 150 \\ 225 \end{bmatrix} \longleftarrow \begin{bmatrix} 125 \\ 75 \\ 175 \end{bmatrix}$$





Color Transformation

Assume **J** is a discolored version of image **I** such that $\mathbf{J} = \Phi[\,\mathbf{I}\,]$. If Φ is linear then it is represented by a 3×3 matrix, **A**:

$$\mathbf{A} = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix}.$$

Then $\mathbf{J} = \mathbf{AI}$ or, more accurately, $\mathbf{J}(r,c) = \mathbf{AI}(r,c)$ for all pixel locations (r,c) in image **I**. $\mathbf{I}(r,c) \in \mathbb{Z}^3$.

Color Transformation

Each color in the output vector is a linear combination of the colors in the input vector.

If at pixel location (r,c),

image $\mathbf{I}(r,c) = \begin{bmatrix} \rho_{\mathbf{I}} \\ \gamma_{\mathbf{I}} \\ \beta_{\mathbf{I}} \end{bmatrix}$ and

image $\mathbf{J}(r,c) = \begin{bmatrix} \rho_{\mathbf{J}} \\ \gamma_{\mathbf{J}} \\ \beta_{\mathbf{J}} \end{bmatrix}$,

then $\mathbf{J}(r,c) = \mathbf{AI}(r,c)$, or

$$\begin{bmatrix} \rho_{\mathbf{J}} \\ \gamma_{\mathbf{J}} \\ \beta_{\mathbf{J}} \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} \begin{bmatrix} \rho_{\mathbf{I}} \\ \gamma_{\mathbf{I}} \\ \beta_{\mathbf{I}} \end{bmatrix}$$

$$= \begin{bmatrix} a_{11}\rho_{\mathbf{I}} & + & a_{12}\gamma_{\mathbf{I}} & + & a_{13}\beta_{\mathbf{I}} \\ a_{21}\rho_{\mathbf{I}} & + & a_{22}\gamma_{\mathbf{I}} & + & a_{23}\beta_{\mathbf{I}} \\ a_{31}\rho_{\mathbf{I}} & + & a_{32}\gamma_{\mathbf{I}} & + & a_{33}\beta_{\mathbf{I}} \end{bmatrix}.$$

Color Transformation

The inverse transform Φ^{-1} (if it exists) maps the discolored image, **J**, back into the correctly colored version, **I**, *i.e.*, $\mathbf{I} = \Phi^{-1}[\,\mathbf{J}\,]$. If Φ is linear then it is represented by the inverse of matrix **A**:

$$\mathbf{A}^{-1} = \left[a_{11}a_{22}a_{33} - a_{11}a_{23}a_{32} - a_{12}a_{21}a_{33} + a_{12}a_{23}a_{31} + a_{13}a_{21}a_{32} - a_{13}a_{22}a_{31}\right]^{-1} \bullet \begin{bmatrix} a_{22}a_{33} - a_{23}a_{32} & a_{13}a_{32} - a_{12}a_{33} & a_{12}a_{23} - a_{13}a_{22} \\ a_{23}a_{31} - a_{21}a_{33} & a_{11}a_{33} - a_{13}a_{31} & a_{13}a_{21} - a_{11}a_{23} \\ a_{21}a_{32} - a_{22}a_{31} & a_{12}a_{31} - a_{11}a_{32} & a_{11}a_{22} - a_{12}a_{21} \end{bmatrix}.$$

$\mathbf{A}^{-1}$ may or may not exist.

Color Correction

Assume we know n colors in the discolored image, **J**, that correspond to another set of n colors (that we also know) in the original image, **I**.

$$\left\{\begin{bmatrix}\rho_{\mathbf{J},k}\\ \gamma_{\mathbf{J},k}\\ \beta_{\mathbf{J},k}\end{bmatrix}\right\}_{k=1}^{n} \qquad \begin{bmatrix}\rho_{\mathbf{J},k}\\ \gamma_{\mathbf{J},k}\\ \beta_{\mathbf{J},k}\end{bmatrix} \leftrightarrow \begin{bmatrix}\rho_{\mathbf{I},k}\\ \gamma_{\mathbf{I},k}\\ \beta_{\mathbf{I},k}\end{bmatrix} \text{ for } k=1,\ldots,n. \qquad \left\{\begin{bmatrix}\rho_{\mathbf{I},k}\\ \gamma_{\mathbf{I},k}\\ \beta_{\mathbf{I},k}\end{bmatrix}\right\}_{k=1}^{n}$$

known wrong colors

known correspondence

known correct colors

Color Correction

To remap the discolored image so that the result matches the original image in a linearly optimal way, we need to find the matrix, **A**, that minimizes

$$\varepsilon^2 = \sum_{k=1}^{n} \left\| \begin{bmatrix} \rho_{\mathbf{I},k} \\ \gamma_{\mathbf{I},k} \\ \beta_{\mathbf{I},k} \end{bmatrix} - \mathbf{A}^{-1} \begin{bmatrix} \rho_{\mathbf{J},k} \\ \gamma_{\mathbf{J},k} \\ \beta_{\mathbf{J},k} \end{bmatrix} \right\|^2$$

Color Correction

To find the solution of this problem, let

$$\mathbf{Y} = \left[\begin{bmatrix} \rho_{\mathbf{I},1} \\ \gamma_{\mathbf{I},1} \\ \beta_{\mathbf{I},1} \end{bmatrix} \cdots \begin{bmatrix} \rho_{\mathbf{I},n} \\ \gamma_{\mathbf{I},n} \\ \beta_{\mathbf{I},n} \end{bmatrix} \right], \quad \text{and} \quad \mathbf{X} = \left[\begin{bmatrix} \rho_{\mathbf{J},1} \\ \gamma_{\mathbf{J},1} \\ \beta_{\mathbf{J},1} \end{bmatrix} \cdots \begin{bmatrix} \rho_{\mathbf{J},n} \\ \gamma_{\mathbf{J},n} \\ \beta_{\mathbf{J},n} \end{bmatrix} \right].$$

Then **X** and **Y** are known $3{\times}n$ matrices such that

$$\mathbf{Y} \approx \mathbf{A}^{-1}\mathbf{X},$$

where **A** is the 3×3 matrix that we want to find.

Color Correction

The linearly optimal solution is the least mean squared solution that is given by

$$\mathbf{B} = \mathbf{A}^{-1} = \mathbf{Y}\mathbf{X}^{\mathsf{T}}\left(\mathbf{X}\mathbf{X}^{\mathsf{T}}\right)^{-1}$$

where $\mathbf{X}^{\mathsf{T}}_{n\times 3}$ represents the transpose of matrix $\mathbf{X}_{3\times n}$.

.

Notes:
1. n, the number of color pairs, must be ≥ 3,
2. $[\mathbf{X}\mathbf{X}^{\mathsf{T}}]_{3\times 3}$ must be invertible, *i.e.*, $\text{rank}(\mathbf{X}\mathbf{X}^{\mathsf{T}}) = 3$,
3. If n=3, then $\mathbf{X}^{\mathsf{T}}(\mathbf{X}\mathbf{X}^{\mathsf{T}})^{-1} = \mathbf{X}^{-1}$.

important

Color Correction

input colors (to be changed):

$$\left[\begin{bmatrix}\rho_{\mathbf{J},1}\\ \gamma_{\mathbf{J},1}\\ \beta_{B,1}\end{bmatrix}\cdots\begin{bmatrix}\rho_{\mathbf{J},n}\\ \gamma_{\mathbf{J},n}\\ \beta_{\mathbf{J},n}\end{bmatrix}\right]\cdot\begin{bmatrix}\begin{bmatrix}\rho_{\mathbf{J},1} & \gamma_{\mathbf{J},1} & \beta_{\mathbf{J},1}\end{bmatrix}\\ \vdots \\ \begin{bmatrix}\rho_{\mathbf{J},n} & \gamma_{\mathbf{J},n} & \beta_{\mathbf{J},n}\end{bmatrix}\end{bmatrix}$$

The linearly optimal soluti… …uared solution that is given by

$$\mathbf{B} = \mathbf{A}^{-1} = \mathbf{Y}\mathbf{X}^{\mathsf{T}}\left(\mathbf{X}\mathbf{X}^{\mathsf{T}}\right)^{-1}$$

output colors (wanted):

$$\left[\begin{bmatrix}\rho_{\mathbf{I},1}\\ \gamma_{\mathbf{I},1}\\ \beta_{\mathbf{I},1}\end{bmatrix}\cdots\begin{bmatrix}\rho_{\mathbf{I},n}\\ \gamma_{\mathbf{I},n}\\ \beta_{\mathbf{I},n}\end{bmatrix}\right]$$

where $\mathbf{X}^{\mathsf{T}}_{n\times 3}$ repr… …se of matrix $\mathbf{X}_{3\times n}$.

Notes:
1. n, t… …r pairs, must be ≥ 3,
2. $[\mathbf{X}\mathbf{X}^{\mathsf{T}}]_{3\times 3}$ must be invertible, *i.e.*, rank($\mathbf{X}\mathbf{X}^{\mathsf{T}}$) = 3,
3. If n=3, then $\mathbf{X}^{\mathsf{T}}(\mathbf{X}\mathbf{X}^{\mathsf{T}})^{-1} = \mathbf{X}^{-1}$.

Color Correction

input colors (to be changed):

$$\left[\begin{bmatrix}\rho_{\mathbf{J},1}\\ \gamma_{\mathbf{J},1}\\ \beta_{B,1}\end{bmatrix}\cdots\begin{bmatrix}\rho_{\mathbf{J},n}\\ \gamma_{\mathbf{J},n}\\ \beta_{\mathbf{J},n}\end{bmatrix}\right]\cdot\begin{bmatrix}\left[\rho_{\mathbf{J},1}\quad\gamma_{\mathbf{J},1}\quad\beta_{\mathbf{J},1}\right]\\ \vdots\\ \left[\rho_{\mathbf{J},n}\quad\gamma_{\mathbf{J},n}\quad\beta_{\mathbf{J},n}\right]\end{bmatrix}$$

The linearly optimal soluti… …ared solution that is given by

$$\mathbf{B} = \mathbf{A}^{-1} = \mathbf{Y}\mathbf{X}^{\mathsf{T}}\left(\mathbf{X}\mathbf{X}^{\mathsf{T}}\right)^{-1}$$

output colors (wanted):

$$\left[\begin{bmatrix}\rho_{\mathbf{I},1}\\ \gamma_{\mathbf{I},1}\\ \beta_{\mathbf{I},1}\end{bmatrix}\cdots\begin{bmatrix}\rho_{\mathbf{I},n}\\ \gamma_{\mathbf{I},n}\\ \beta_{\mathbf{I},n}\end{bmatrix}\right]$$

where $\mathbf{X}^{\mathsf{T}}_{n\times 3}$ repr… …se of matrix $\mathbf{X}_{3\times n}$.

.

Notes:

1. n, t… …r pairs, must be ≥ 3,
2. $[\mathbf{X}\mathbf{X}^{\mathsf{T}}]_{3\times 3}$ must be invertible, *i.e.*, rank$(\mathbf{X}\mathbf{X}^{\mathsf{T}}) = 3$,
3. If n=3, then $\mathbf{X}^{\mathsf{T}}(\mathbf{X}\mathbf{X}^{\mathsf{T}})^{-1} = \mathbf{X}^{-1}$.

Color Correction

Then the image is color corrected by performing

$$\mathbf{I}(r,c) = \mathbf{B}\,\mathbf{J}(r,c), \text{ for all } (r,c) \in \text{supp}(\mathbf{J}).$$

In MATLAB this is easily performed by

```
>> I = reshape(((B*(reshape(double(J),R*C,3))')'),R,C,3);
>> m = min(I(:));
>> M = max(I(:));
>> I = uint8(255*(I-m)/(M-m));
```

where $\mathbf{B}=\mathbf{A}^{-1}$ is computed directly through the LMS formula on the previous page, and *R* & *C* are the number of rows and columns in the image.

Color Correction

Then the image is color corrected by performing

$$\mathbf{I}(r,c) = \mathbf{B}\,\mathbf{J}(r,c)\quad\ldots\quad).$$

The first reshape must be as R*C rows by 3 columns. Then it must be transposed to be premultiplied by B. If you reshape it directly into a 3 by R*C matrix, it will not work.

In MATLAB this is easil

```
>> I = reshape(((B*(reshape(double(J),R*C,3))')'),R,C,3);
>> m = min(I(:));
>> M = max(I(:));
>> I = uint8(255*(I-m)/(M-
```

After the matrix multiply is done, the result must be transposed again to R*C rows by 3 columns. Then it can be reshaped to R by C by 3.

where $\mathbf{B}=\mathbf{A}^{-1}$ is computed dir … rmula on the previous page, and R & C are the number of rows and columns in the image.

Color Correction

Then the image is color corrected by performing

$$\mathbf{I}(r,c) = \mathbf{B}\,\mathbf{J}(r,c), \text{ for all } (r,c) \in \text{supp}(\mathbf{J}).$$

In MATLAB this is easily performed by

```
>> I = reshape(((B*(reshape(double(J),R*C,3))')'),R,C,3);
>> m = min(I(:));
>> M = max(I(:));
>> I = uint8(255*(I-m)/(M-m));
```

Depending on the image, you might get better results if you directly convert I to uint8 rather than scaling it first. Try both, and select the version that looks best.

…ted directly through the LMS formula …nd *R* & *C* are the number of rows and

Linear Color Correction

NASA Summer Faculty Fellows at Ellington Air Force Base, Houston, TX, July 2002. Airplane is a T-38.

Original Image

“Aged” Image

Color Mapping 1

Original Image

"Aged" Image

Color Mapping 2

Original Image

“Aged” Image

Color Mapping 3

Original Image

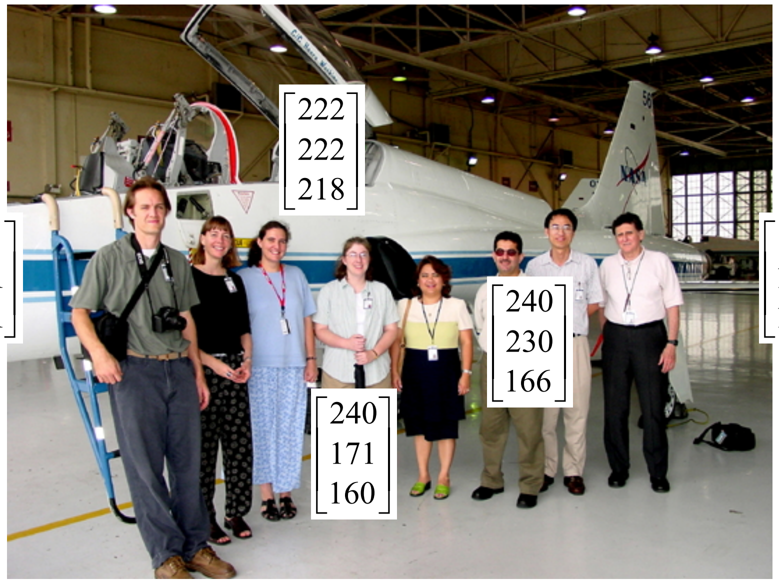
“Aged” Image

Color Mapping 4

Original Image

"Aged" Image

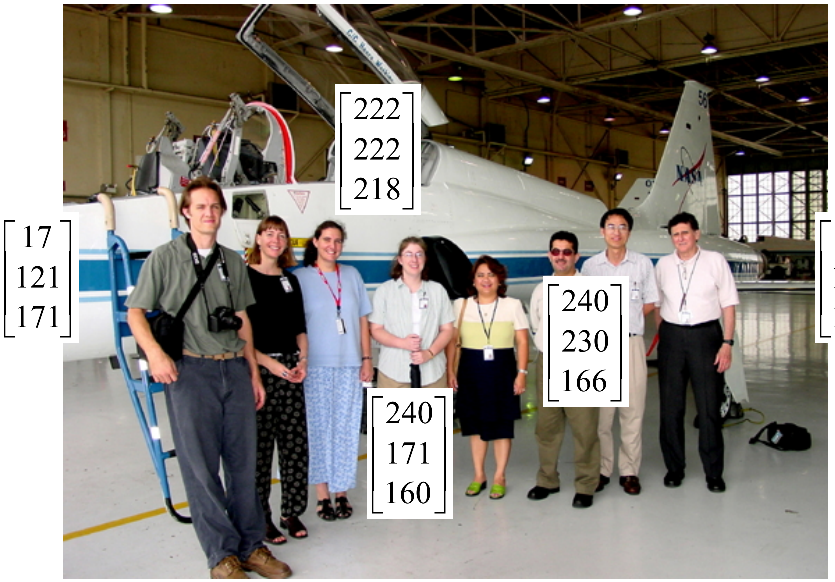
Color Transformations

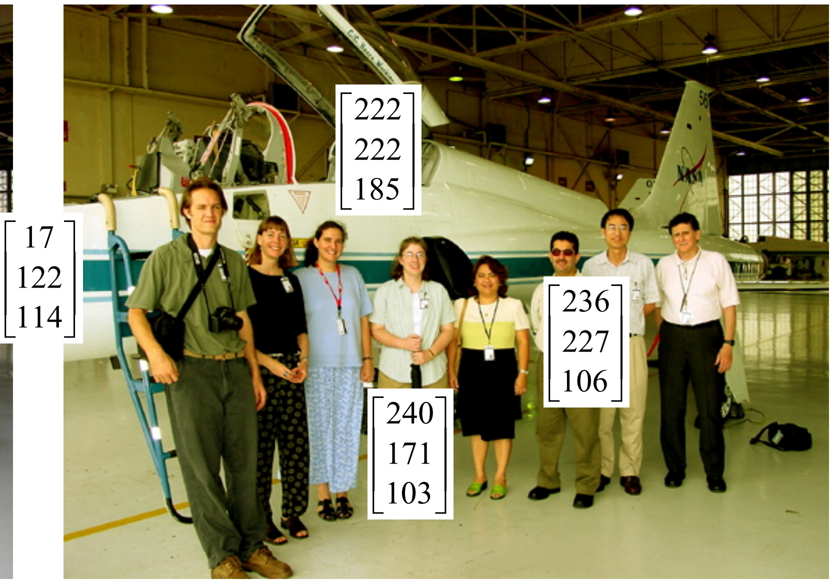


The aging process was a transformation, Φ, that mapped:

$$\begin{bmatrix}17\\122\\114\end{bmatrix}=\Phi\left\{\begin{bmatrix}17\\121\\171\end{bmatrix}\right\}\quad\begin{bmatrix}222\\222\\185\end{bmatrix}=\Phi\left\{\begin{bmatrix}222\\222\\218\end{bmatrix}\right\}\quad\begin{bmatrix}240\\171\\103\end{bmatrix}=\Phi\left\{\begin{bmatrix}240\\171\\160\end{bmatrix}\right\}\quad\begin{bmatrix}236\\227\\106\end{bmatrix}=\Phi\left\{\begin{bmatrix}240\\230\\166\end{bmatrix}\right\}$$

Color Transformations





To undo the process we need to find, Φ^{-1}, that maps:

$$\begin{bmatrix}17\\121\\171\end{bmatrix} = \Phi^{-1}\left\{\begin{bmatrix}17\\122\\114\end{bmatrix}\right\} \quad \begin{bmatrix}222\\222\\218\end{bmatrix} = \Phi^{-1}\left\{\begin{bmatrix}222\\222\\185\end{bmatrix}\right\} \quad \begin{bmatrix}240\\171\\160\end{bmatrix} = \Phi^{-1}\left\{\begin{bmatrix}240\\171\\103\end{bmatrix}\right\} \quad \begin{bmatrix}240\\230\\166\end{bmatrix} = \Phi^{-1}\left\{\begin{bmatrix}236\\227\\106\end{bmatrix}\right\}$$

Correction Using 3 Mappings

$$\mathbf{B} = \mathbf{A}^{-1} = \mathbf{YX}^{-1}$$

$$\mathbf{X} = \begin{bmatrix} 222 & 17 & 240 \\ 222 & 122 & 171 \\ 185 & 114 & 103 \end{bmatrix}$$

$$\mathbf{Y} = \begin{bmatrix} 222 & 17 & 240 \\ 222 & 121 & 171 \\ 218 & 171 & 160 \end{bmatrix}$$

Correction Using 3 Mappings

$$\mathbf{B} = \mathbf{A}^{-1} = \mathbf{Y}\mathbf{X}^{-1}$$

original

$$\mathbf{X} = \begin{bmatrix} 222 & 17 & 240 \\ 222 & 122 & 171 \\ 185 & 114 & 103 \end{bmatrix}$$

corrected

$$\mathbf{Y} = \begin{bmatrix} 222 & 17 & 240 \\ 222 & 121 & 171 \\ 218 & 171 & 160 \end{bmatrix}$$

Another Correction Using 3 Mappings

$$\mathbf{B} = \mathbf{A}^{-1} = \mathbf{Y}\mathbf{X}^{-1}$$

$$\mathbf{X} = \begin{bmatrix} 222 & 17 & 236 \\ 222 & 122 & 227 \\ 185 & 114 & 106 \end{bmatrix}$$

$$\mathbf{Y} = \begin{bmatrix} 222 & 17 & 240 \\ 222 & 121 & 230 \\ 218 & 171 & 166 \end{bmatrix}$$

Another Correction Using 3 Mappings

$$\mathbf{B} = \mathbf{A}^{-1} = \mathbf{Y}\mathbf{X}^{-1}$$

original

$$\mathbf{X} = \begin{bmatrix} 222 & 17 & 236 \\ 222 & 122 & 227 \\ 185 & 114 & 106 \end{bmatrix}$$

corrected

$$\mathbf{Y} = \begin{bmatrix} 222 & 17 & 240 \\ 222 & 121 & 230 \\ 218 & 171 & 166 \end{bmatrix}$$

Correction Using All 4 Mappings

$$\mathbf{B} = \mathbf{A}^{-1} = \mathbf{Y}\mathbf{X}^{\mathsf{T}}\left(\mathbf{X}\mathbf{X}^{\mathsf{T}}\right)^{-1}$$

$$\mathbf{X} = \begin{bmatrix} 222 & 17 & 236 & 240 \\ 222 & 122 & 227 & 171 \\ 185 & 114 & 106 & 103 \end{bmatrix} \qquad \mathbf{Y} = \begin{bmatrix} 222 & 17 & 240 & 240 \\ 222 & 121 & 230 & 171 \\ 218 & 171 & 166 & 160 \end{bmatrix}$$

Correction Using All 4 Mappings

$$\mathbf{B} = \mathbf{A}^{-1} = \mathbf{Y}\mathbf{X}^{\mathsf{T}}\left(\mathbf{X}\mathbf{X}^{\mathsf{T}}\right)^{-1}$$

original

corrected

$$\mathbf{X} = \begin{bmatrix} 222 & 17 & 236 & 240 \\ 222 & 122 & 227 & 171 \\ 185 & 114 & 106 & 103 \end{bmatrix}$$

$$\mathbf{Y} = \begin{bmatrix} 222 & 17 & 240 & 240 \\ 222 & 121 & 230 & 171 \\ 218 & 171 & 166 & 160 \end{bmatrix}$$

Random Sampling of Color Values

```
>> rr = round(R*rand([1 n]));
>> rc = round(C*rand([1 n]));
>> idx = [rr;rc];
>> Y(:,1) = diag(I(rr,rc,1));
>> Y(:,2) = diag(I(rr,rc,2));
>> Y(:,3) = diag(I(rr,rc,3));
>> X(:,1) = diag(J(rr,rc,1));
>> X(:,2) = diag(J(rr,rc,2));
>> X(:,3) = diag(J(rr,rc,3));
```

R = number of rows in image
C = number of columns in image
n = number of pixels to select

rand([1 n]) : 1 × n matrix of random numbers between 0 and 1.

diag(I(rr,rc,1)): vector from main diagonal of matrix I(rr,rc,1).

Correction Using 128 Mappings

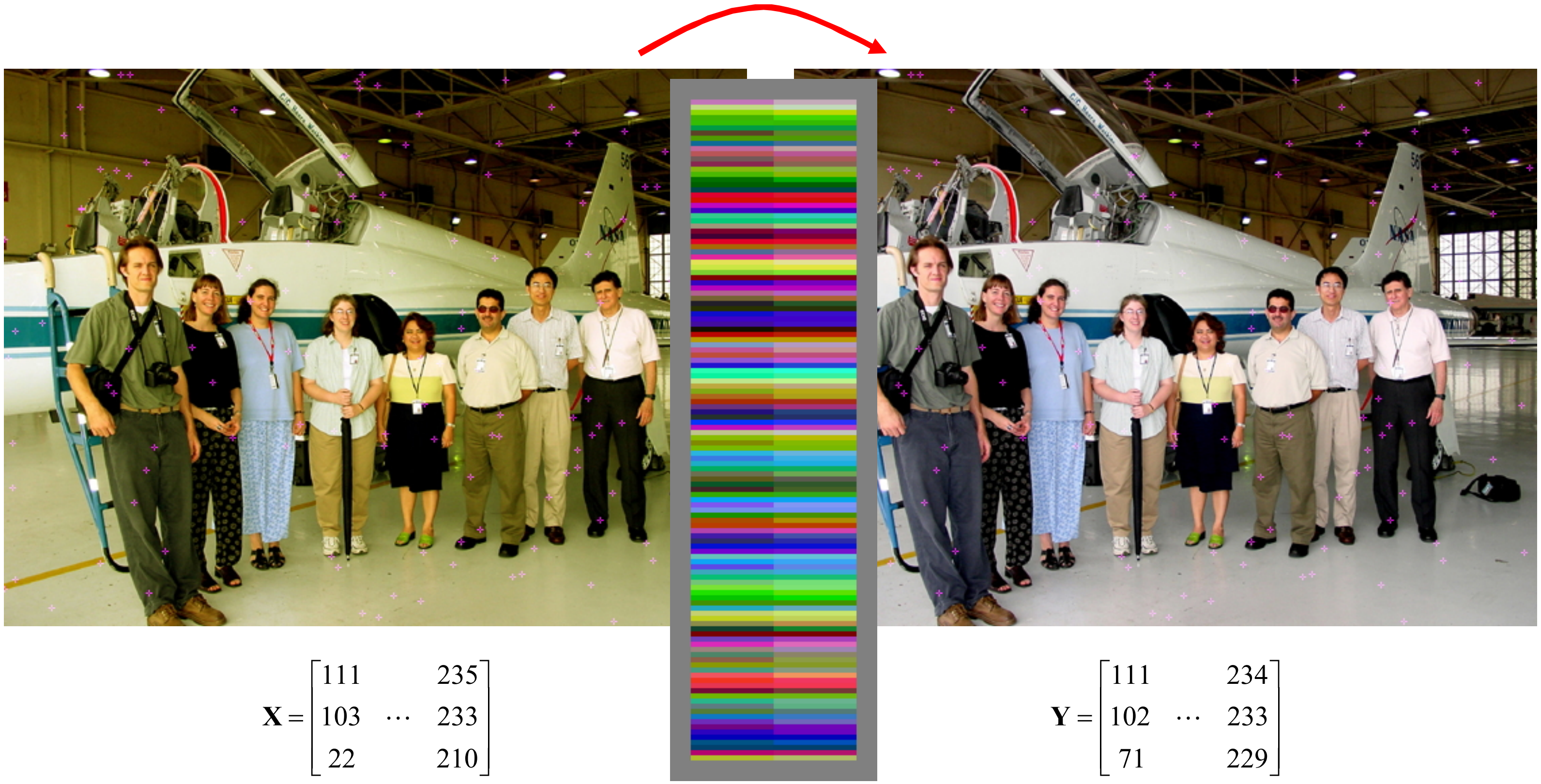
$$\mathbf{B} = \mathbf{A}^{-1} = \mathbf{Y}\mathbf{X}^{\mathsf{T}}\left(\mathbf{X}\mathbf{X}^{\mathsf{T}}\right)^{-1}$$

$$\mathbf{X} = \begin{bmatrix} 111 & & 235 \\ 103 & \cdots & 233 \\ 22 & & 210 \end{bmatrix}$$

$$\mathbf{Y} = \begin{bmatrix} 111 & & 234 \\ 102 & \cdots & 233 \\ 71 & & 229 \end{bmatrix}$$

Correction Using 128 Mappings

$$\mathbf{B} = \mathbf{A}^{-1} = \mathbf{Y}\mathbf{X}^{\mathsf{T}}\left(\mathbf{X}\mathbf{X}^{\mathsf{T}}\right)^{-1}$$



$$\mathbf{X} = \begin{bmatrix} 111 & & 235 \\ 103 & \cdots & 233 \\ 22 & & 210 \end{bmatrix} \qquad \mathbf{Y} = \begin{bmatrix} 111 & & 234 \\ 102 & \cdots & 233 \\ 71 & & 229 \end{bmatrix}$$

Correction Using 128 Mappings

$$\mathbf{B} = \mathbf{A}^{-1} = \mathbf{Y}\mathbf{X}^{\mathsf{T}}\left(\mathbf{X}\mathbf{X}^{\mathsf{T}}\right)^{-1}$$

original

$$\mathbf{X} = \begin{bmatrix} 111 & & 235 \\ 103 & \cdots & 233 \\ 22 & & 210 \end{bmatrix}$$

corrected

$$\mathbf{Y} = \begin{bmatrix} 111 & & 234 \\ 102 & \cdots & 233 \\ 71 & & 229 \end{bmatrix}$$

Correction Using 4 Mappings

$$\mathbf{B} = \mathbf{A}^{-1} = \mathbf{Y}\mathbf{X}^{\mathsf{T}}\left(\mathbf{X}\mathbf{X}^{\mathsf{T}}\right)^{-1}$$

original

corrected

$$\mathbf{X} = \begin{bmatrix} 222 & 17 & 236 & 240 \\ 222 & 122 & 227 & 171 \\ 185 & 114 & 106 & 103 \end{bmatrix}$$

$$\mathbf{Y} = \begin{bmatrix} 222 & 17 & 240 & 240 \\ 222 & 121 & 230 & 171 \\ 218 & 171 & 166 & 160 \end{bmatrix}$$

MATLAB Linear Color Transformation Function

```
function J = LinTrans(I,B)
    [R C D] = size(I);
    if D ~= 3
        error('Image must have 3 bands');
    end
    I = double(I);
    J = reshape(((B*(reshape(I,R*C,3))')'),R,C,3);
end;
```

This function returns an image of class double. To get a good uint8 you may have to linearly scale the result as shown on slide 55. Or not.