# EECE/CS 4353 Image Processing

## Lecture Notes: The Point Processing of Images

Richard Alan Peters II

Department of Electrical Engineering and Computer Science

Fall Semester 2016

# Point Processing of Images

- In a digital image, a point = a pixel.
- Point processing transforms a pixel's value as function of its value alone.
- It does not depend on the values of the pixel's neighbors.

# Point Processing of Images

Examples include:

- Brightness and contrast adjustment
- Gamma correction
- Histogram equalization
- Histogram matching
- Color correction.

# Point Processing

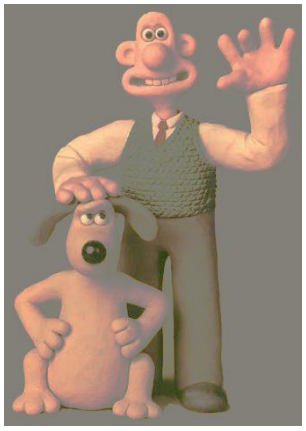
- gamma

- brightness

original

+ brightness

+ gamma



histogram mod

- contrast

original

+ contrast

histogram EQ

# Point Processing: Pixel Values

A point process transforms one intensity level (or color) into another as a function of that one alone. So a point process is

$$\mathbf{p}_{\text{out}} = f\left(\mathbf{p}_{\text{in}}\right).$$

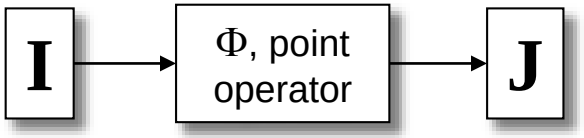
That is, the pixel value output is dependent on only the pixel value input. That implies

$$\mathbf{p}_{\text{out}}\left(r,c\right) = f\left(\mathbf{p}_{\text{in}}\left(r,c\right)\right).$$

In words, the output at one location is dependent only the value of the input image at that same location. Other locations don't matter.

# Point Ops via Functional Mappings

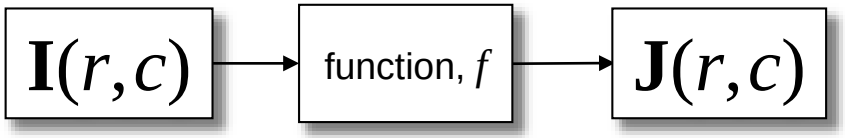
Image:




$$\mathbf{J} = \Phi[\mathbf{I}]$$

Input Output

Pixel:




If $\mathbf{I}(r,c,b) = g$
and $f(g) = k$
then $\mathbf{J}(r,c,b) = k.$

The transformation of image **I** into image **J** is accomplished by replacing each input intensity, *g*, with a specific output intensity, *k*, at every location (*r*,*c*,*b*) where **I**(*r*,*c*,*b*) = *g*.

The rule that associates *k* with *g* is usually specified with a function, *f*, so that *f* (*g*) = *k*.

# Point Ops via Functional Mappings

One-band Image

$$\mathbf{J}(r,c) = f\,(\,\mathbf{I}(r,c)\,),$$
for all pixel locations, $(r,c)$.

Three-band Image

$$\mathbf{J}(r,c,b) = f\,(\,\mathbf{I}(r,c,b)\,),\ \text{or}$$
$$\mathbf{J}(r,c,b) = f_b\,(\,\mathbf{I}(r,c,b)\,),\ \text{or}$$
for $b = 1, 2, 3$, and all $(r,c)$.

# Point Ops via Functional Mappings

One-band Image

$$\mathbf{J}(r,c) = f(\mathbf{I}(r,c)),$$
for all pixel locations, $(r,c)$.

Either all 3 bands are mapped through the same function, f, or …

Three-band Image

$$\mathbf{J}(r,c,b) = f(\mathbf{I}(r,c,b)), \text{ or}$$
$$\mathbf{J}(r,c,b) = f_b(\mathbf{I}(r,c,b)), \text{ or}$$
for $b = 1, 2, 3$, and all $(r,c)$

… each band is mapped through a separate function, $f_b$.

# Look-Up Tables

# Lookup Tables

A lookup table is an indexed list of numbers – a vector – that can be used to implement a discrete function – a mapping from a set of integers, $\{g_{\text{in},1}, g_{\text{in},2}, \ldots, g_{\text{in},n}\}$, to a set of numbers (integers or not), $\{g_{\text{out},1}, g_{\text{out},2}, \ldots, g_{\text{out},n}\}$. A lookup table can implement a function such as:

$$\text{if } g_{\text{out}} = f\left(g_{\text{in}}\right), \text{ where } g_{\text{in}} \in \{0,\ldots,n-1\} \text{ and } g_{\text{out}} \in \left\{ g_{\text{out},k} \right\}_{k=1}^{n}$$
$$\text{then define } \text{LUT}\left(g_{\text{in},k}+1\right) = \text{LUT}\left(k\right) = g_{\text{out},k}, \text{ for } k = 1,\ldots \text{n}.$$

We've already seen one of these, the *colormap*.[1]

[1] Lecture 2, slides 29-40.

# Point Operations using Lookup Tables

A lookup table (LUT) can implement a functional mapping.

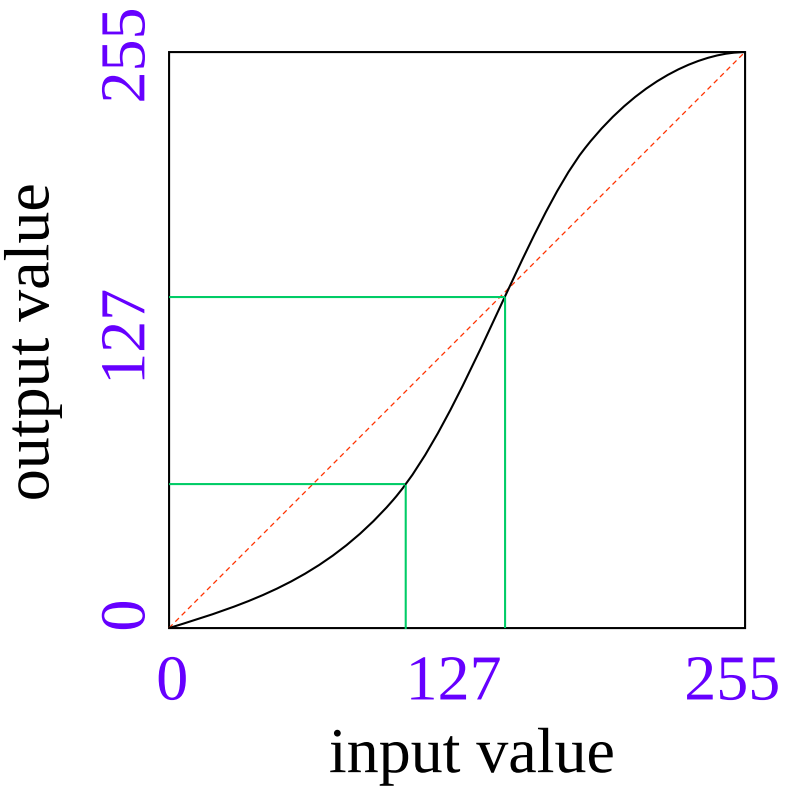
… then the LUT that implements $f$ is a 256x1 array whose $(g+1)^{th}$ value is $k = f(g)$.

If $k = f(g)$,
for $g = 0,\ldots,255$,
and if $k$ takes on values in $\{0,\ldots,255\}, \ldots$

LUT is 256x1. But I may be RxC or RxCx3.

To remap an image, **I**, to **J** : $\mathbf{J} = \mathrm{LUT}(\mathbf{I}+1)$

# Point Operations = Lookup Table Ops




| *E.g.*: | index | value |
|---|---|---|
| | ... | ... |
| | 101 | 64 |
| | 102 | 68 |
| | 103 | 69 |
| | 104 | 70 |
| | 105 | 70 |
| | 106 | 71 |
| | ... | ... |
| | input | output |

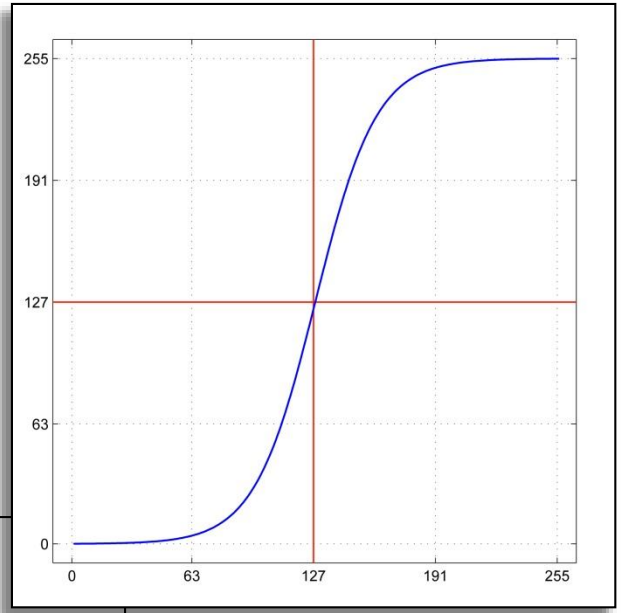
# How to Generate a Lookup Table

For example, a *sigmoid*:

Let $a = 2.$

Let $x \in \{0,\ldots,255\}$

$$\sigma(x;a) = \frac{255}{1+e^{-a(x-127)/32}}$$




Or in Matlab:

```
a = 2;
x = 0:255;
LUT = 255 ./ (1+exp(-a*(x-127)/32));
```

This is just one example.

# Point Ops on RGB Images using Lookup Tables

If **I** is 3-band, then

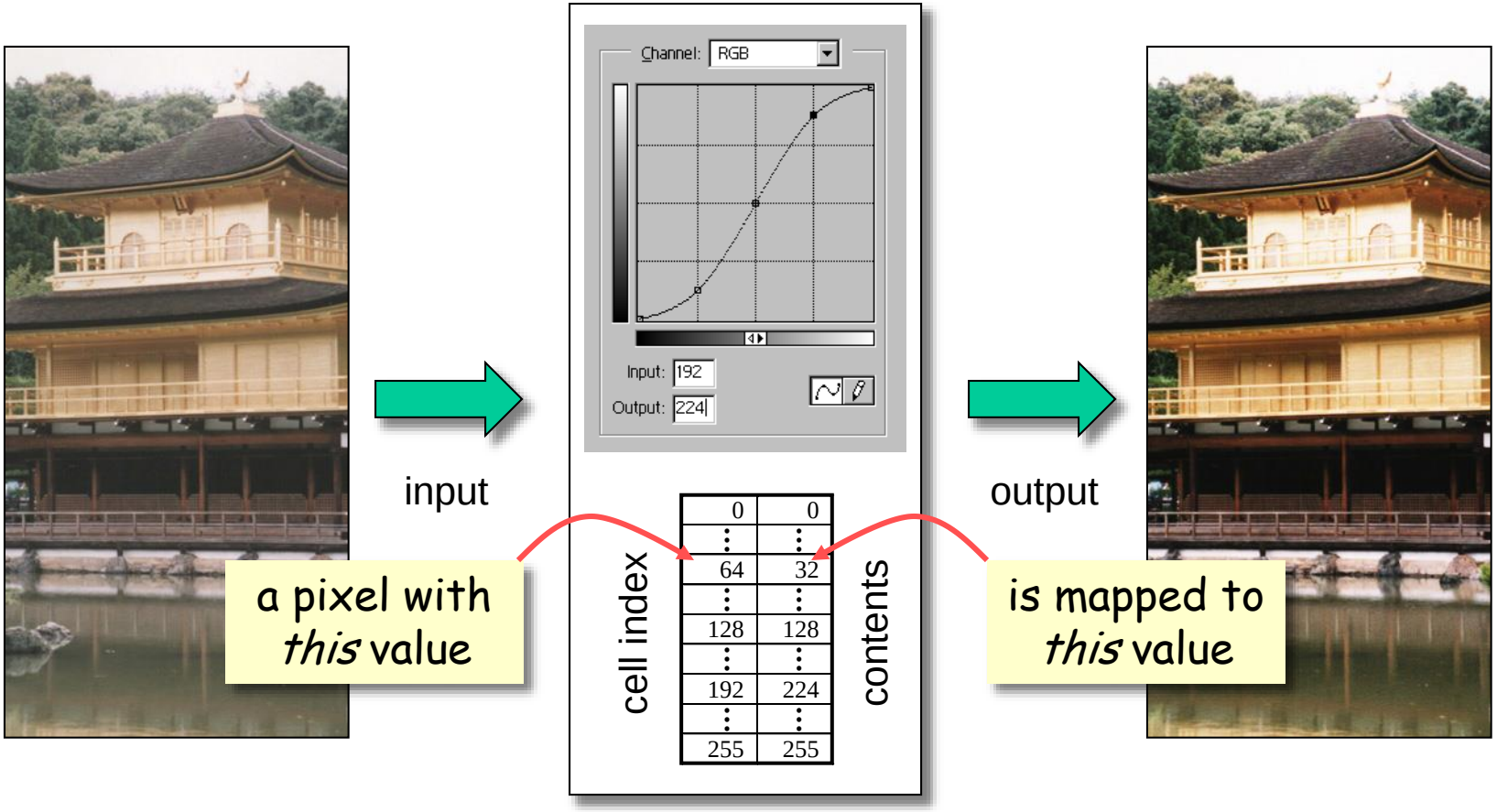
a) each band is mapped separately using the same LUT for each band *or*

b) each band is mapped using different LUTs – one for each band.

a) $\mathbf{J} = \mathrm{LUT}(\mathbf{I}+1)$,

b) $\mathbf{J}(:,:,b) = \mathrm{LUT}_b(\mathbf{I}(:,:,b)+1)$, for $b = 1, 2, 3$.

# 1-Band Lookup Table for 3-Band Image

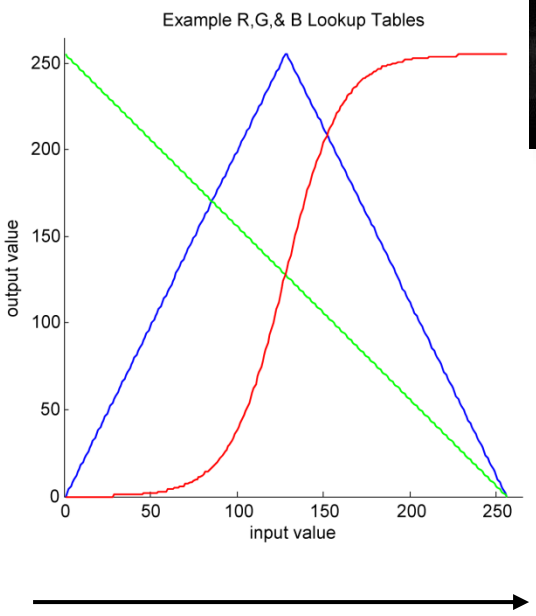


# Example 3-Band Image …

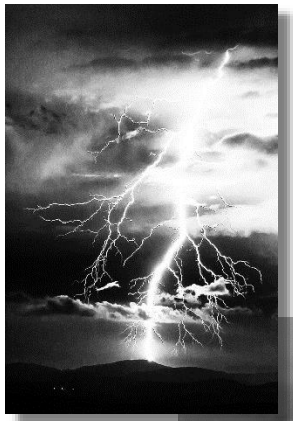
R, G, B Bands


Lightning at Ramasse, Rhone-Alpes, France by Flickr user, Regarde là-bas,
https://www.flickr.com/photos/marcel_s_s/8624344496/in/pool-tbasab/

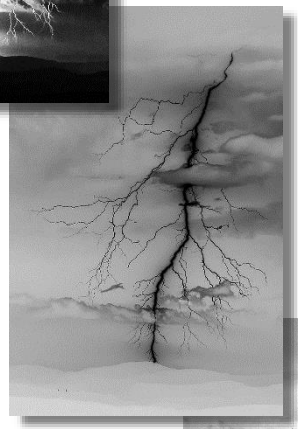
# … with 3-Band LUT.



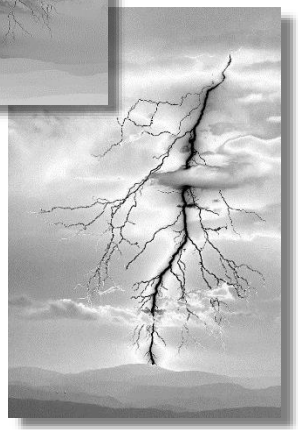

R



G



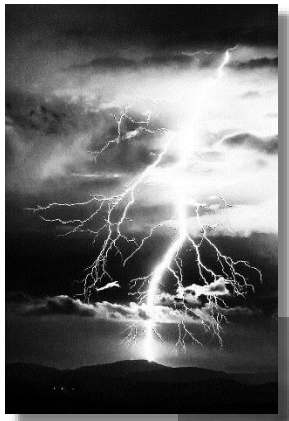
B



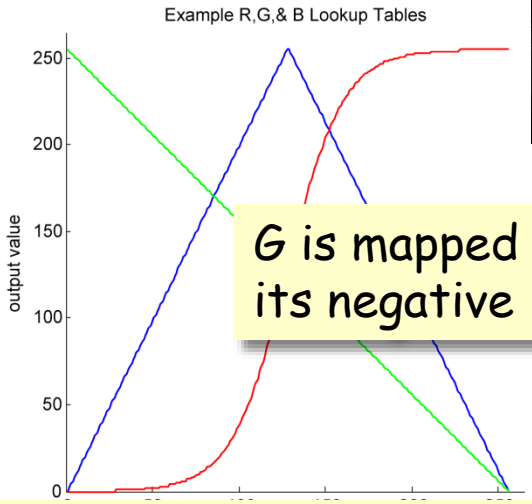
Lightning at Ramasse, Rhone-Alpes, France by Flickr user, Regarde là-bas, https://www.flickr.com/photos/marcel_s_s/8624344496/in/pool-tbasab/

# … with 3-Band LUT.

R

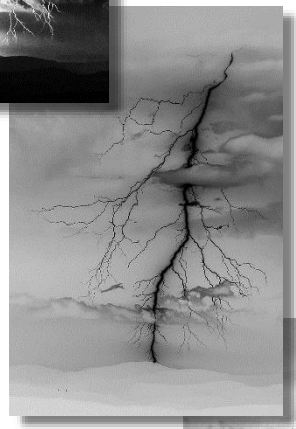


R is mapped through a sigmoid.



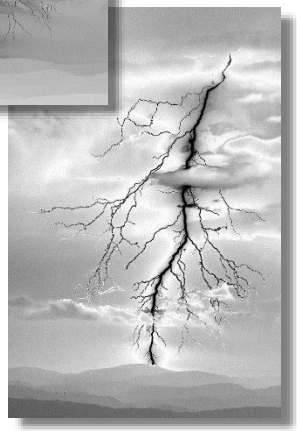

G is mapped to its negative

G



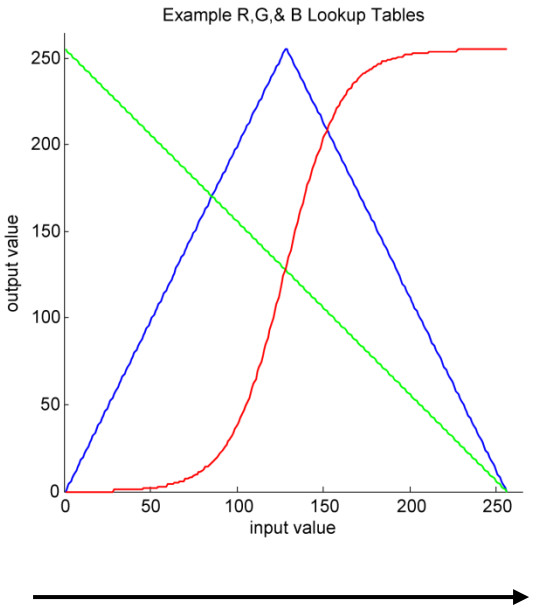
The dark values of B are contrast stretched while the light values are negated and stretched.

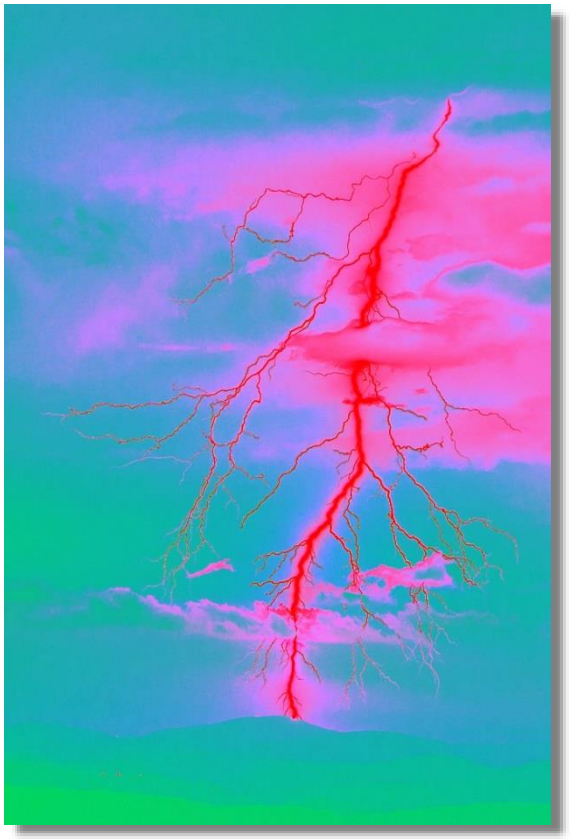
B



Lightning at Ramasse, Rhone-Alpes, France by Flickr user, Regarde là-bas, https://www.flickr.com/photos/marcel_s_s/8624344496/in/pool-tbasab/

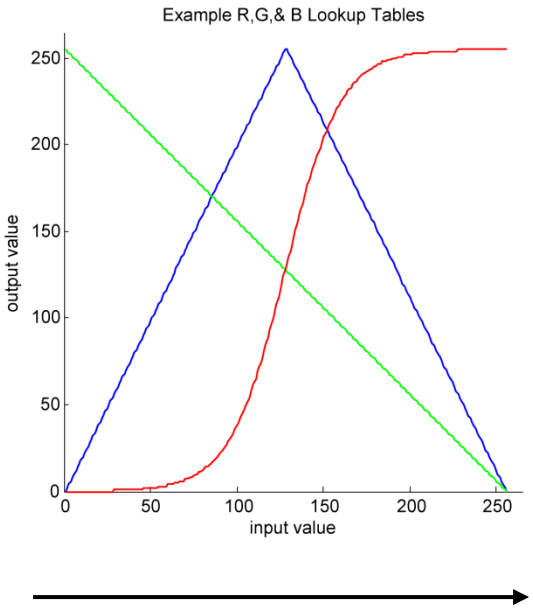
# 3-Band Image with 3-Band LUT.

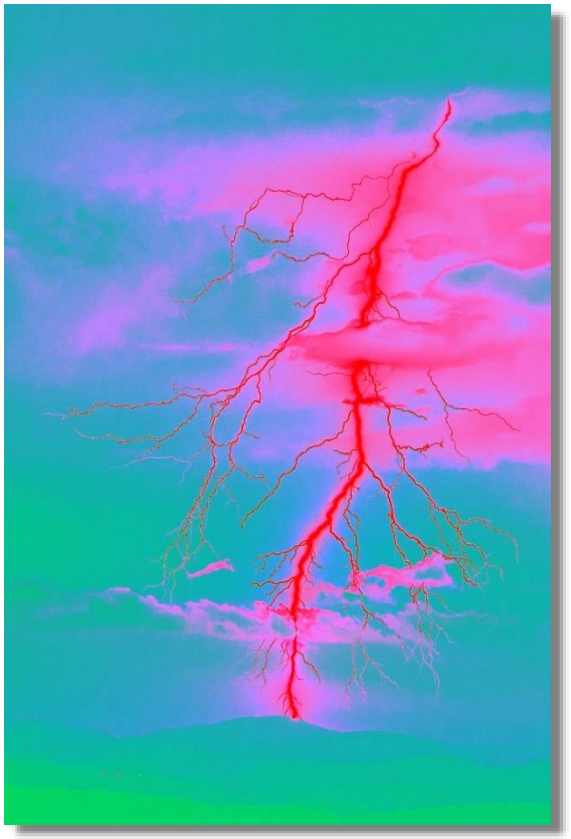


Lightning at Ramasse, Rhone-Alpes, France by Flickr user, Regarde là-bas, https://www.flickr.com/photos/marcel_s_s/8624344496/in/pool-tbasab/

# 3-Band Image with 3-Band LUT.



Lightning at Ramasse, Rhone-Alpes, France by Flickr user, Regarde là-bas, https://www.flickr.com/photos/marcel_s_s/8624344496/in/pool-tbasab/

# Basic Point Processing: Brightness, Contrast, and Gamma

# Point Processing

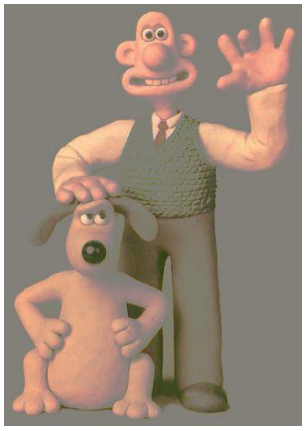
- gamma

- brightness

original

+ brightness

+ gamma



histogram mod

- contrast

original

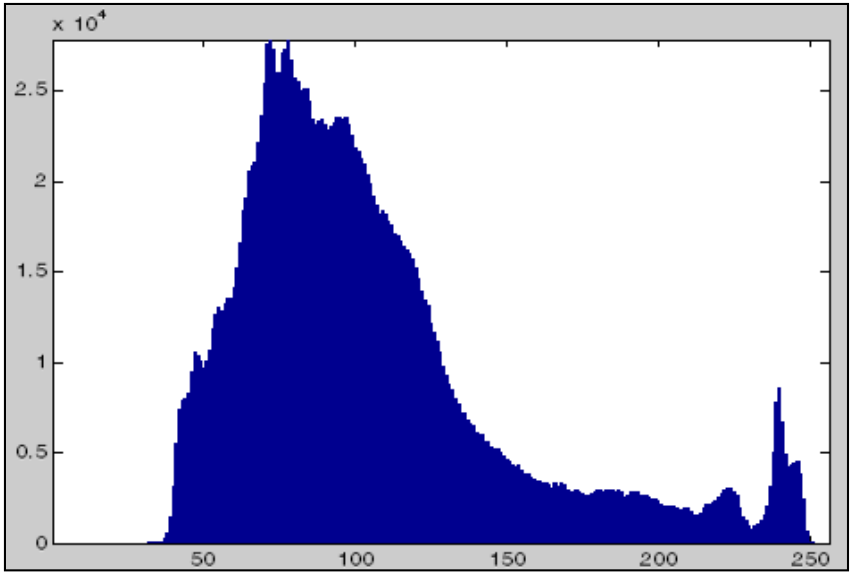
+ contrast

histogram EQ

# Point Processes: Original Image

Kinkaku-ji (金閣寺, Temple of the Golden Pavilion), also known as Rokuon-ji (鹿苑寺, Deer Garden Temple), is a Zen Buddhist temple in Kyoto, Japan.

Photo by Alan Peters, August 1993.



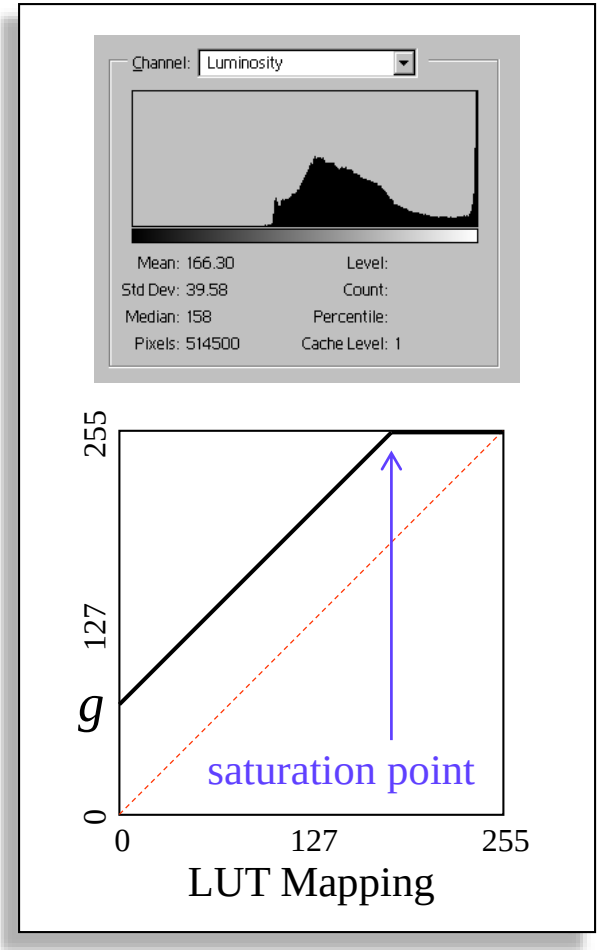

Luminance Histogram

For more information on this fascinating, unique place read the historical novel by Mishima,Yukio, *The Temple of the Golden Pavilion*, translated by Ivan Morris, Shinchosha Publishing Co, Ltd., 1956.

# Point Processes: Increase Brightness

$$\mathbf{J}(r,c,b) = \begin{cases} \mathbf{I}(r,c,b) + g, & \text{if } \mathbf{I}(r,c,b) + g < 256 \\ 255, & \text{if } \mathbf{I}(r,c,b) + g > 255 \end{cases}$$

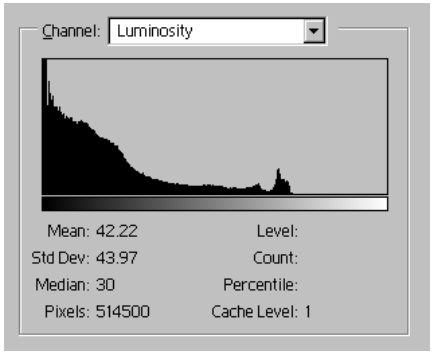
$g \geq 0$ and $b \in \{1,2,3\}$ is the band index.

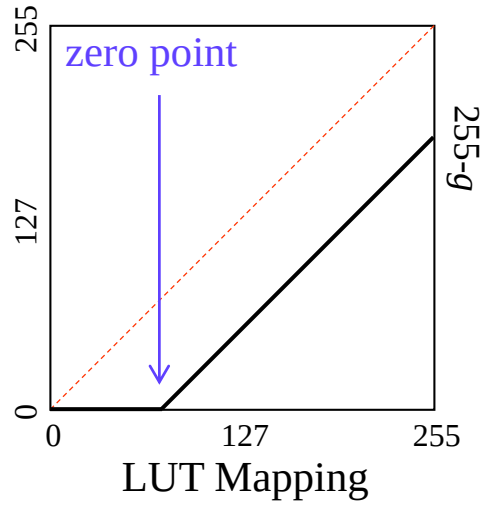


# Point Processes: Decrease Brightness

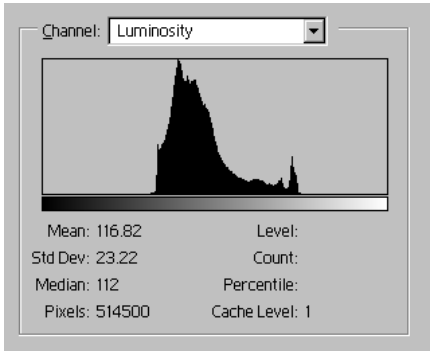
$$\mathbf{J}(r,c,b) = \begin{cases} 0, & \text{if } \mathbf{I}(r,c,b) - g < 0 \\ \mathbf{I}(r,c,b) - g, & \text{if } \mathbf{I}(r,c,b) - g > 0 \end{cases}$$

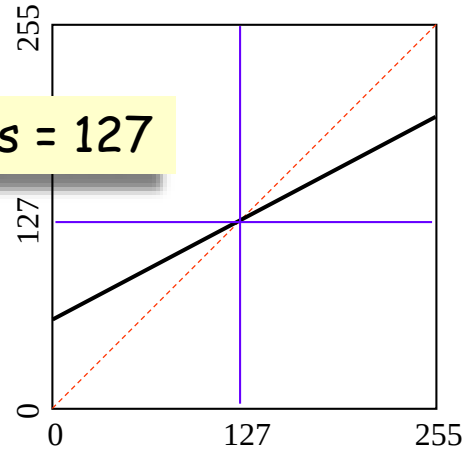
$g \geq 0$ and $b \in \{1,2,3\}$ is the band index.



# Point Processes: Decrease Contrast




Here, s = 127

$$\mathbf{T}(r,c,b) = a\left[\mathbf{I}(r,c,b) - s\right] + s,$$

where $0 \le a < 1.0$,

$s \in \{0,1,2,\ldots,255\}$, and

$b \in \{1,2,3\}$.

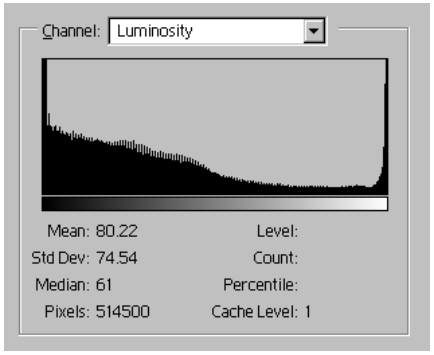
s is the center of the contrast function.

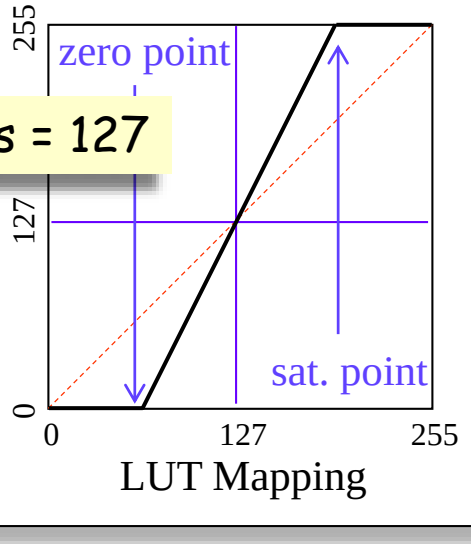
# Point Processes: Increase Contrast

$$\mathbf{T}(r,c,b) = a\left[\mathbf{I}(r,c,b) - s\right] + s$$

$$\mathbf{J}(r,c,b) = \begin{cases} 0, & \text{if } \mathbf{T}(r,c,b) < 0, \\ \mathbf{T}(r,c,b), & \text{if } 0 \le \mathbf{T}(r,c,b) \le 255, \\ 255, & \text{if } \mathbf{T}(r,c,b) > 255. \end{cases}$$

$$a > 1, \quad s \in \{0,\ldots,255\}, \quad b \in \{1,2,3\}$$




Here, s = 127

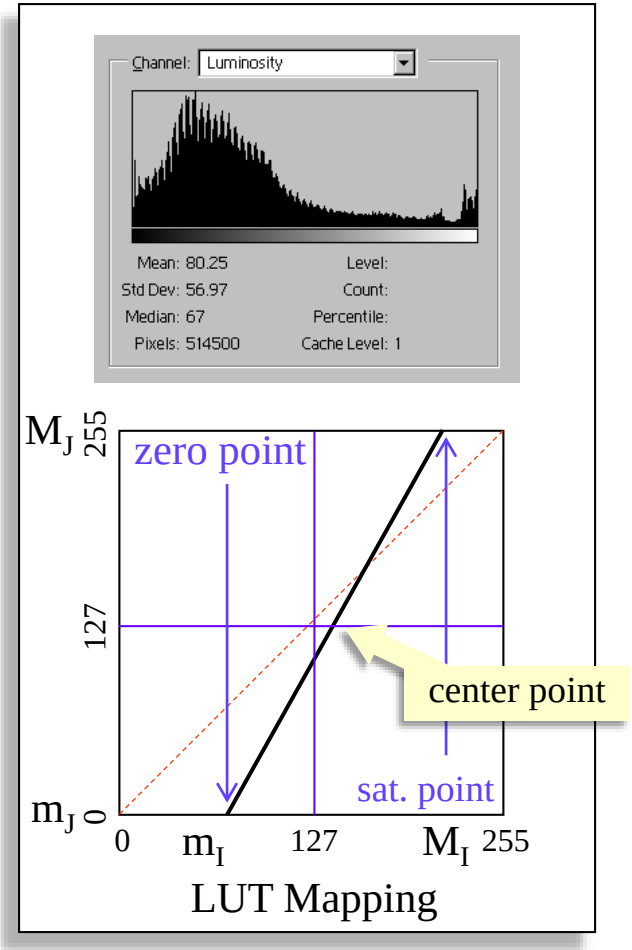
# Point Processes: Contrast Stretch

Let $m_{\mathrm{I}} = \min\left[\mathbf{I}(r,c)\right]$, $M_{\mathrm{I}} = \max\left[\mathbf{I}(r,c)\right]$,

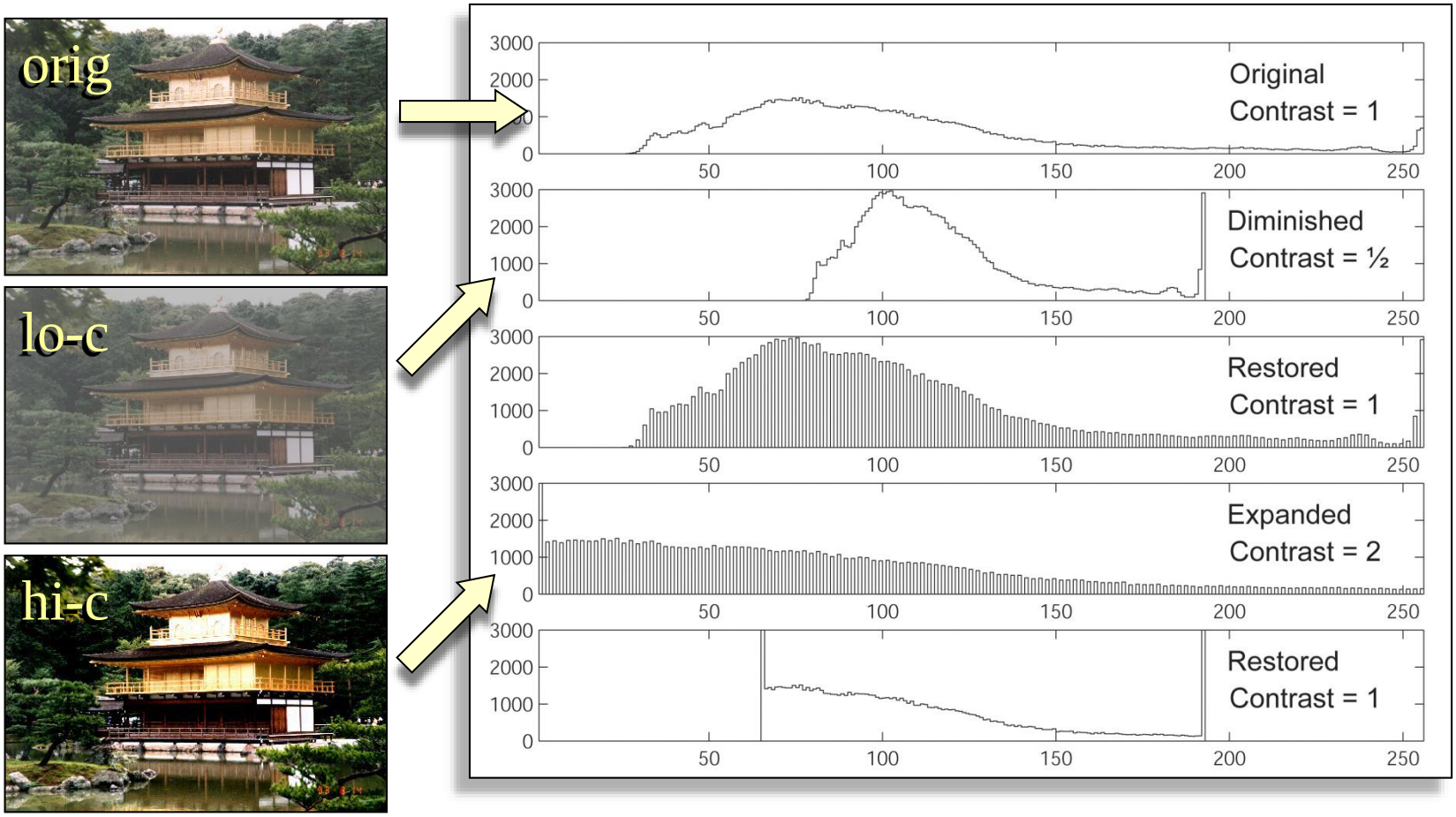
$m_{\mathrm{J}} = \min\left[\mathbf{J}(r,c)\right]$, $M_{\mathrm{J}} = \max\left[\mathbf{J}(r,c)\right]$.

Then,

$$\mathbf{J}(r,c) = \left(M_{\mathrm{J}} - m_{\mathrm{J}}\right)\frac{\mathbf{I}(r,c) - m_{\mathrm{I}}}{M_{\mathrm{I}} - m_{\mathrm{I}}} + m_{\mathrm{J}}.$$



# Information Loss from Contrast Adjustment

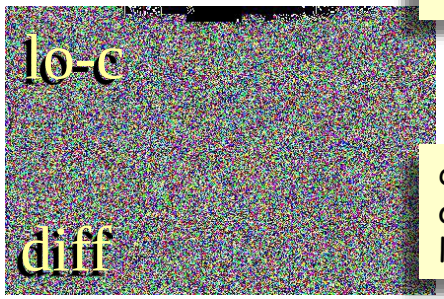


# Information Loss from Contrast Adjustment




abbreviations:
original
low-contrast
high-contrast
restored
difference





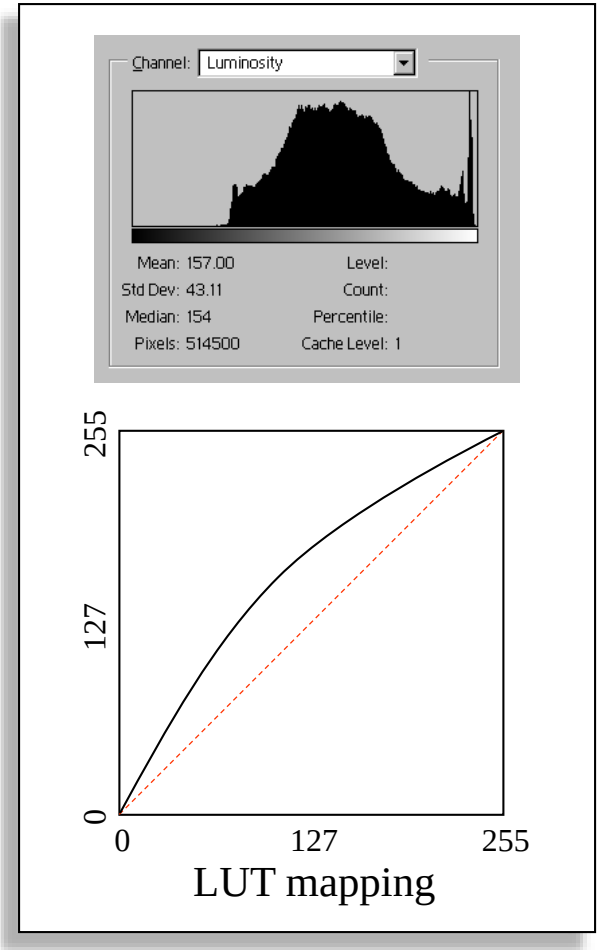

difference between original and restored low-contrast



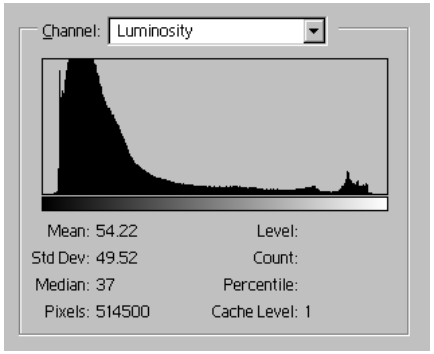

difference between original and restored high-contrast

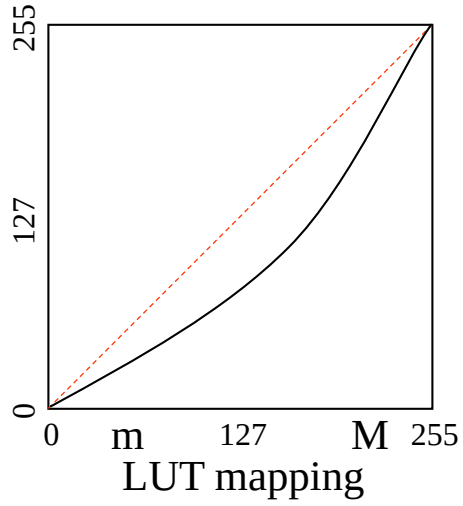
# Point Processes: Increased Gamma

$$\mathbf{J}(r,c) = 255 \cdot \left[ \frac{\mathbf{I}(r,c)}{255} \right]^{\frac{1}{\gamma}} \quad \text{for} \quad \gamma > 1.0$$



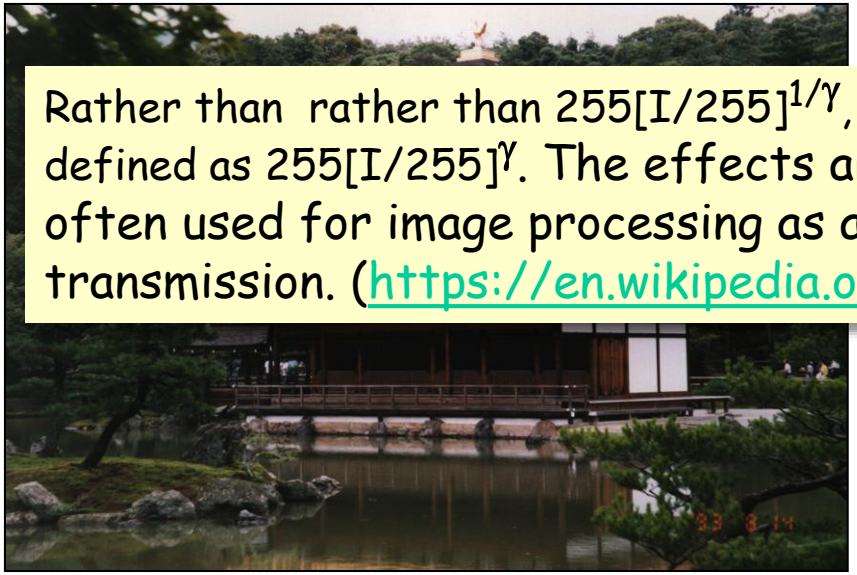
# Point Processes: Decreased Gamma



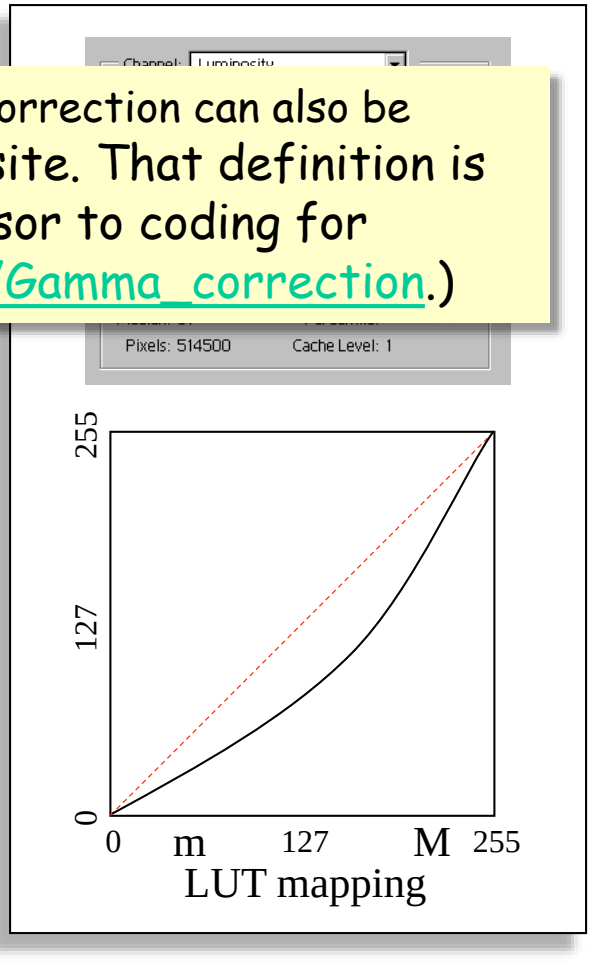




$$\mathbf{J}(r,c) = 255 \cdot \left[ \frac{\mathbf{I}(r,c)}{255} \right]^{\frac{1}{\gamma}} \quad \text{for} \quad 0 < \gamma < 1.0$$

# Point Processes: Decreased Gamma

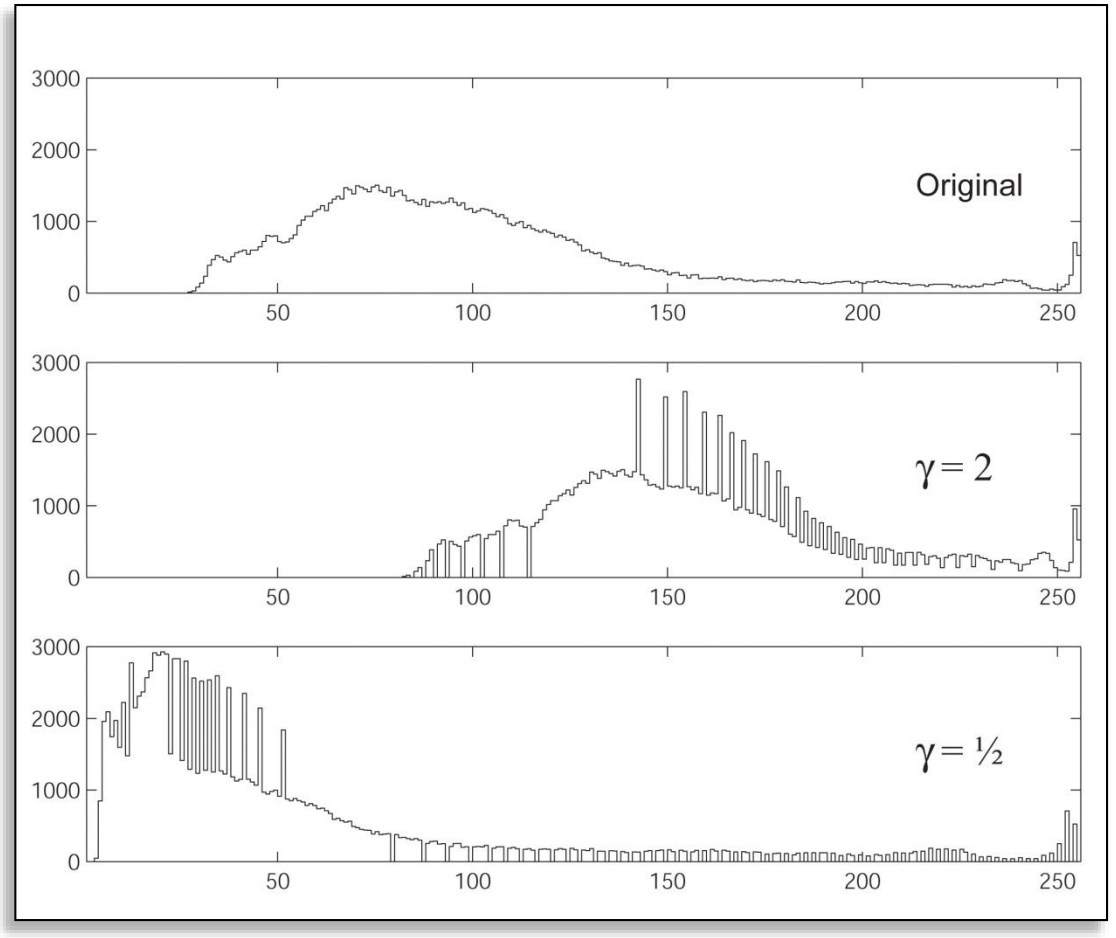


Rather than rather than $255[I/255]^{1/\gamma}$, gamma correction can also be defined as $255[I/255]^{\gamma}$. The effects are opposite. That definition is often used for image processing as a precursor to coding for transmission. (https://en.wikipedia.org/wiki/Gamma_correction.)




$$\mathbf{J}(r,c) = 255 \cdot \left[ \frac{\mathbf{I}(r,c)}{255} \right]^{\frac{1}{\gamma}} \quad \text{for} \quad 0 < \gamma < 1.0$$

# Gamma Correction: Effect on Histogram



# Point Processing: Histogram Equalization

# Point Processes: Histogram Equalization

Task: remap a 1-band image **I** so that its histogram is as close to constant as possible. This maximizes the contrast evenly across the entire intensity range.

Let $P_{\mathbf{I}}(\gamma+1)$ be the cumulative (probability) distribution function of **I**. Then **J** has, as closely as possible, a flat (constant) histogram if:

$$\mathbf{J}(r,c,b) = 255 \cdot P_{\mathbf{I}}\left[\mathbf{I}(r,c,b)+1\right].$$

one-band image

The scaled CDF itself is used as the LUT.

That is, to equalize a one-band image, map it through its own CDF multiplied by the maximum desired output value.

# Point Processes: Histogram Equalization

Task: remap image **I** with min = $m_{\mathbf{I}}$ and max = $M_{\mathbf{I}}$ so that its histogram is as close to constant as possible and has min = $m_{\mathbf{J}}$ and max = $M_{\mathbf{J}}$.

Let $P_{\mathbf{I}}(\gamma+1)$ be the cumulative (probability) distribution function of **I**.

Then **J** has, as closely as possible, the correct histogram if

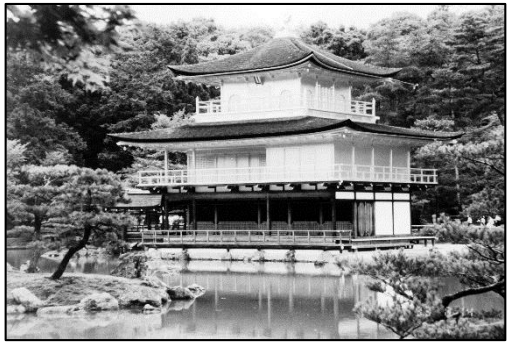
Using intensity extrema

$$\mathbf{J}(r,c) = (M_{\mathbf{J}} - m_{\mathbf{J}}) \frac{P_{\mathbf{I}}[\mathbf{I}(r,c)+1] - P_{\mathbf{I}}(m_{\mathbf{I}}+1)}{P_{\mathbf{I}}(M_{\mathbf{I}}+1) - P_{\mathbf{I}}(m_{\mathbf{I}}+1)} + m_{\mathbf{J}}.$$

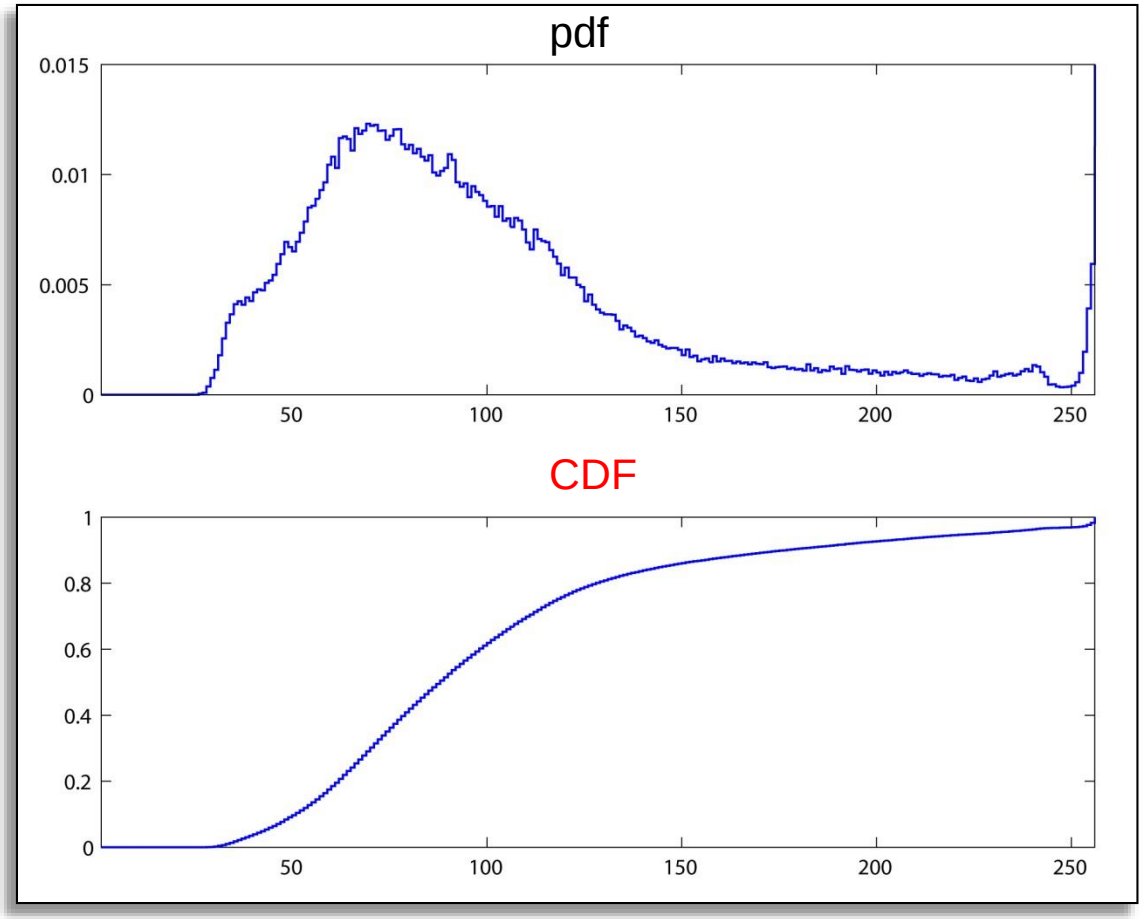
# Histogram EQ

The CDF (cumulative distribution) × 255 is the LUT for remapping.

Value Image



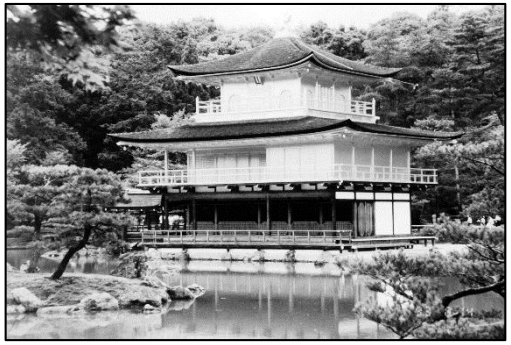
Histogram EQ'd Value Image



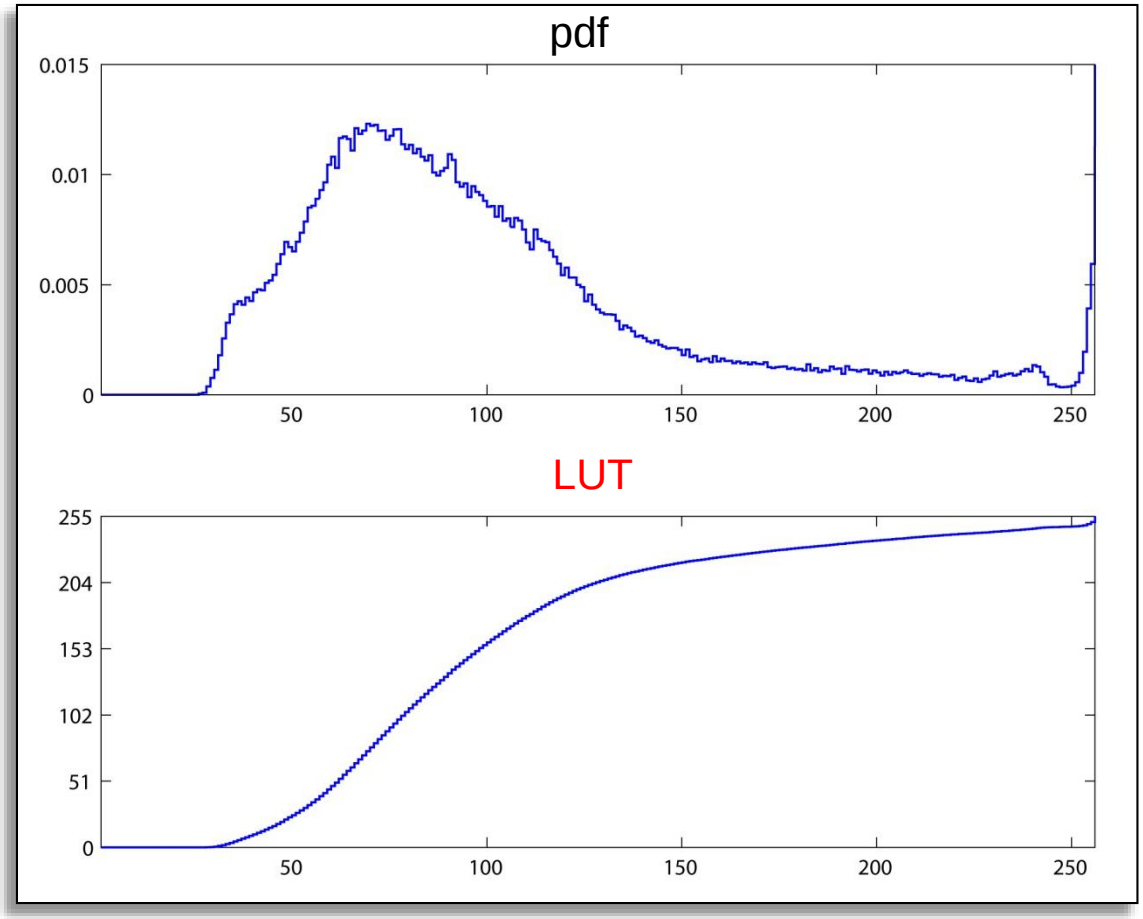
# Histogram EQ

The CDF (cumulative distribution) × 255 is the LUT for remapping.

Value Image



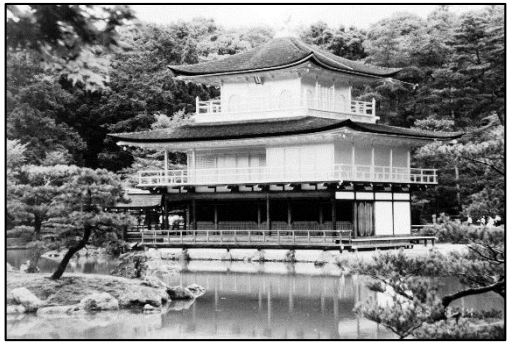
Histogram EQ'd Value Image



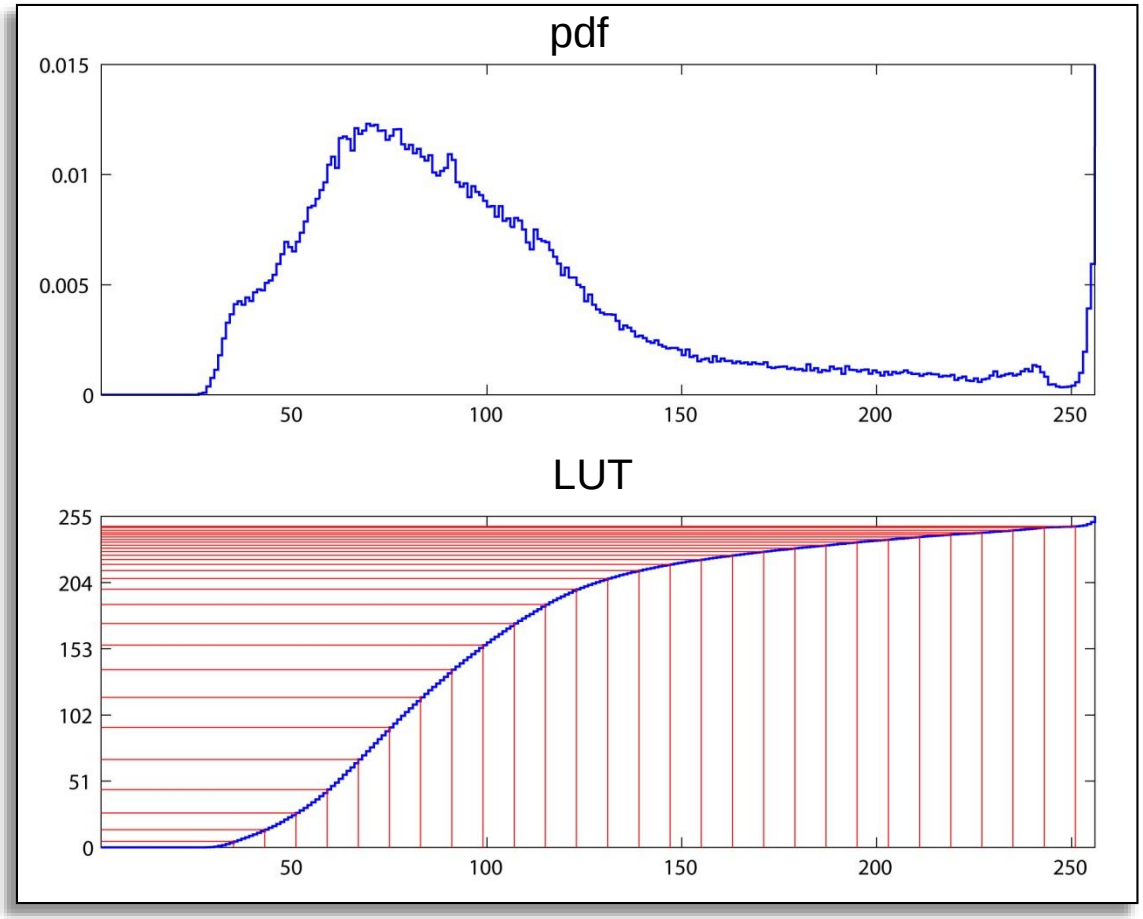
# Histogram EQ

The CDF (cumulative distribution) × 255 is the LUT for remapping.

Value Image

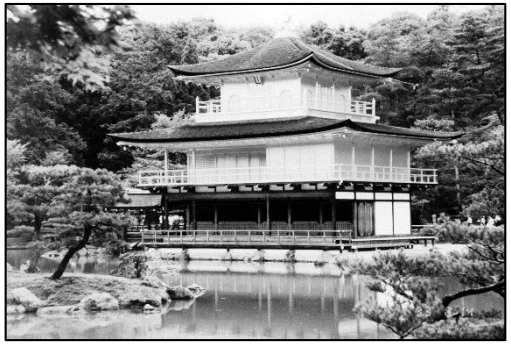


Histogram EQ'd Value Image

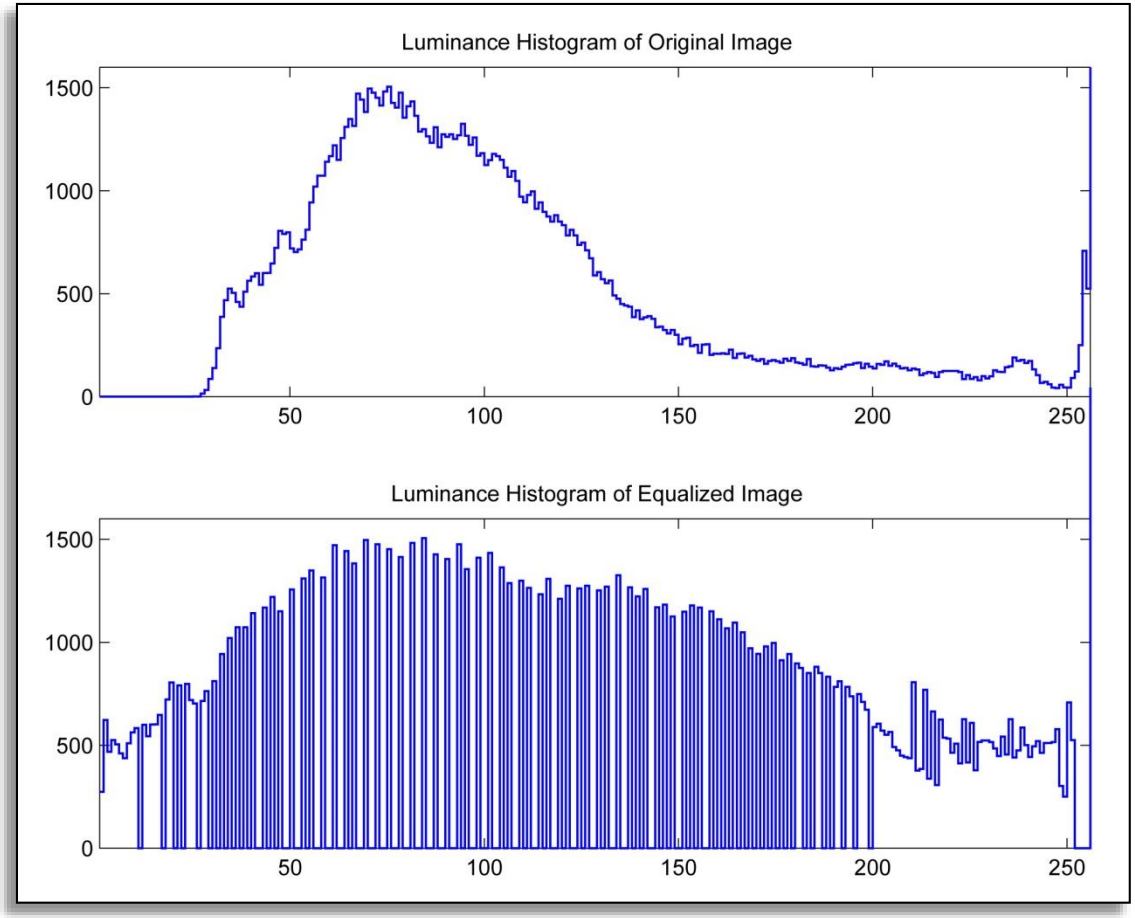


# Histogram EQ

Value Image

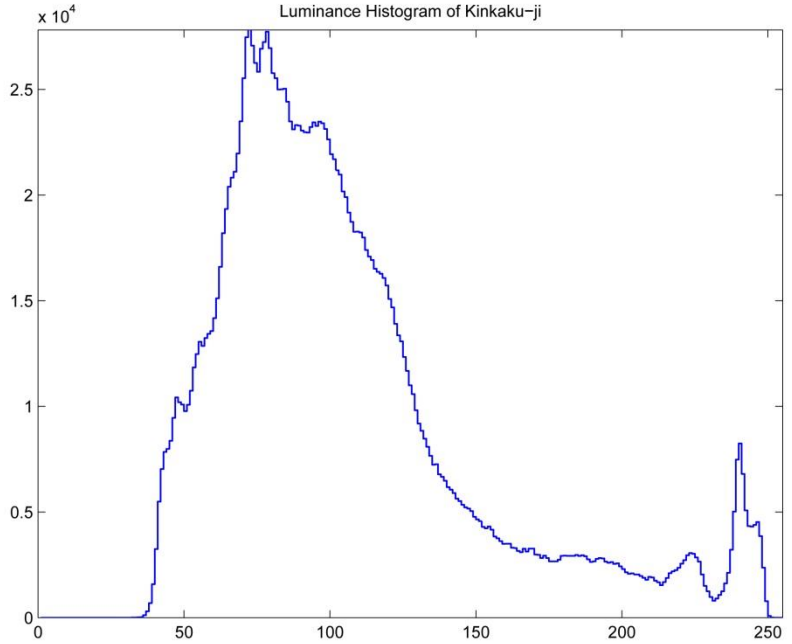


Histogram EQ'd Value Image



# Histogram EQ of a Grayscale Image

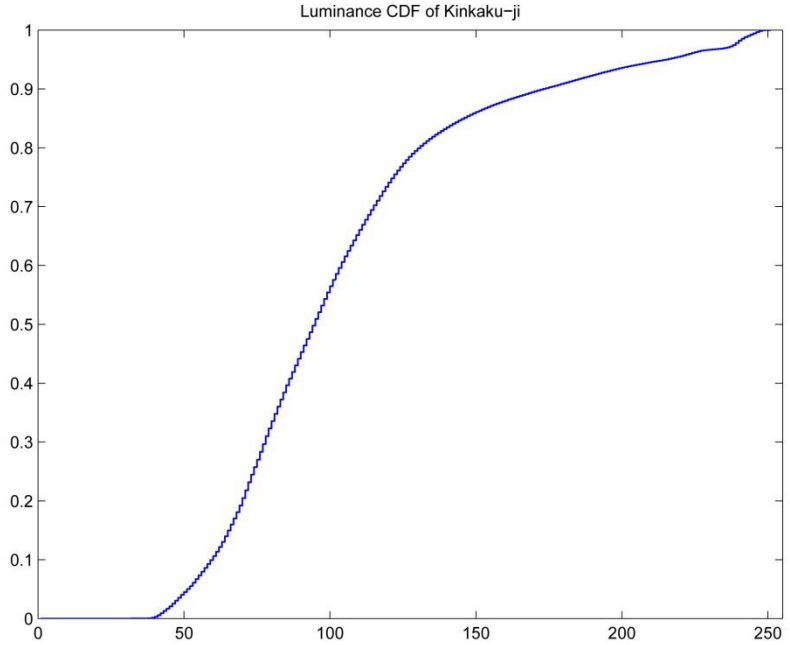
Luminance Image




Luminance Histogram

# Histogram EQ of a Grayscale Image

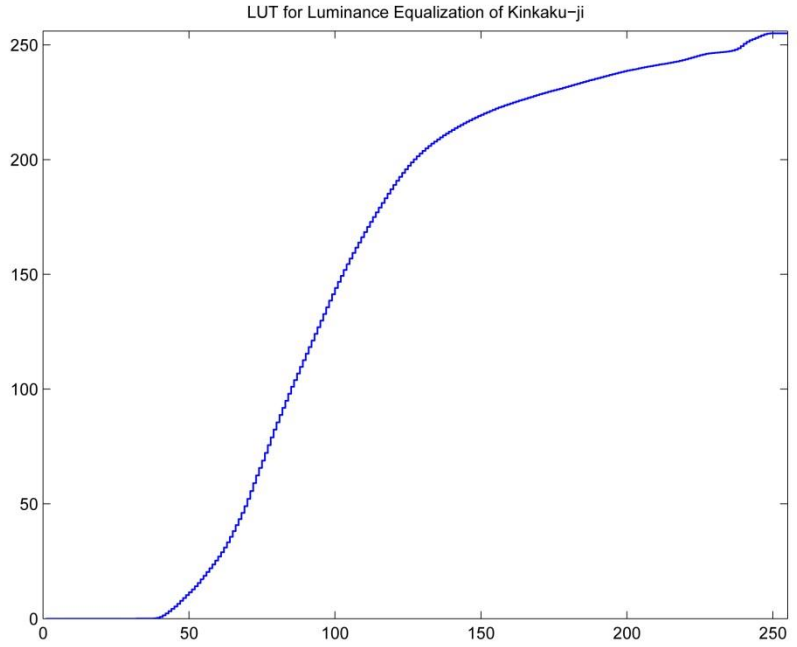
Luminance Image




Luminance CDF

# Histogram EQ of a Grayscale Image

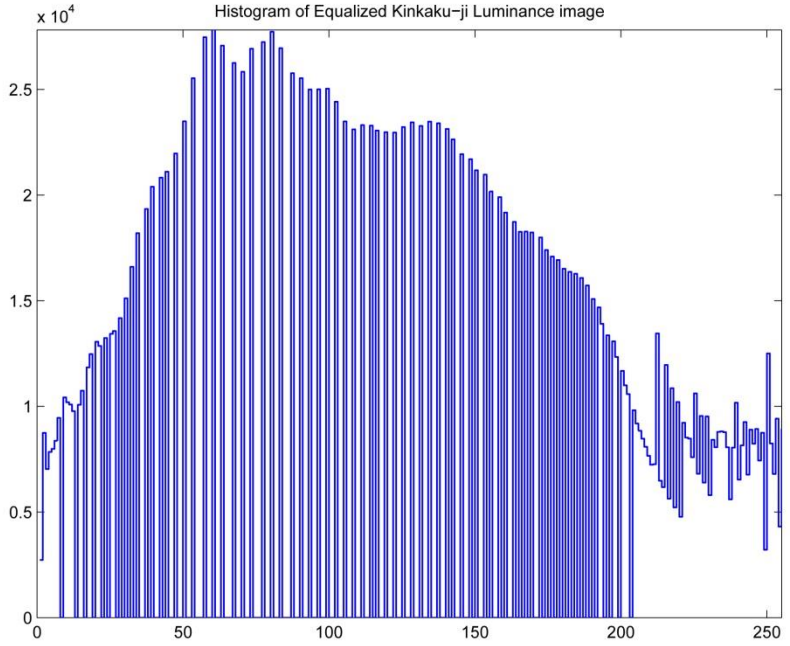
Luminance Image




Equalization LUT

# Histogram EQ of a Grayscale Image

Equalized Luminance Image




Histogram of Eq'd Image

# Histogram EQ of a Grayscale Image

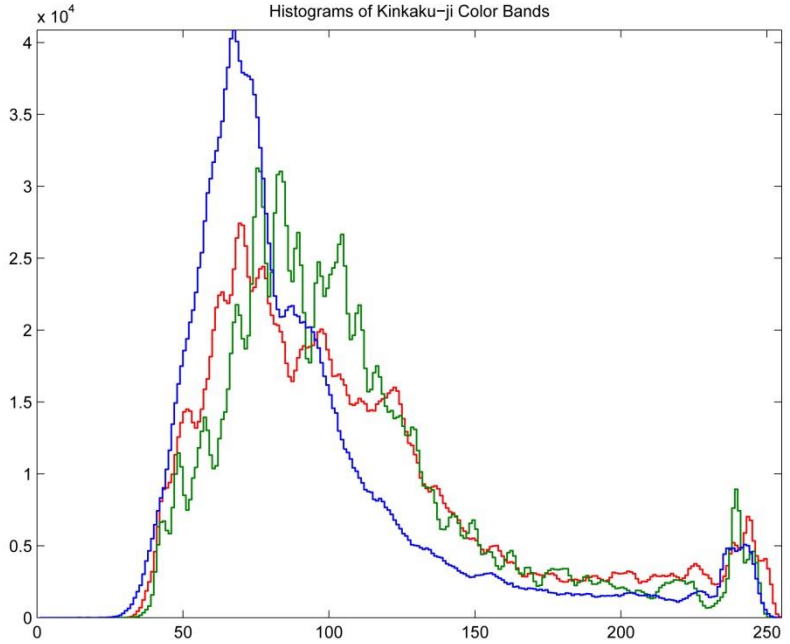
Luminance Image

Equalized Luminance Image

Note the detail loss in saturated areas.

# Histogram EQ of Individual Color Bands

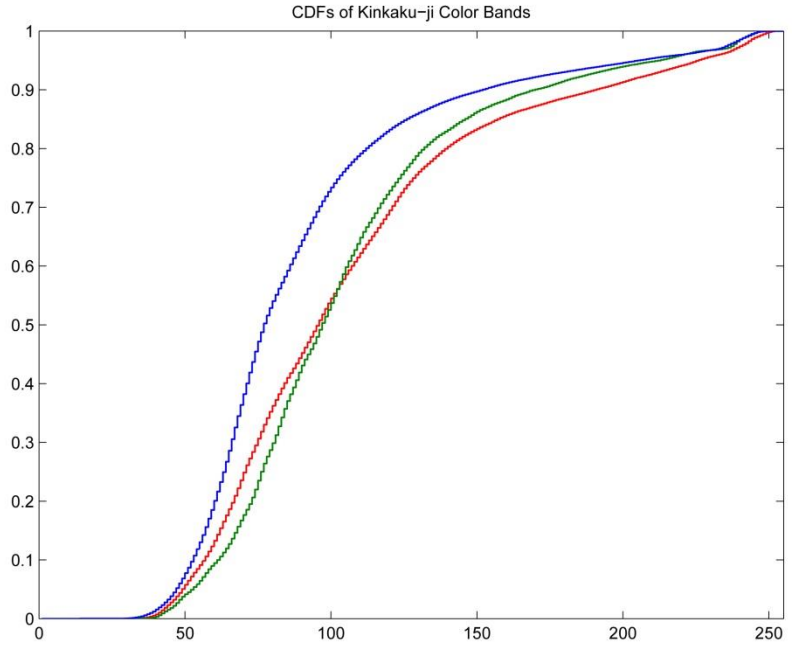
Original Color Image




Color Histograms

Here we will create a separate equalization LUT for each of the color bands from each of their CDFs.

# Histogram EQ of Individual Color Bands



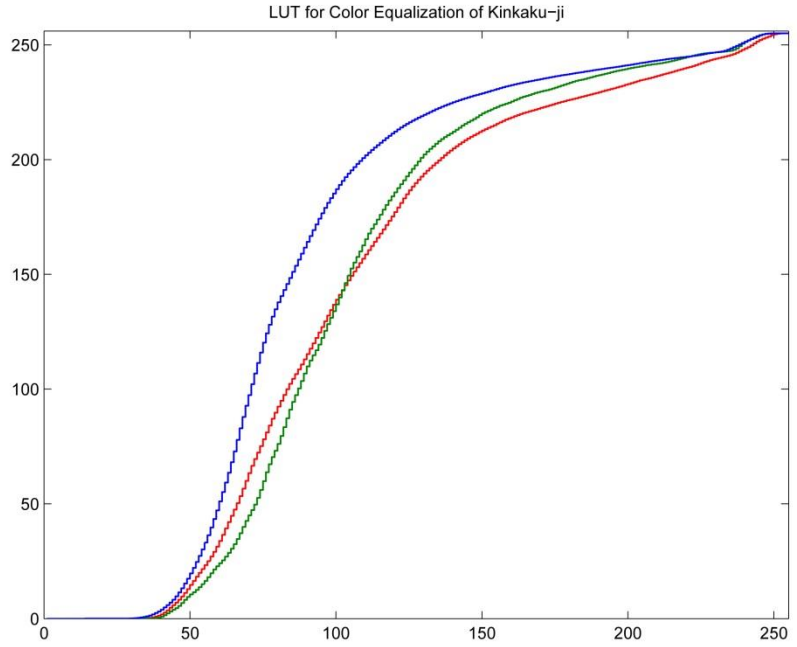

Original Color Image

Color CDFs

Here we will create a separate equalization LUT for each of the color bands from each of their CDFs.

# Histogram EQ of Individual Color Bands

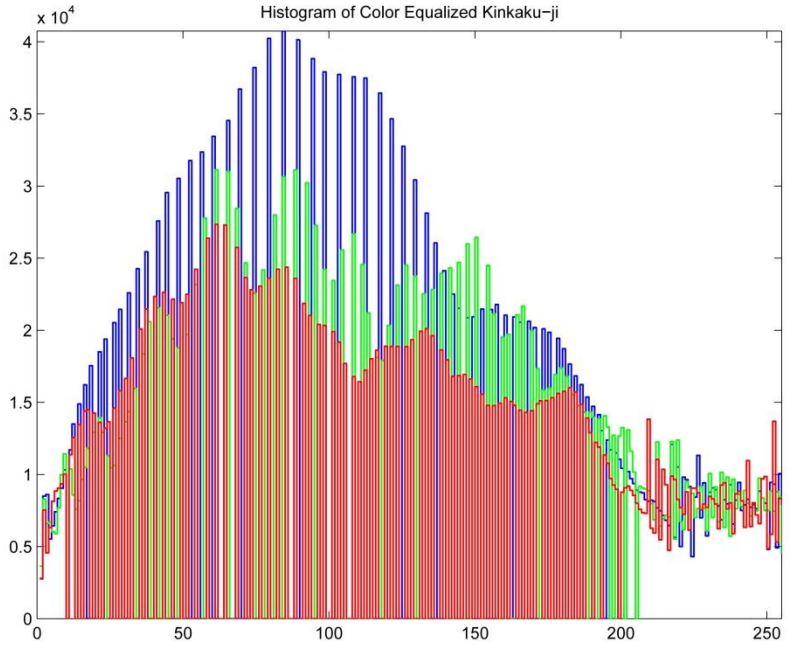
Original Color Image




Equalization LUTs

These are the separate equalization LUTs – one for each of the color bands.

# Histogram EQ of Individual Color Bands



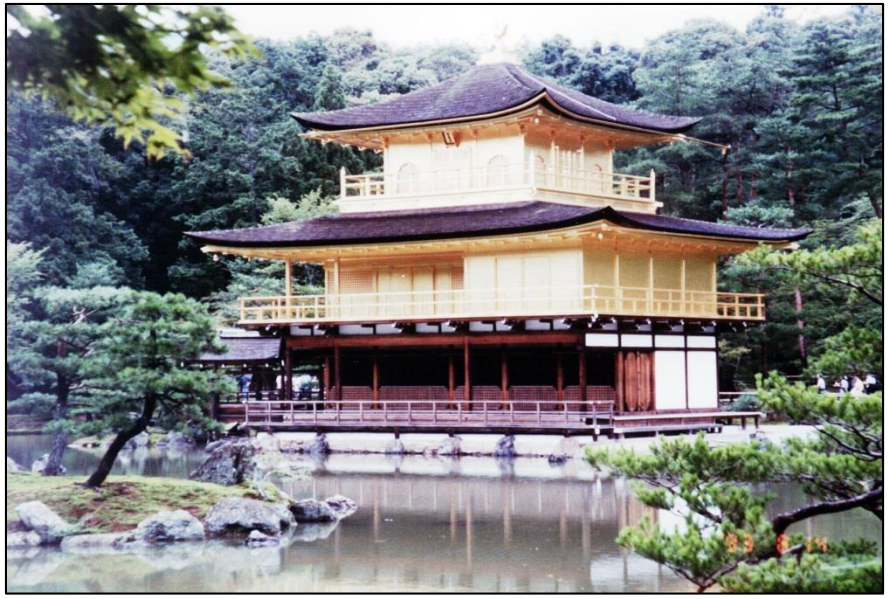

Equalized Color Image

Histogram of Eq'd Image

This is the result of mapping each band through its own equalization LUT.
Notice how the hues have changed in some parts of the image.

# Histogram EQ of Individual Color Bands

Original Color Image



Equalized Color Image

This is the result of mapping each band through its own equalization LUT.
Notice how the hues have changed in some parts of the image.

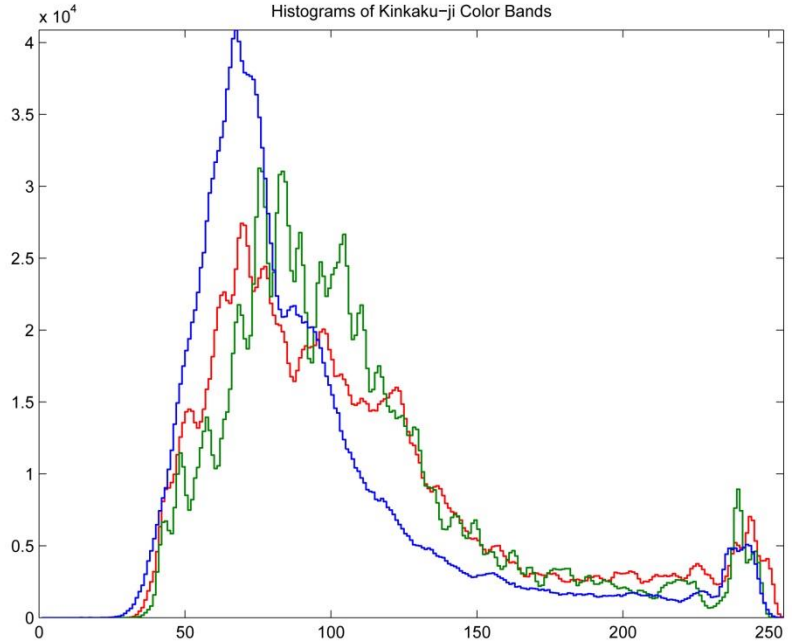
# Histogram EQ of Individual Color Bands

Equalized Color Image

Original Color Image

This is the result of mapping each band through its own equalization LUT.
Notice how the hues have changed in some parts of the image.

# Luminance EQ of a Color Image

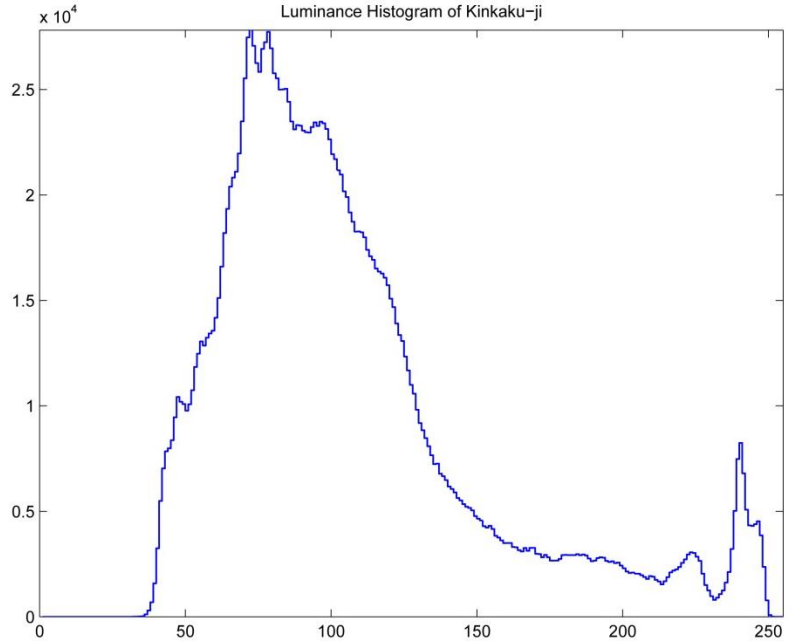



Original Color Image

Color Histograms

Here we compute the CDF of the luminance or value image, make a LUT from it, then remap all three color bands through the same LUT.

# Luminance EQ of a Color Image

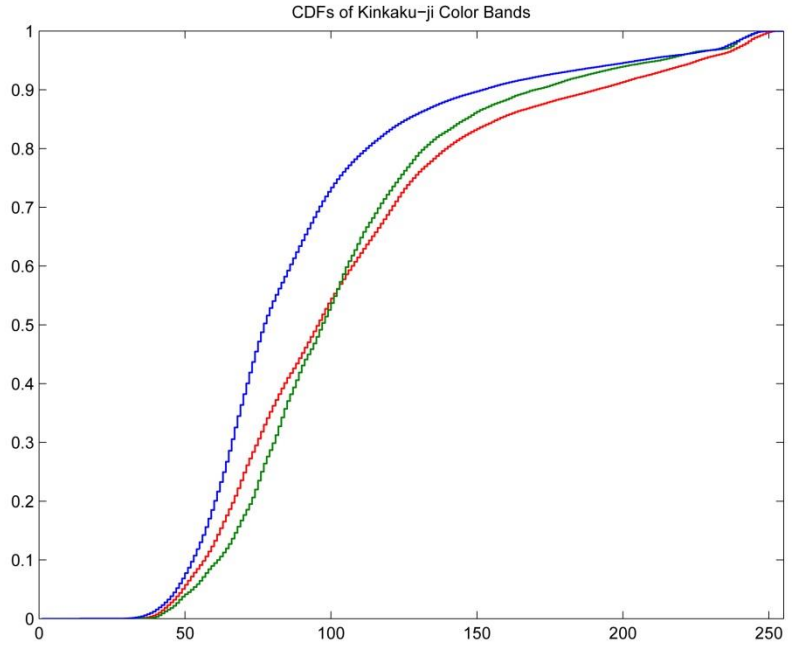



Luminance Image

Luminance Histogram

Here we compute the CDF of the luminance or value image, make a LUT from it, then remap all three color bands through the same LUT.

# Luminance EQ of a Color Image

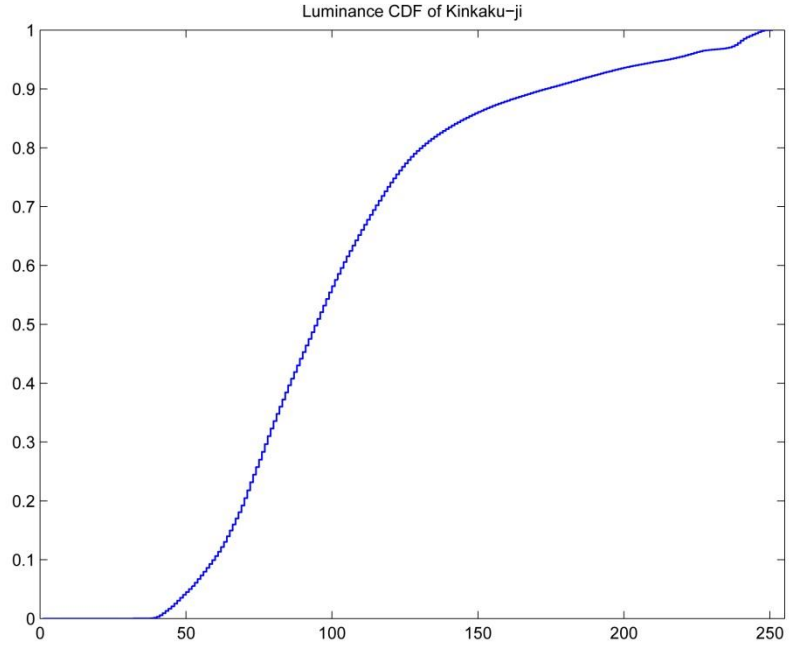



Original Color Image

Color CDFs

Here we compute the CDF of the luminance or value image, make a LUT from it, then remap all three color bands through the same LUT.

# Luminance EQ of a Color Image

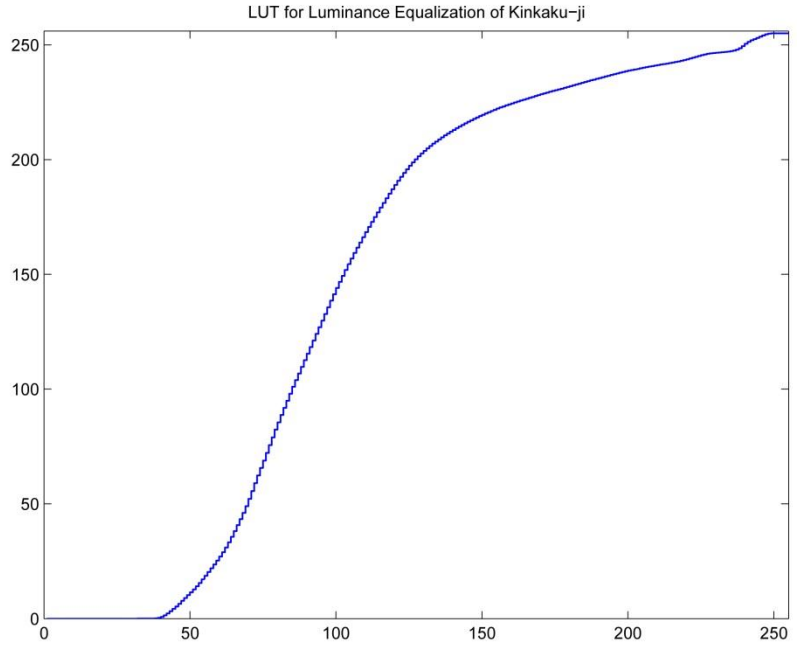



Luminance Image

Luminance CDF

Here we compute the CDF of the luminance or value image, make a LUT from it, then remap all three color bands through the same LUT.

# Luminance EQ of a Color Image




Original Color Image

Equalization LUT

Here we compute the CDF of the luminance or value image, make a LUT from it, then remap all three color bands through the same LUT.

# Luminance EQ of a Color Image

Luminance Eq'd Color Image

Original Color Image

This is what we get when we map each of the three bands through the LUT created from the CDF of the image's luminance.

# Histogram EQ of the Individual Bands

Each band EQ'd Separately  Luminance EQ'd Color Image

The left image is what we got when we equalized each of the three bands separately. The right image is what we get when we map each color band through the luminance CDF

# Luminance EQ of a Color Image

Original Color Image

Luminance Eq'd Color Image

Remapping each of the color bands through the luminance CDF LUT preserves the hues of the original image better than the individual equalization of each color band.

# Point Processing: Histogram Matching

# Point Processing: Histogram Matching

Task: Remap image $\mathbf{I}_d$ (discolored) so that it has, as closely as possible, the same histogram as image $\mathbf{J}_r$ (reference).

Q: Why do this?
A: Restore a degraded image based on an original. Match the characteristics of images of the same scene from different cameras.

# Point Processing: Histogram Matching

Task: Remap image $\mathbf{I}_d$ (discolored) so that it has, as closely as possible, the same histogram as image $\mathbf{J}_r$ (reference).

Because the images are digital it is not, in general, possible to make $h_d \equiv h_r$ . Therefore, $p_d \not\equiv p_r$ .

Q: How, then, can the matching be done?
A: By matching percentiles.

# Matching Percentiles

… assuming a 1-band image, one band of a color image, or its luminance image.

Recall:

- CDF $P_d$ of image $\mathbf{I}_d$ is such that $0 \le P_d(g_d) \le 1$.
- $P_d(g_d + 1) = c$ means that $c$ is the fraction of pixels in $\mathbf{I}_d$ that have a value less than or equal to $g_d$.
- $100c$ is the *percentile* of pixels in $\mathbf{I}_d$ that are less than or equal to $g_d$.

To match percentiles, replace all occurrences of value $g_d$ in image $\mathbf{I}_d$ with value $g_r$ from image $\mathbf{J}_r$ whose percentile in $\mathbf{J}_r$ most closely matches the percentile of $g_d$ in image $\mathbf{I}_d$.

# Matching Percentiles

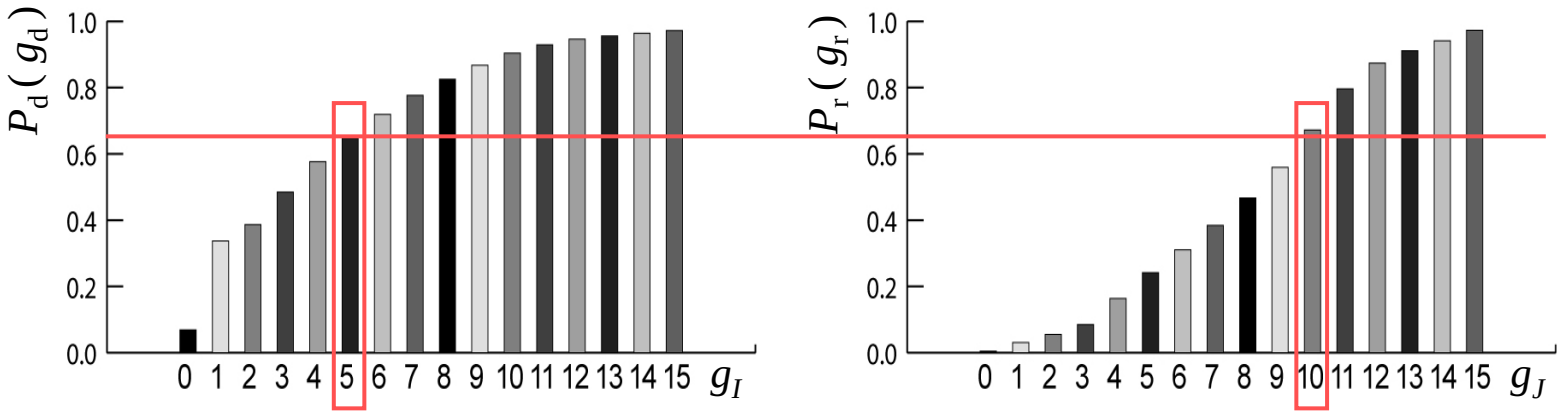
… assuming a 1-band image, one band of a color image or its luminance image.

So, to create an image, **K**, from image $\mathbf{I}_d$ such that **K** has nearly the same CDF as image $\mathbf{J}_r$ do the following:

If $\mathbf{I}_d(r,c) = g_d$ then let $K(r,c) = g_r$ where $g_r$ is such that

$$P_d(g_d) > P_r(g_r - 1) \text{ AND } P_d(g_d) \leq P_r(g_r).$$

Example:
$\mathbf{I}_d(r,c) = 5$
$P_d(5) = 0.65$
$P_r(9) = 0.56$
$P_r(10) = 0.67$
$\mathbf{K}(r,c) = 10$



# Histogram Matching Algorithm

… assuming a 1-band image, one band of a color image or its luminance image.

$[R,C] = \texttt{size}(\mathbf{I}_d);$
$\mathbf{K} = \texttt{zeros}(R,C);$
$g_r = m_r;$
for $g_d = m_d$ to $M_d$
  while $g_r < 255$ AND $P_d(g_d+1) < 1$ AND
    $P_r(g_r+1) < P_d(g_d+1)$
    $g_r = g_r + 1;$
  end
  $\mathbf{K} = \mathbf{K} + \left[ g_r \cdot (\mathbf{I} == g_d) \right]$
end

This directly matches image $\mathbf{I}_d$ to image $\mathbf{J}_r$.

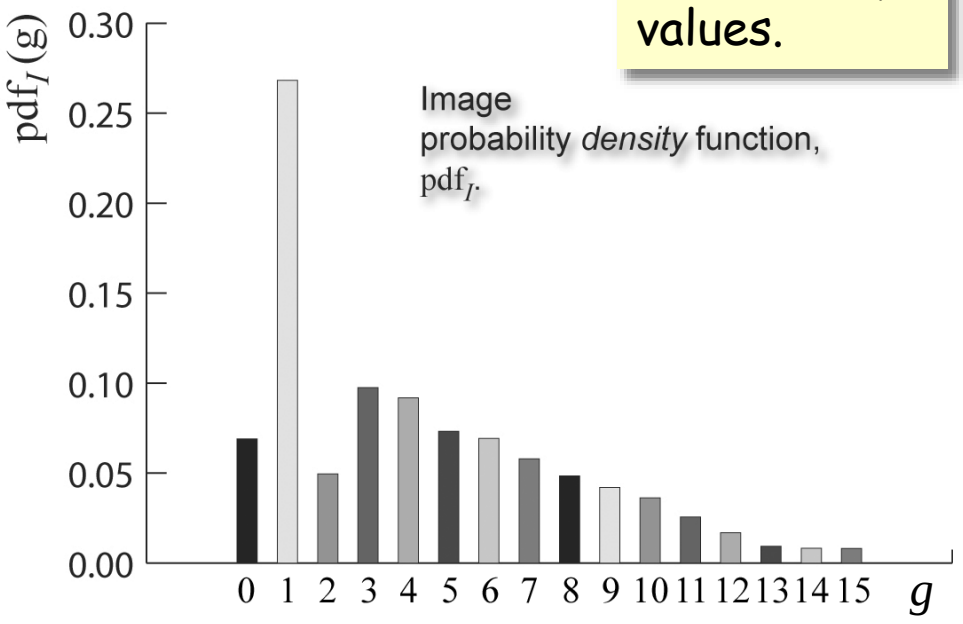
$P_d(g_d+1)$ : CDF of $\mathbf{I}_d$,
$P_r(g_r+1)$ : CDF of $\mathbf{J}_d$.
$m_r = \min \mathbf{J}_r$,
$M_r = \max \mathbf{J}_r$,
$m_d = \min \mathbf{I}_d$,
$M_d = \max \mathbf{I}_d$.

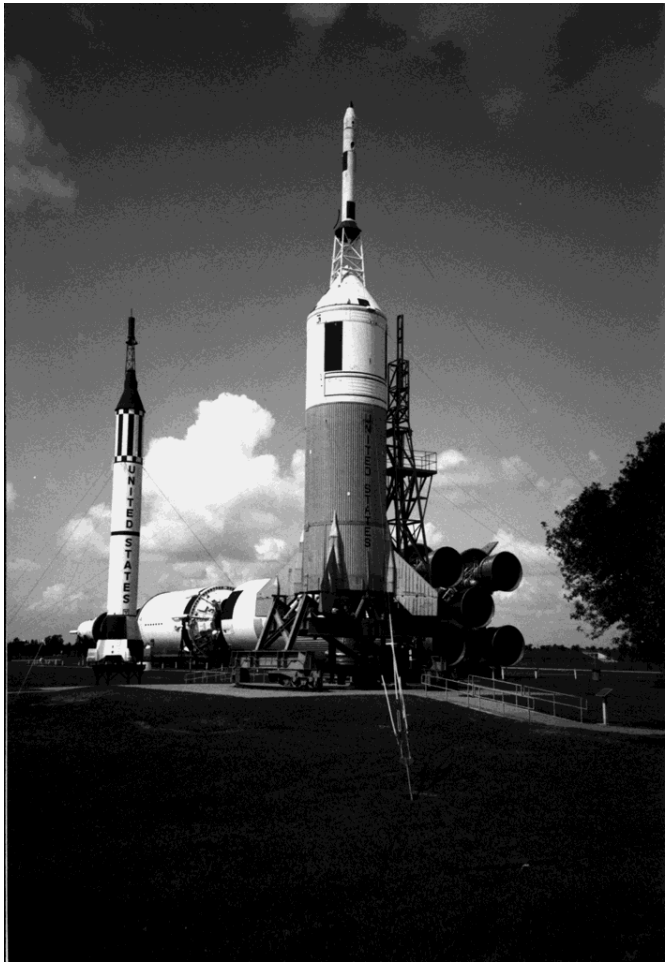
Better to use a LUT. See slide 74.

# Example: Histogram Matching

## Discolored Image pdf

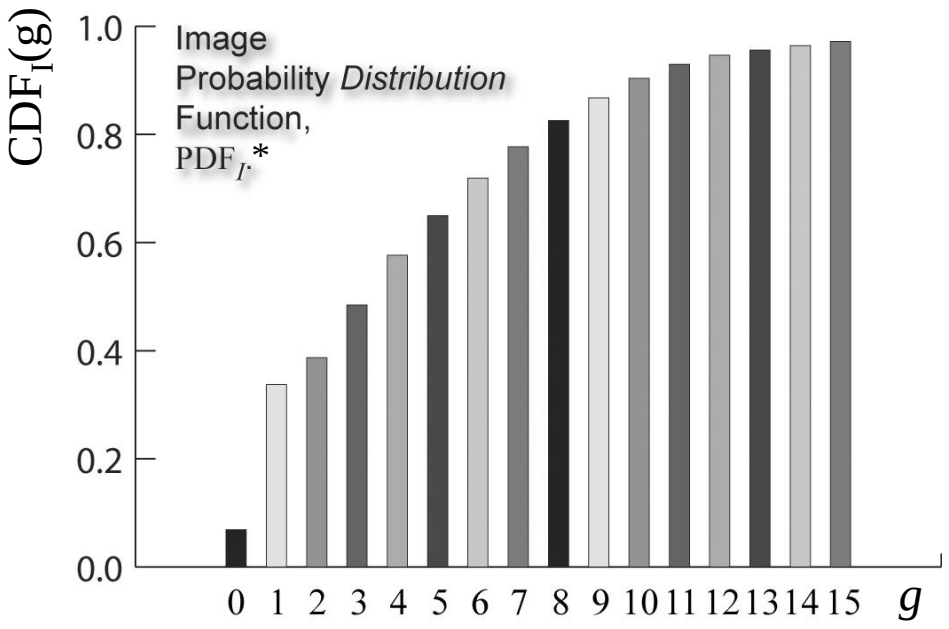
Discolored Image with 16 intensity values.

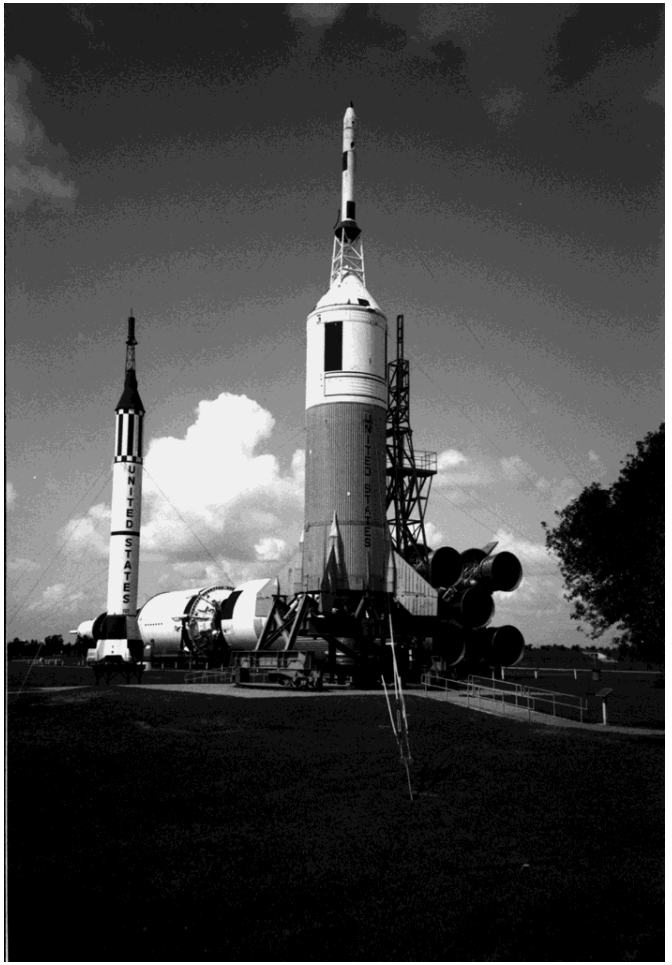


# Example: Histogram Matching

## Discolored Image CDF



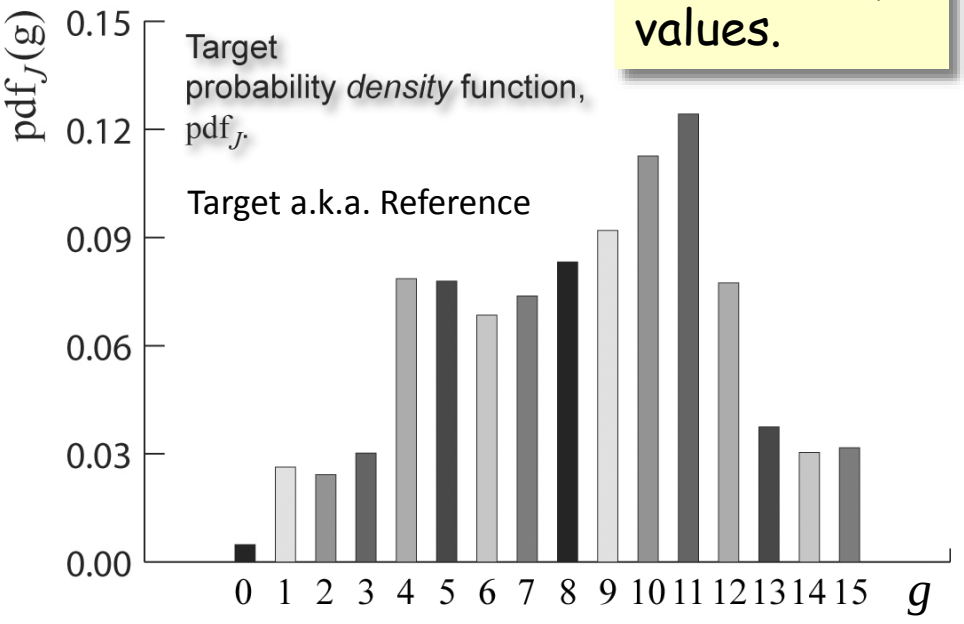



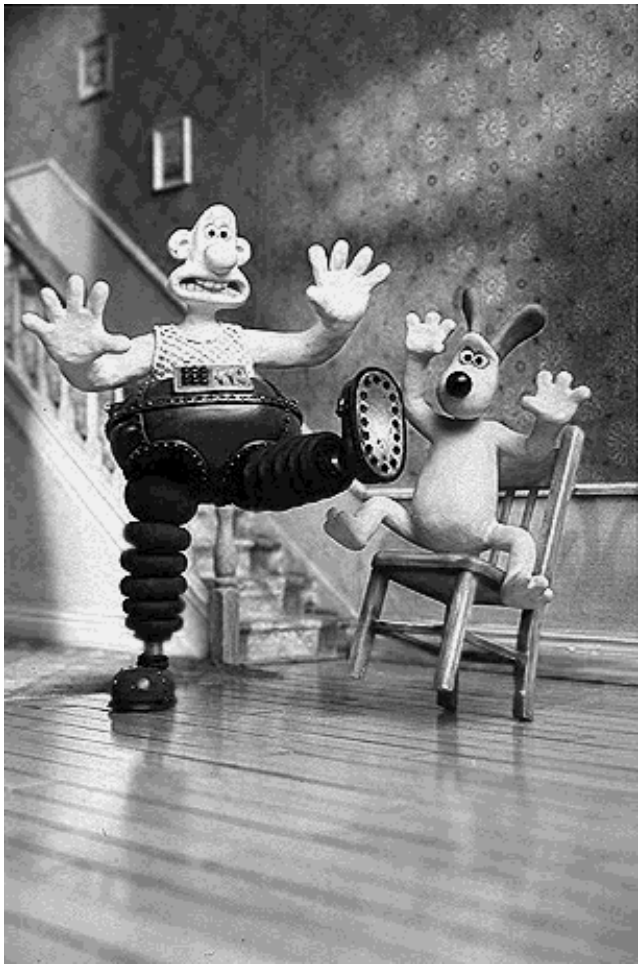
*a.k.a Cumulative Distribution Function, $CDF_I$.

# Example: Histogram Matching

## Reference Image pdf

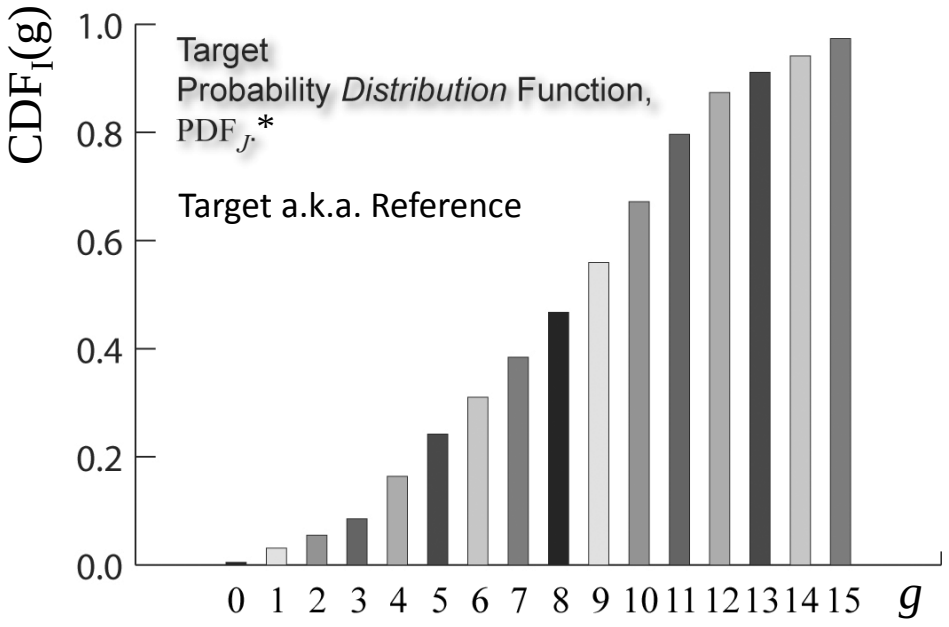
Reference image with 16 intensity values.



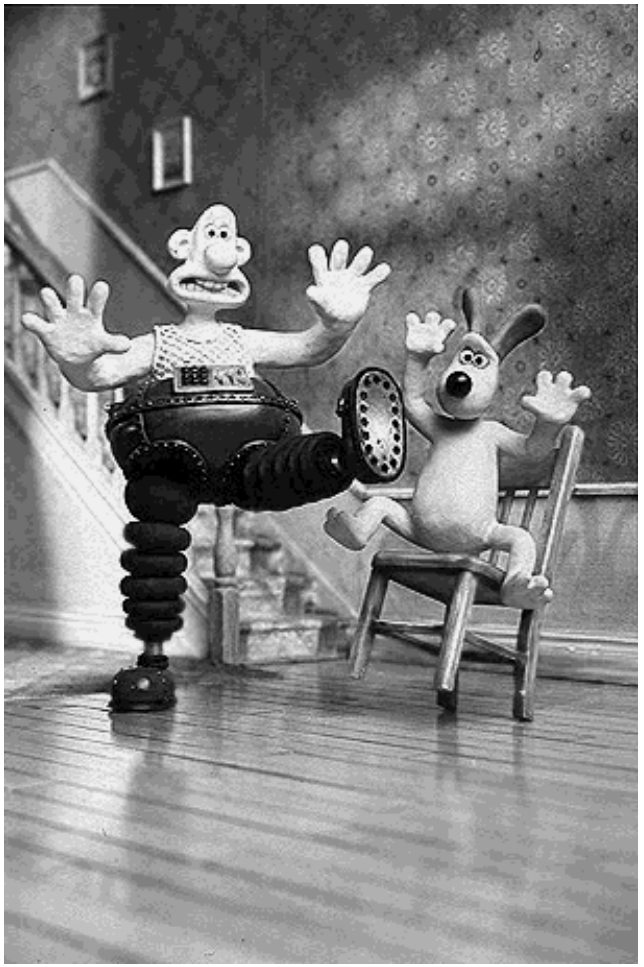



# Example: Histogram Matching

## Reference Image CDF




*a.k.a Cumulative Distribution Function, $CDF_J$.



# Histogram Matching with a Lookup Table

The algorithm on slide 66 matches one image to another directly. Often it is faster or more versatile to use a lookup table (LUT). Rather than remapping each pixel in the image separately, one can create a table that indicates to which target value each input value should be mapped. Then

$$\mathbf{K} = \mathrm{LUT}[\mathbf{I}_d + 1]$$

In *Matlab* if the LUT is a 256 × 1 matrix with values from 0 to 255 and if image $\mathbf{I}_d$ is **one-band** of type **`uint8`**, it can be remapped with the following code:

```
K = uint8(LUT(I+1));
```

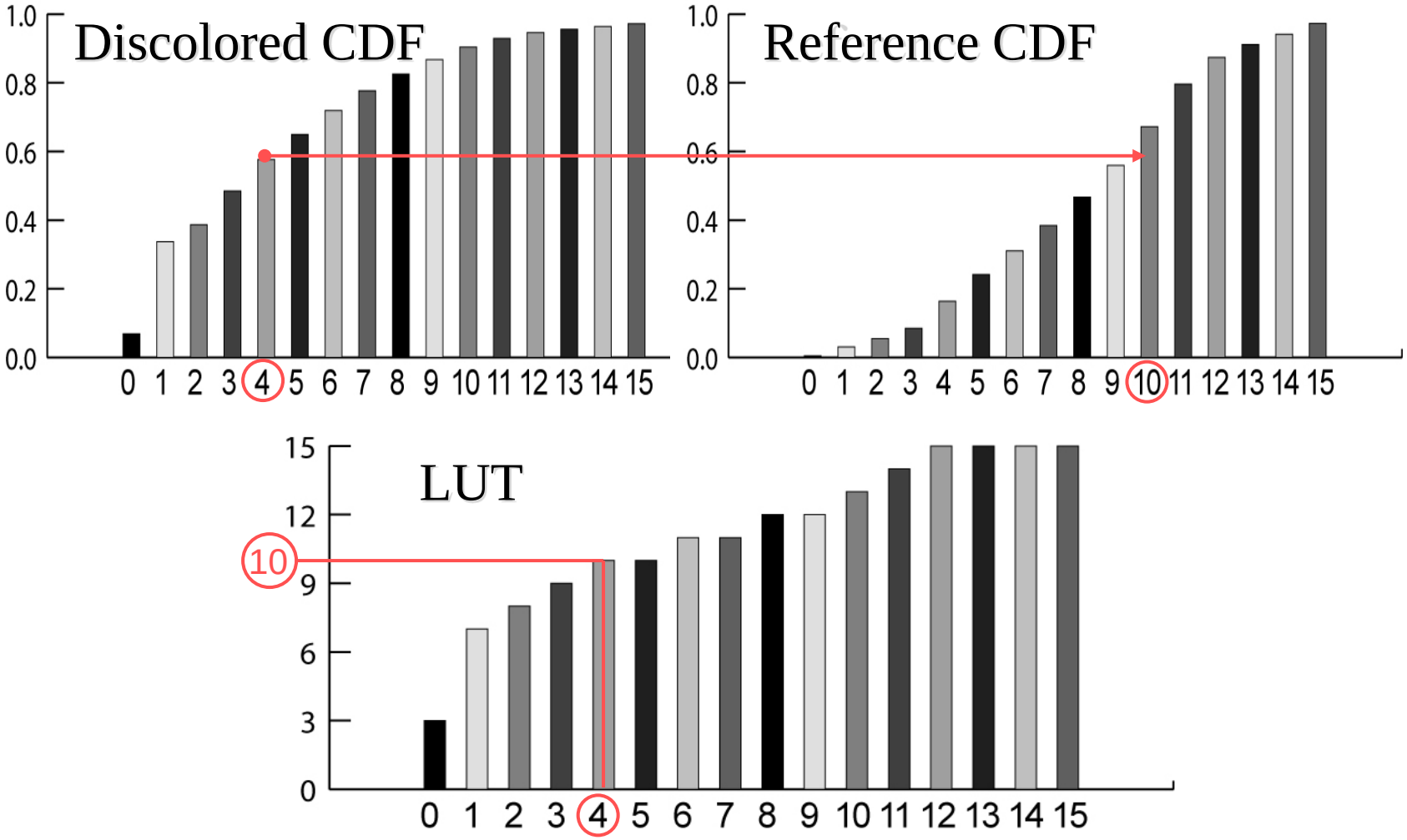
# Histogram Matching with a Lookup Table

The E-Z teenage New York version* on the previous page only works for one-band images. For truecolor or other multiband images you need to execute the LUT on each band separately. *Viz*:

```
if E == 1 % single band LUT; multi-band image
   for b = 1:B
       J(:,:,b) = LUT(1+double(I(:,:,b)));
   end
else   % multiband LUT; multi-band image
   for b = 1:B
       LUT1D = squeeze(LUT(:,1,b));
       J(:,:,b) = LUT1D(1+double(I(:,:,b)));
   end
end
```

*http://www.science.uva.nl/~robbert/zappa/albums/Zappa_In_New_York/10.html & http://youtu.be/GDwRJK8bpb4

# LUT Creation



# Look Up Table for Histogram Matching

LUT = zeros(256,1);
$g_r = 0$;
for $g_d = 0$ to 255
    while $P_r(g_r+1) < P_d(g_d+1)$ AND $g_r < 255$
        $g_r = g_r + 1$;
    end
    $\mathrm{LUT}(g_d+1) = g_r$;
end

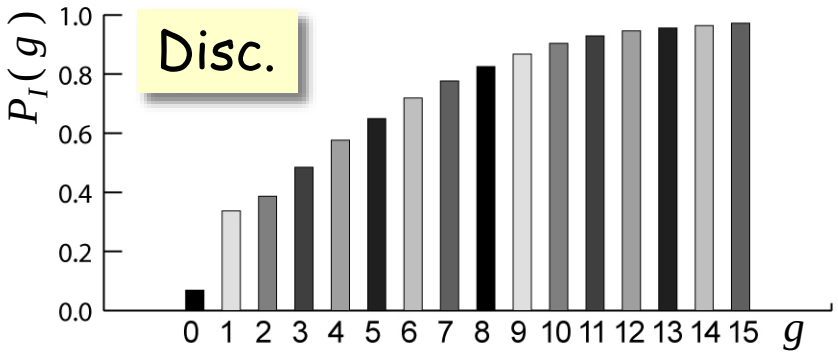
This creates a look-up table which can then be used to remap the image.

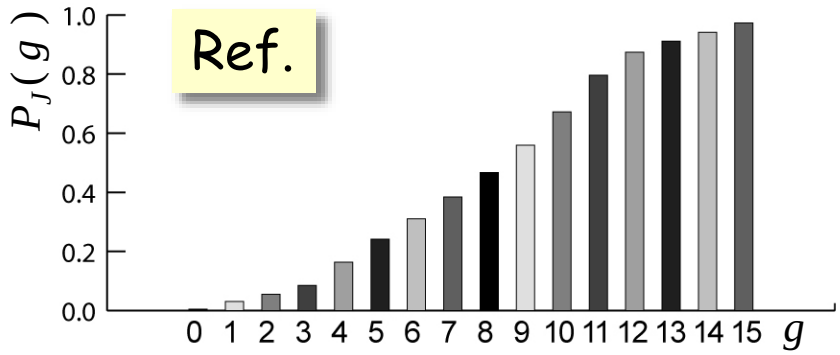
$P_d(g_d+1)$: CDF of $\mathbf{I}_d$,

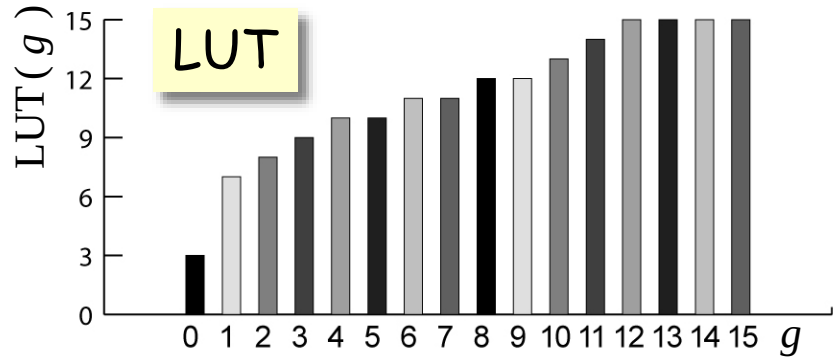
$P_r(g_r+1)$: CDF of $\mathbf{J}_r$,

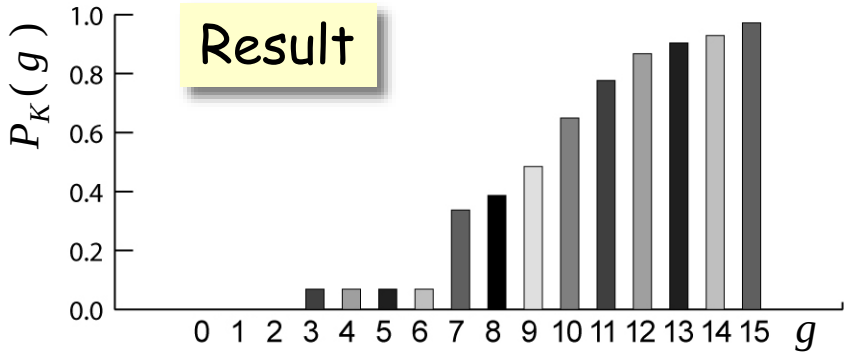
$\mathrm{LUT}(g_d+1)$: Look-Up Table

# Discolored & Reference CDFs, LUT, and Resultant CDF

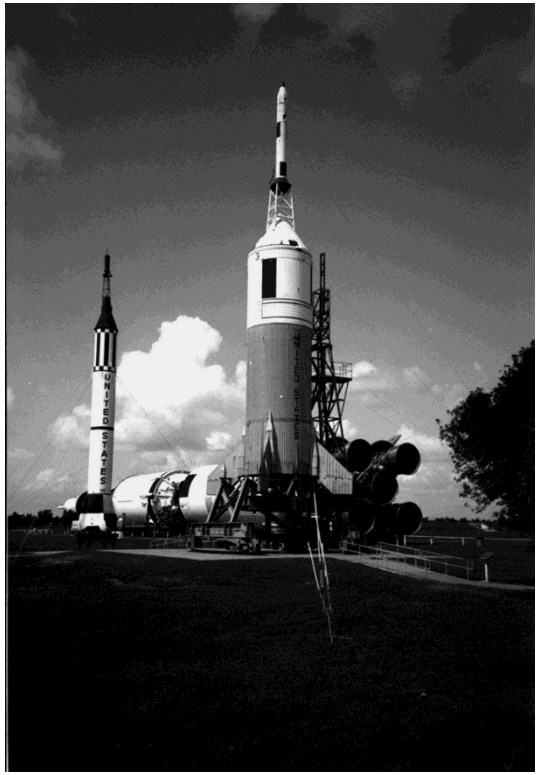


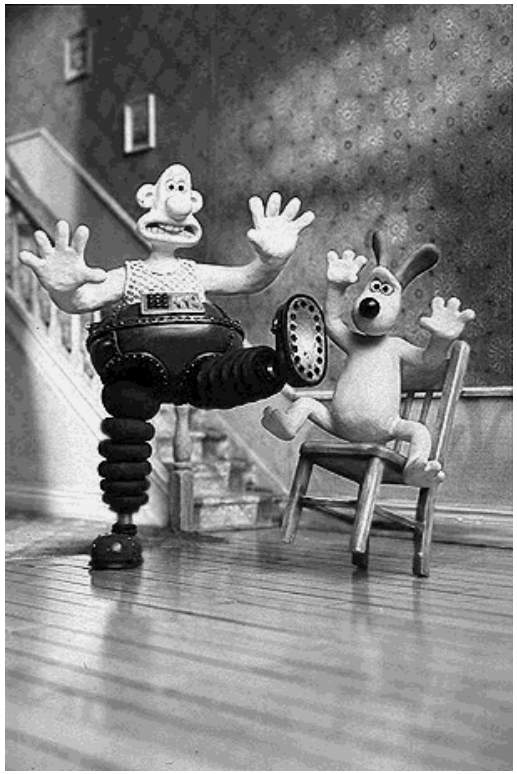



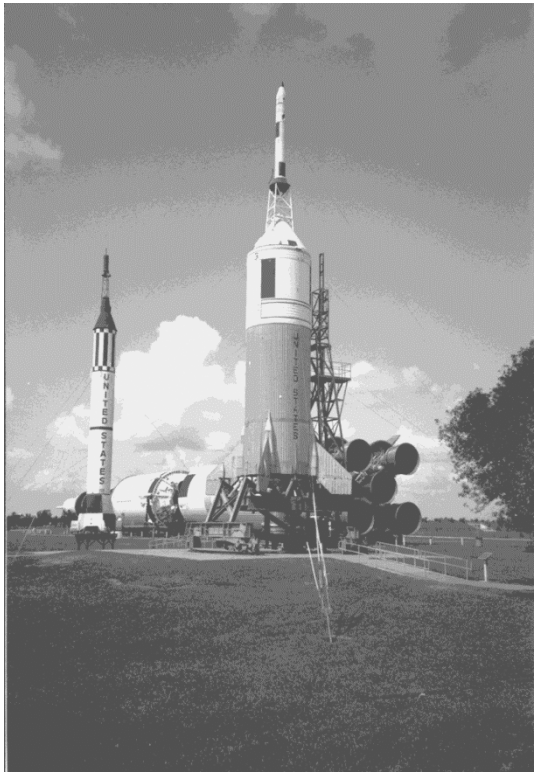
# Example: Histogram Matching



discolored

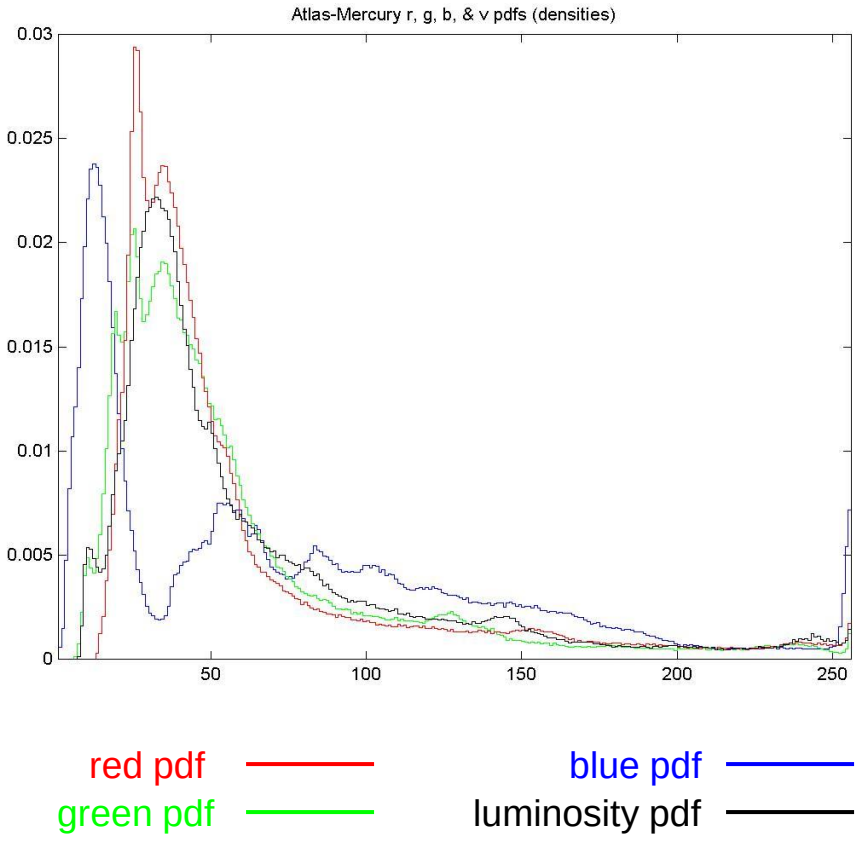


reference

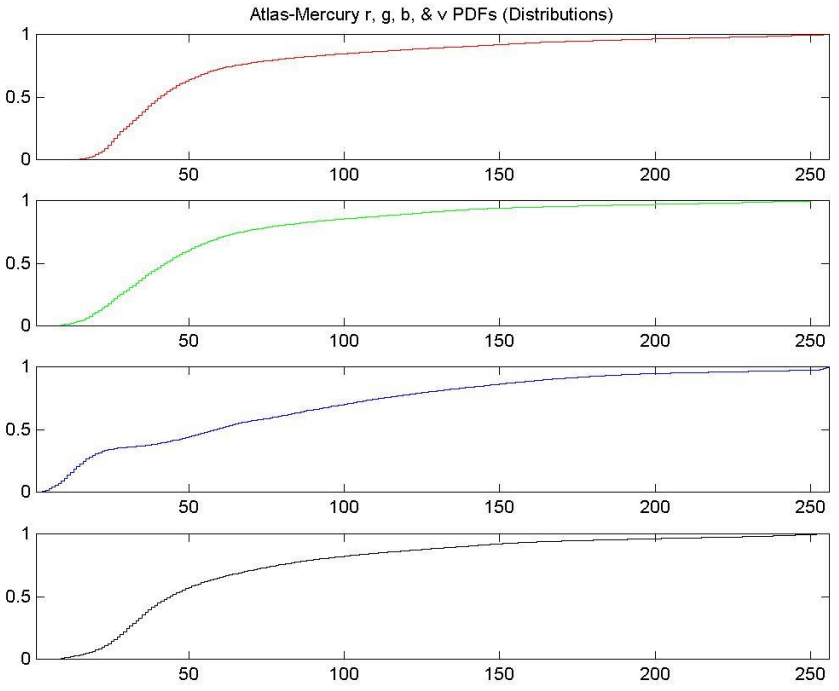


remapped

# Probability Density Functions (pdfs)

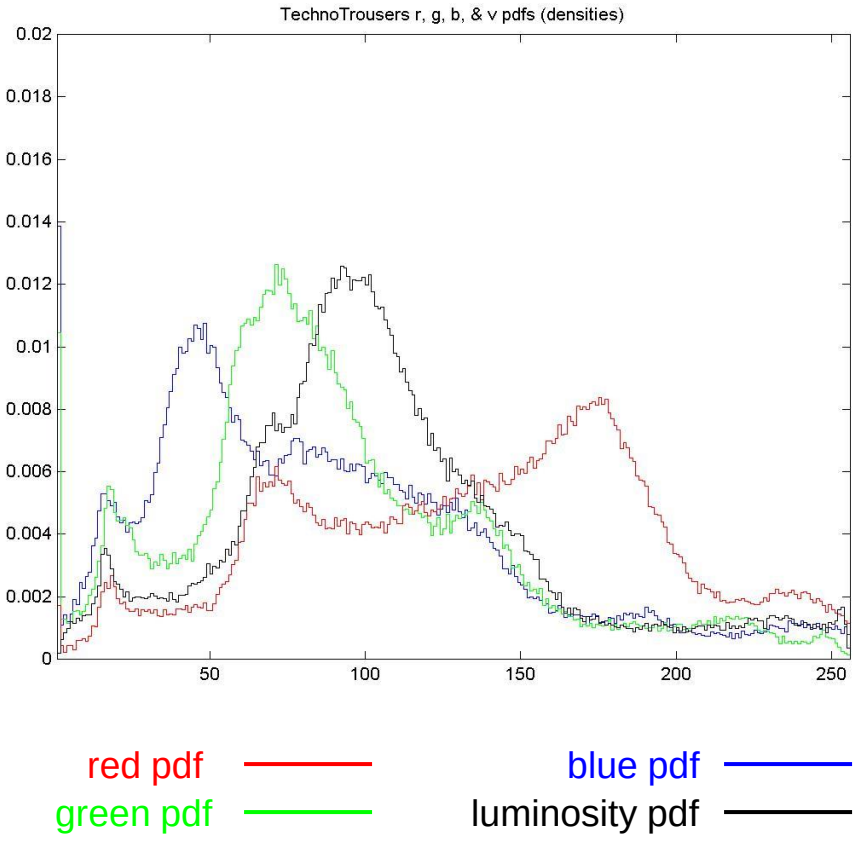



# Cumulative Distribution Functions (CDF)

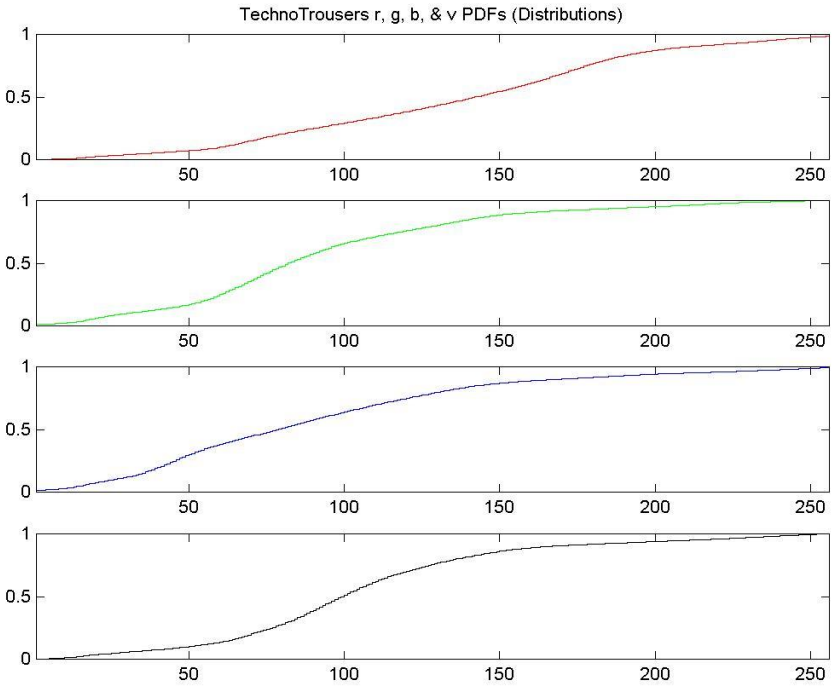



red CDF
green CDF
blue CDF
luminosity CDF

# Probability Density Functions (pdfs)

# Cumulative Distribution Functions (CDF)




red CDF
green CDF
blue CDF
luminosity CDF

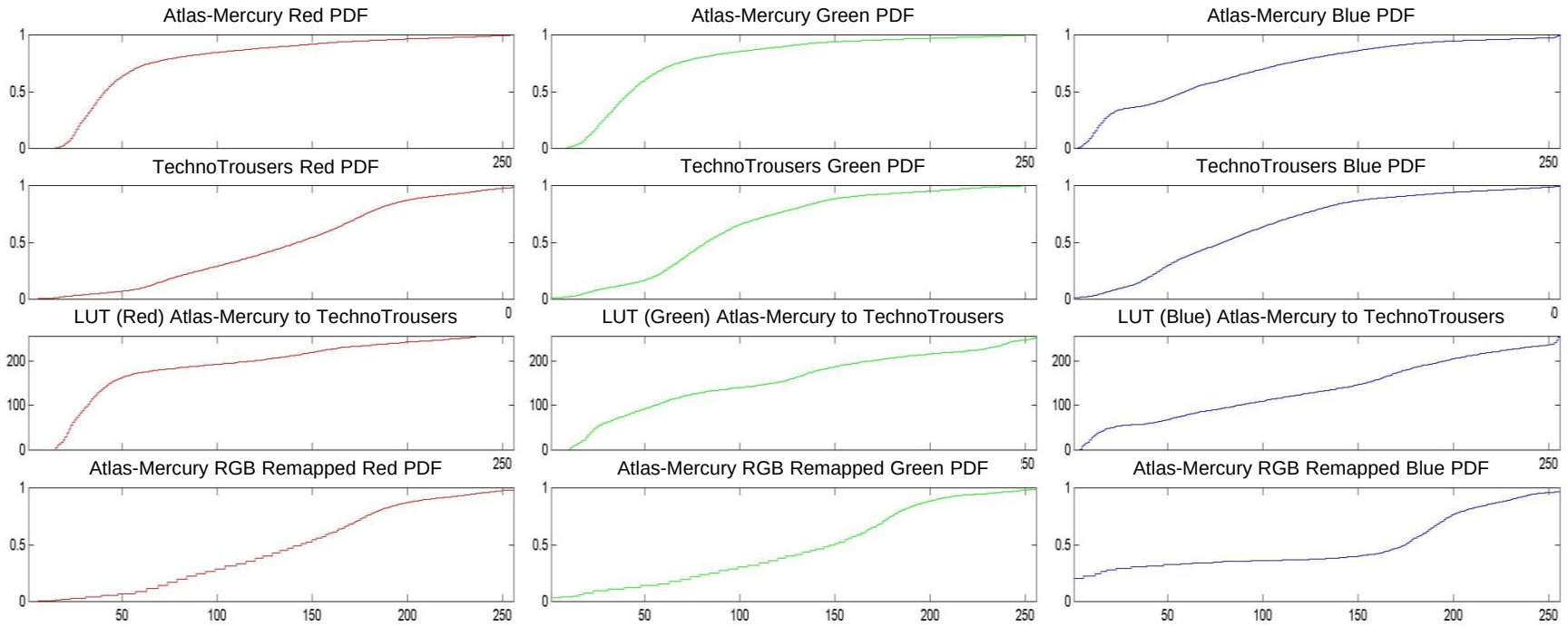
# Remap one Image to have the RGB CDF of Another

original “discolored”

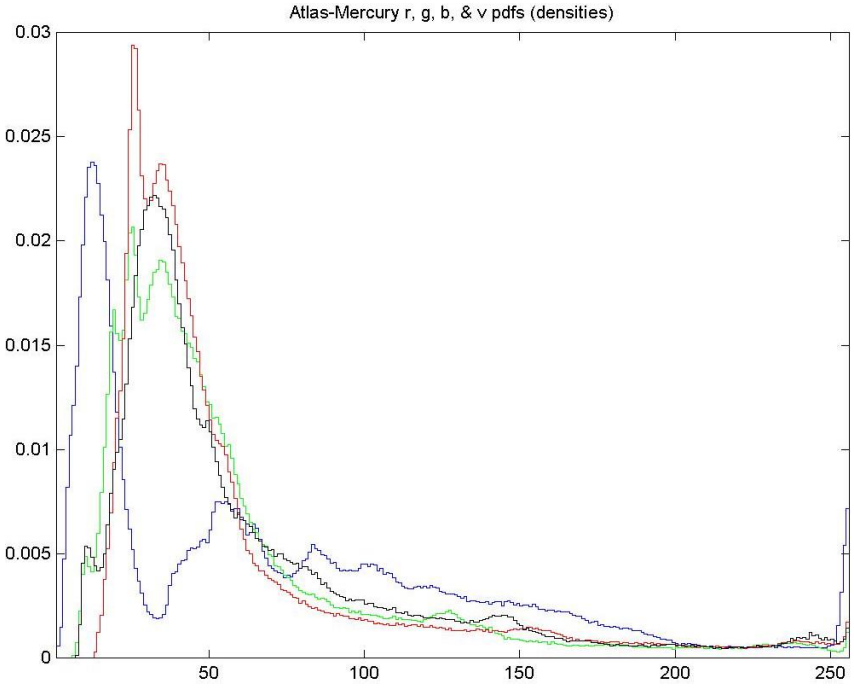
target “reference”

R, G, & B remapped

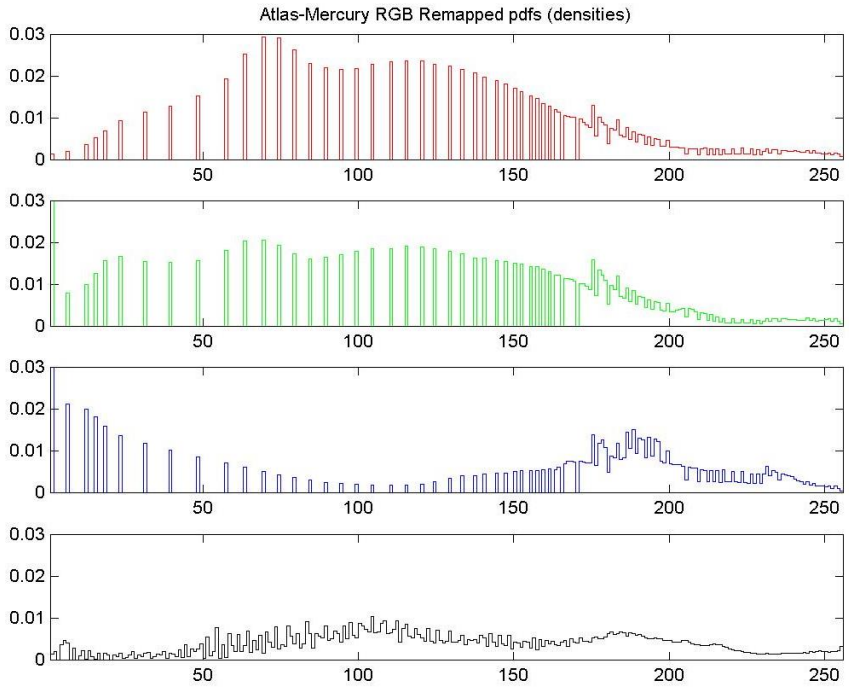
# RGB CDFs and the LUTs



# Effects of RGB Remapping on pdfs

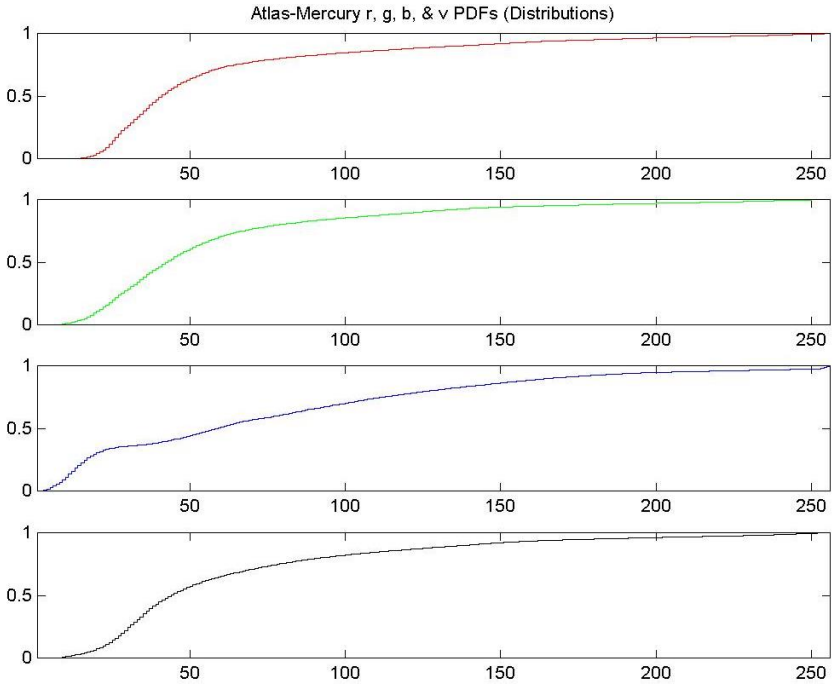



Before

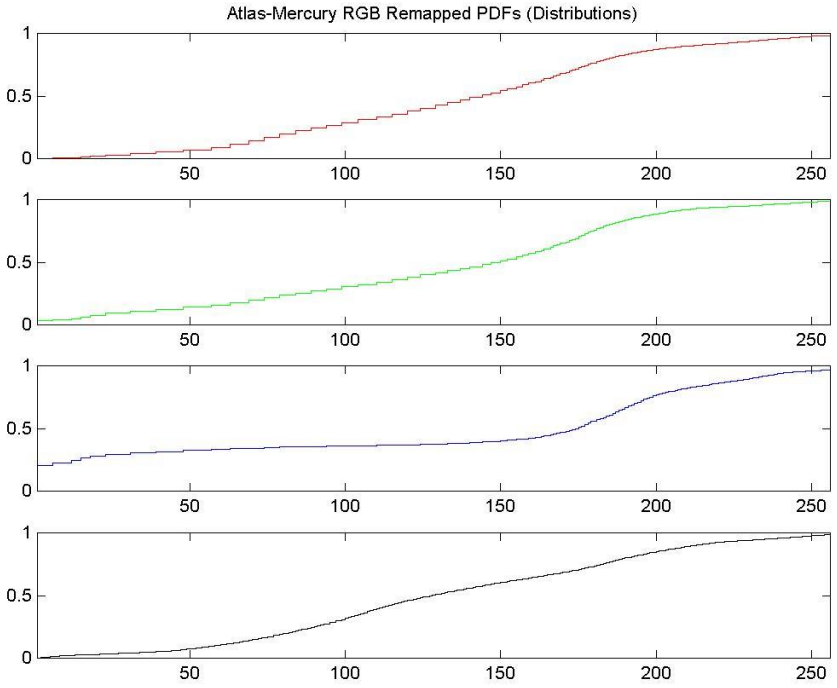



After

# Effects of RGB Remapping on CDFs




Before




After

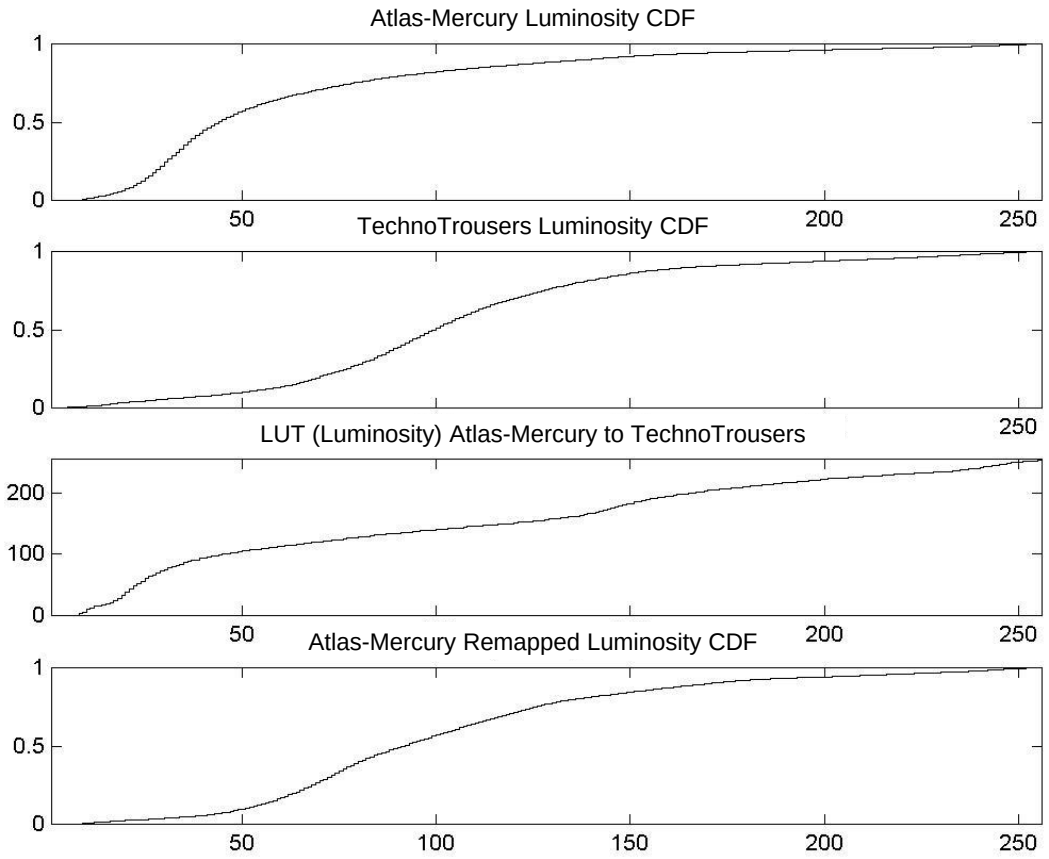
# Remap one Image to have the Lum. CDF of Another

original “discolored”

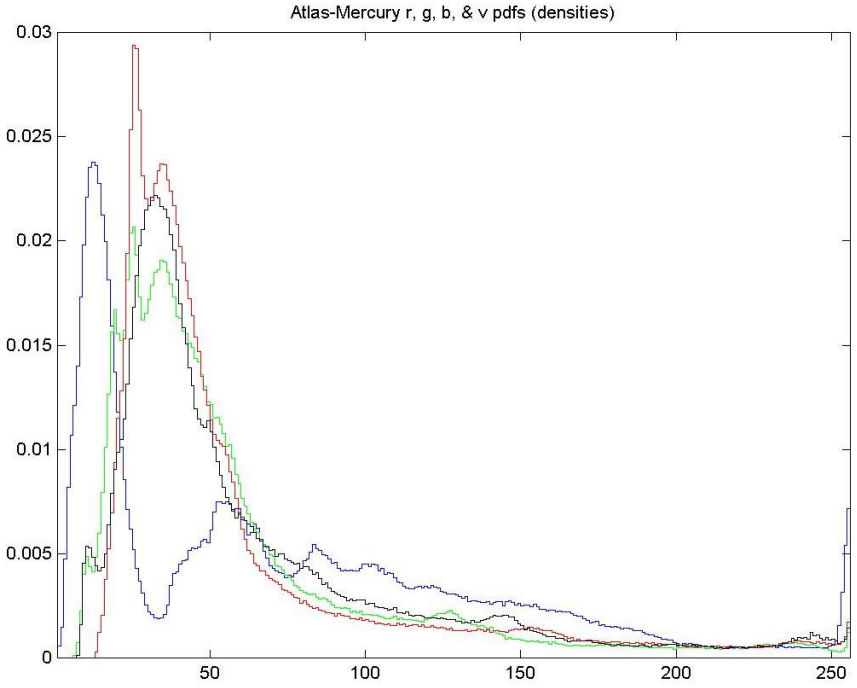
target “reference”

luminosity remapped

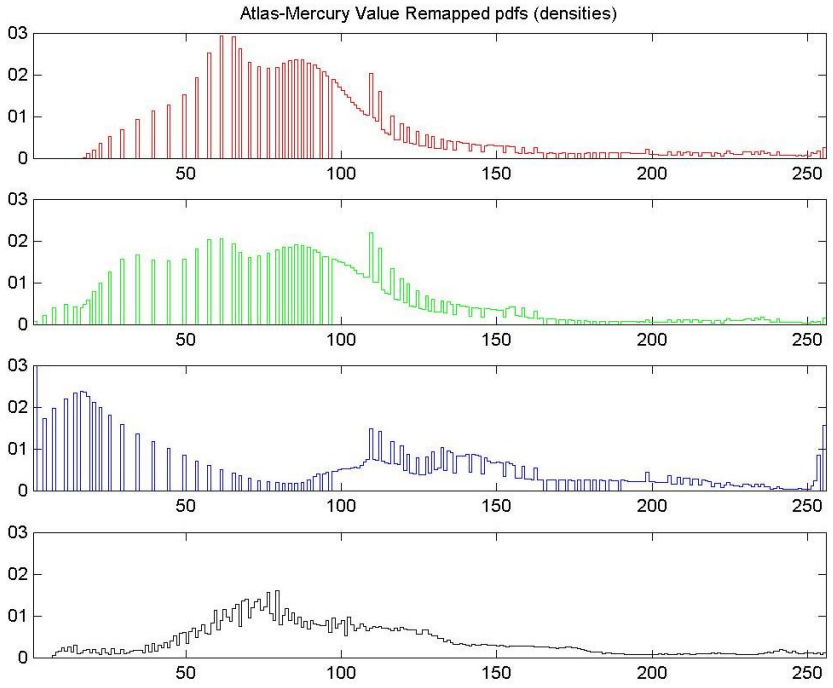
# Luminance CDFs and the LUT



# Effects of Luminance Remapping on pdfs

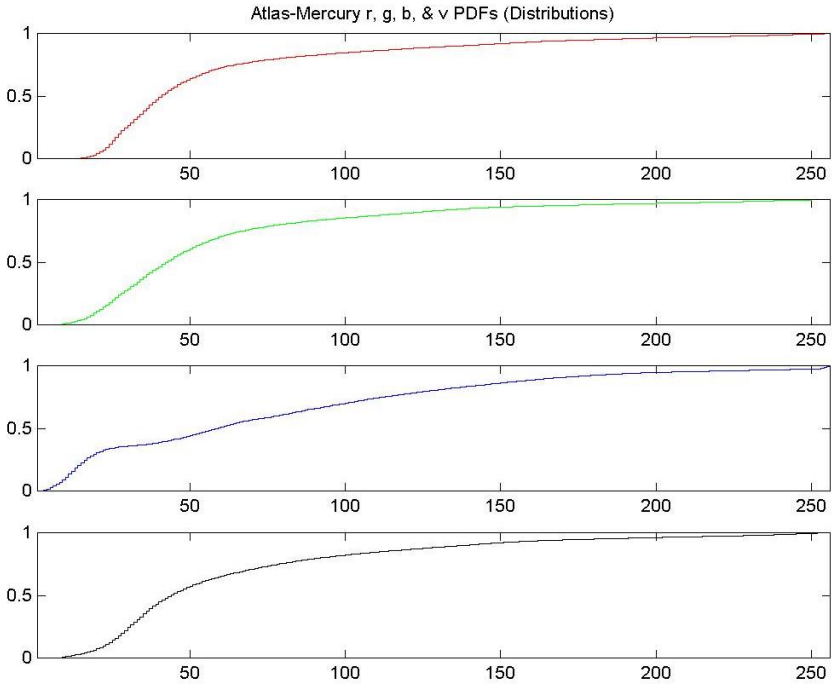



Before

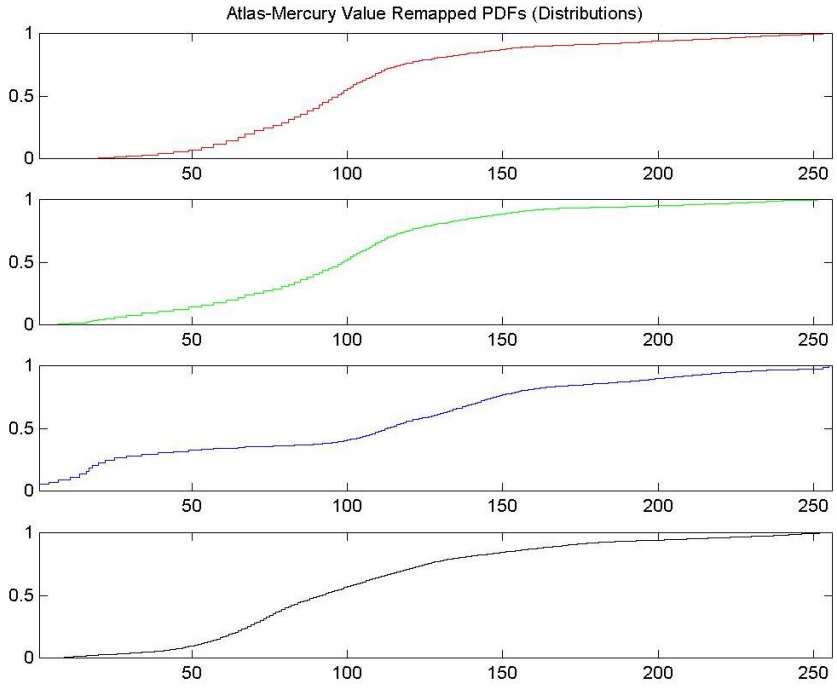



After

# Effects of Luminance Remapping on CDFs




Before




After

# Histogram Equalization Revisited

Our first attempt to equalize the color version of Kinkaku-ji by mapping each band through its own CDF LUT led to the unsatisfactory results below (left):

(a) Direct equalization of the color bands by mapping each band though its own CDF. $LUT_b = 255*CDF_{b.}$

(b) Equalization of the color bands by mapping each band though the CDF of the luminance image. $LUT_b = 255*CDF_L$.

# Histogram Equalization Revisited

Histogram matching presents another alternative: Match each color band CDF with the CDF from the luminance image. Then we get:

(a) Original image.

(b) Equalization by mapping band, *b*, though $LUT_b = 255*[CDF_L]^{-1}.*CDF_b$.

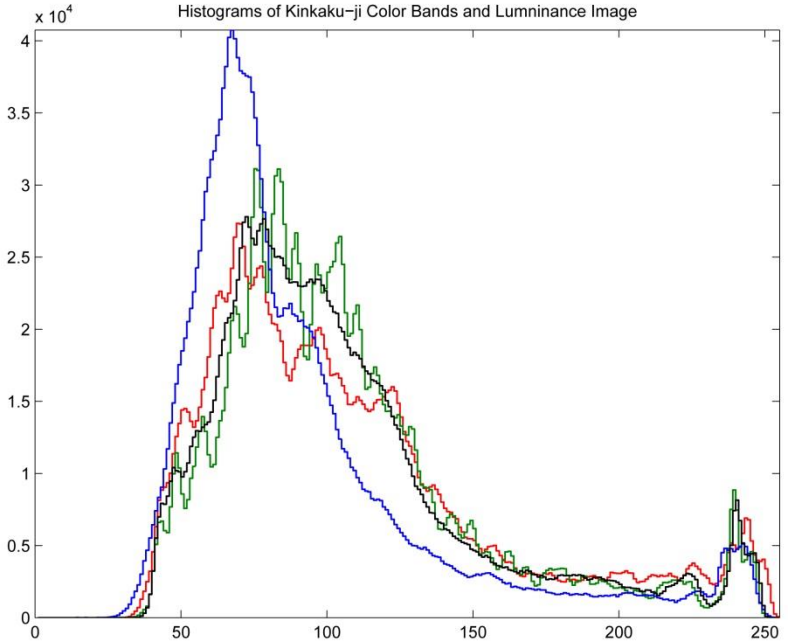
# EQ by matching each color-band to image luminance

In 6 (count 'em) 6 E-Z steps!

1. Convert image **I** into grayscale image, **L**, via your favorite weighting scheme.
2. Compute the 3 color histograms, $h_C$, $C \in \{R,G,B\}$ of **I** and the histogram, $h_L$, of **L**.
3. Compute the 4 probability density functions, $p_C$, $C \in \{R,G,B,L\}$, from the $h_C$.
4. Compute the 4 cumulative distribution functions, $H_C$, $C \in \{R,G,B,L\}$, from the $p_C$.
5. Generate 3 lookup tables, $T_C$, $C \in \{R,G,B\}$, by matching $H_R$ to $H_L$, $H_G$ to $H_L$, & $H_B$ to $H_L$.
6. Map each image band {**R**,**G**,**B**} through its corresponding lookup table $T_C$.

# EQ by matching each color-band to image luminance

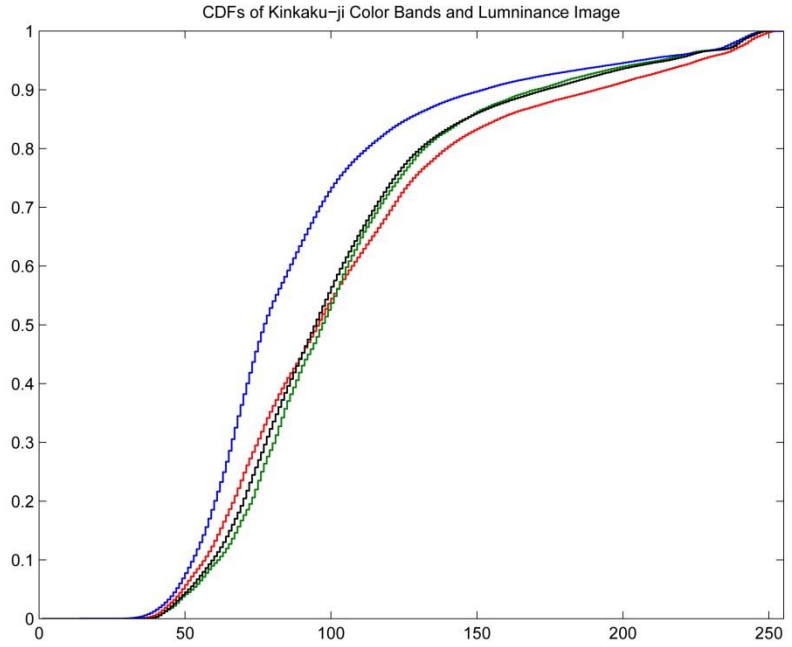
Original Color Image




Color & Lum. Histograms

# EQ by matching each color-band to image luminance

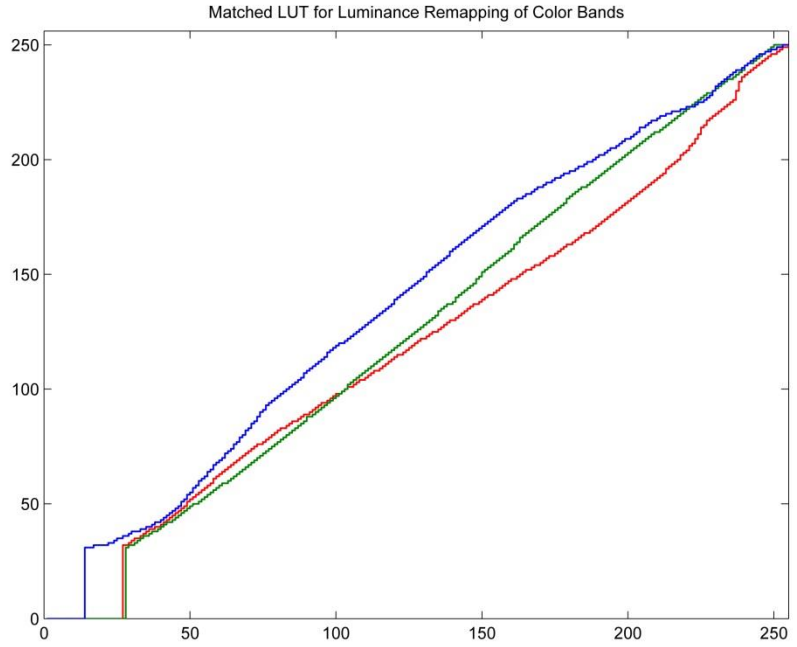
Original Color Image




Color & Lum. CDFs

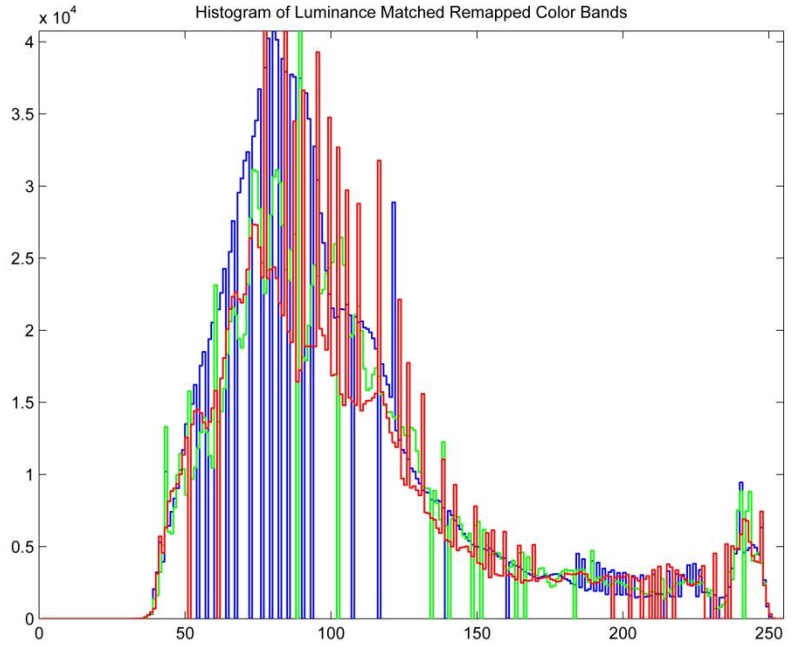
# EQ by matching each color-band to image luminance

Original Color Image




Equalization LUT

# EQ by matching each color-band to image luminance




Luminance matched image

Histo of lum matched image

Note the desaturation of the colors.

# Method 1: Match each band, use 3 different LUTs

Original color image

Each band eq'd separately

Right image: equalized each of the three bands separately (pp. 47-52).

# Method 2: Map each band through 1 luminance LUT

Original color image

Each band eq'd through lum.

Right image: mapped each band through the LUT from luminance CDF (pp. 53-58).

# Method 3: Match each band to luminance, 3 LUTs

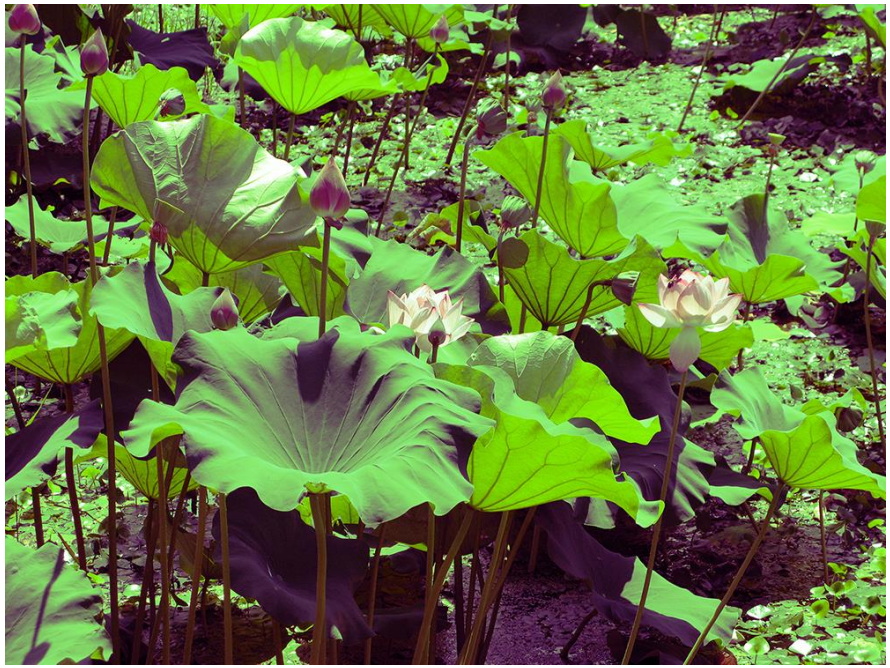
Original Color Image

Each band matched to lum.

Right image: matched each band to the luminance (pp. 90-95).

# Histogram Matching for Image Restoration

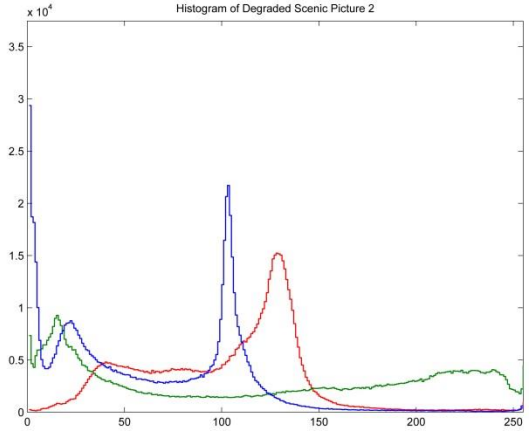
Degraded image

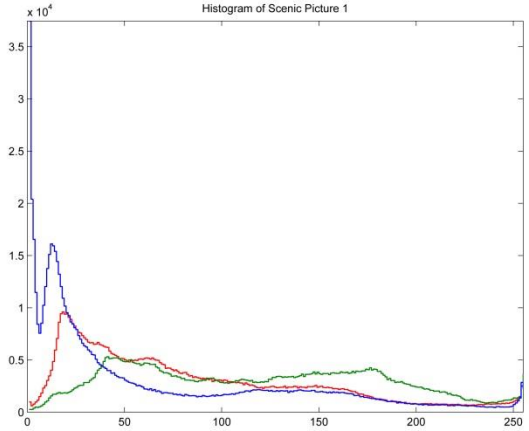


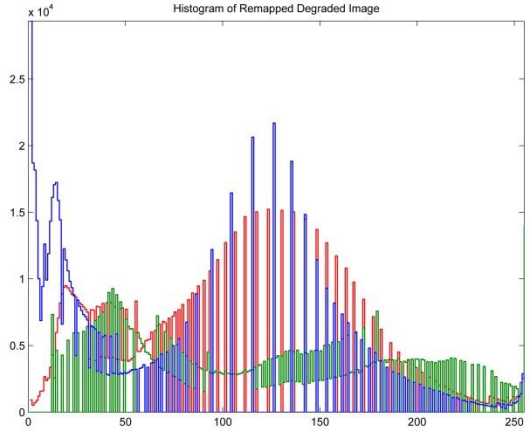
Another image of the same scene, not degraded

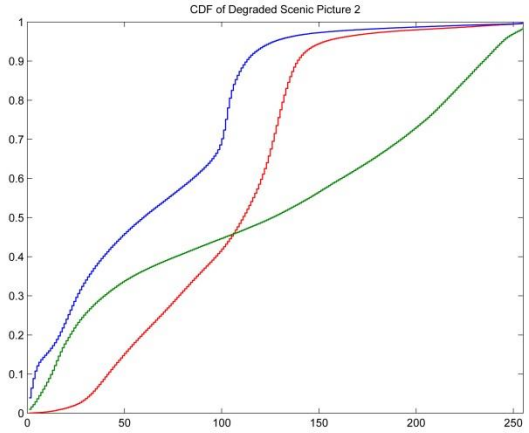
Lotus Flowers at Turtle Head Park, Lake Tai, Wuxi, Jiangsu Province, China.
莲花 在 黿头渚 太湖 无锡市 江苏省 中国. Photos by R. A .Peters II, July 2013.

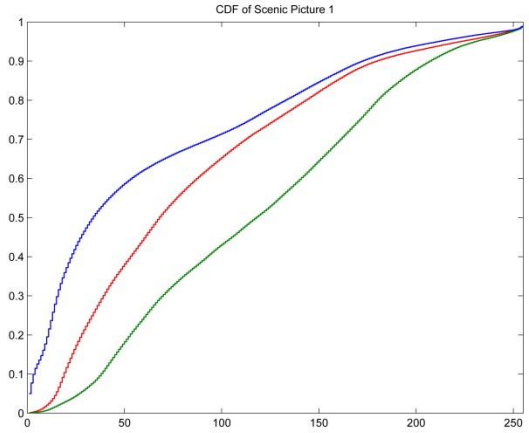
# Histogram Matching for Image Restoration

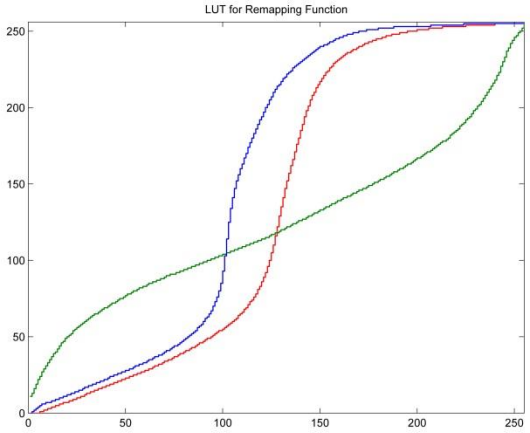


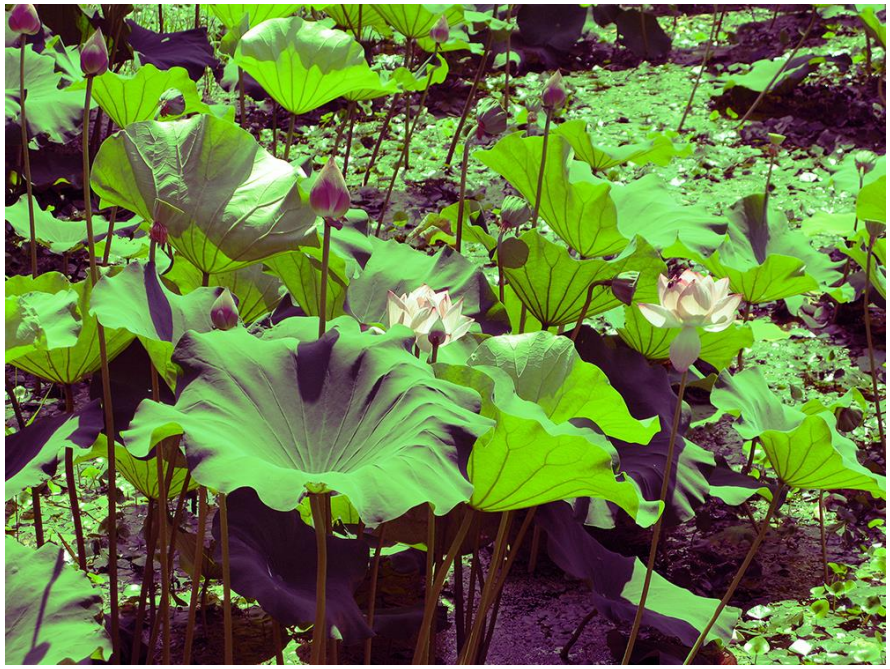



# Histogram Matching for Image Restoration

Degraded image



Another image of the same scene, not degraded

Lotus Flowers at Turtle Head Park, Lake Tai, Wuxi, Jiangsu Province, China.
莲花 在 黿头渚 太湖 无锡市 江苏省 中国. Photos by R. A .Peters II, July 2013.

# Histogram Matching for Image Restoration

Remapped degraded image

Another image of the same scene, not degraded

Lotus Flowers at Turtle Head Park, Lake Tai, Wuxi, Jiangsu Province, China.
莲花 在 黿头渚 太湖 无锡市 江苏省 中国. Photos by R. A .Peters II, July 2013.

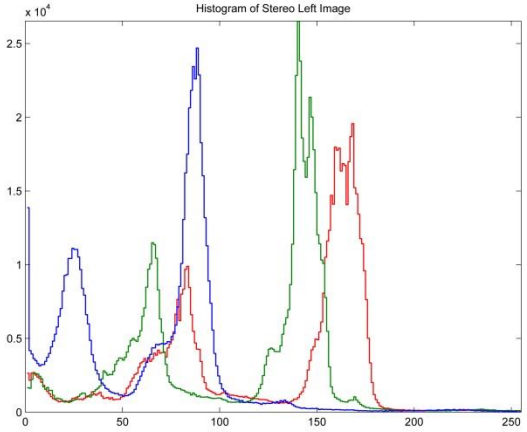
# Histogram Matching for Image Restoration

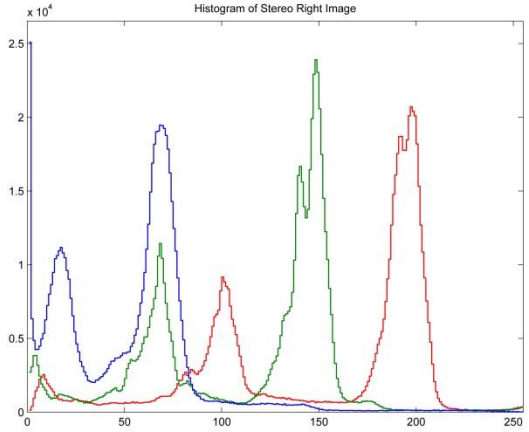
Left Image

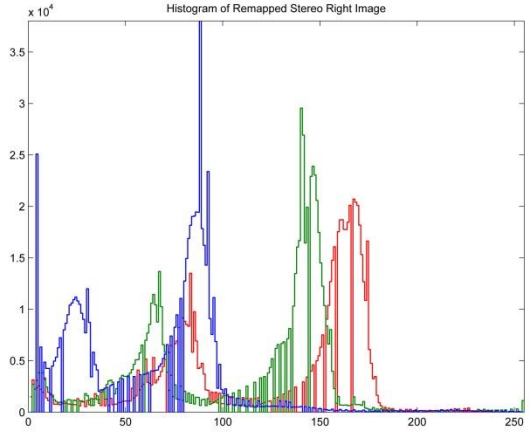
Right Image

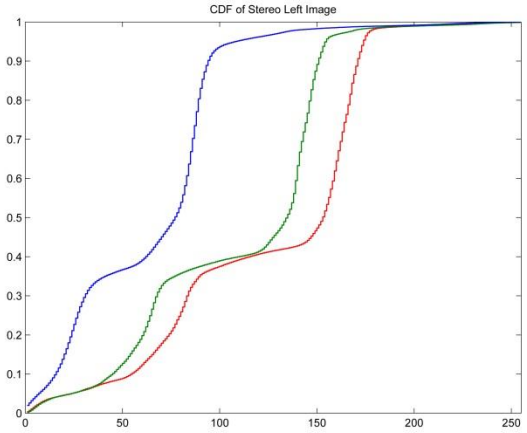
Images from a stereo pair of inexpensive web cams. Such cameras have different color characteristics of-the-shelf. Once can be corrected to match the other using histo. matching.

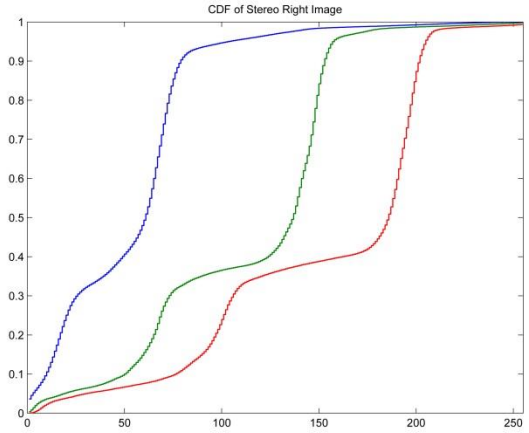
# Histogram Matching for Image Restoration

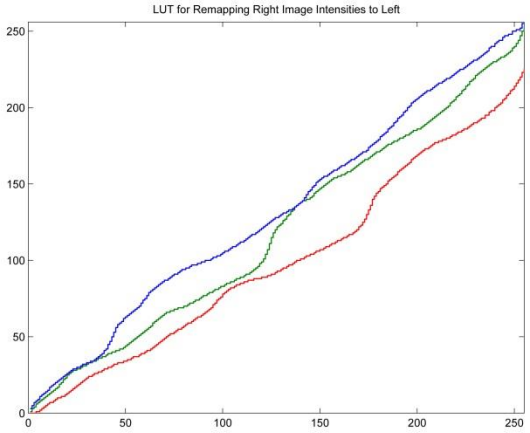


# Histogram Matching for Image Restoration

Left Image

Right Image

Images from a stereo pair of inexpensive web cams. Such cameras have different color characteristics of-the-shelf. Once can be corrected to match the other using histo. matching.

# Histogram Matching for Image Restoration

Left Image

Right Image

Right image histogram matched to left image.