Pengolahan Citra - Pertemuan III – Image Enhancement

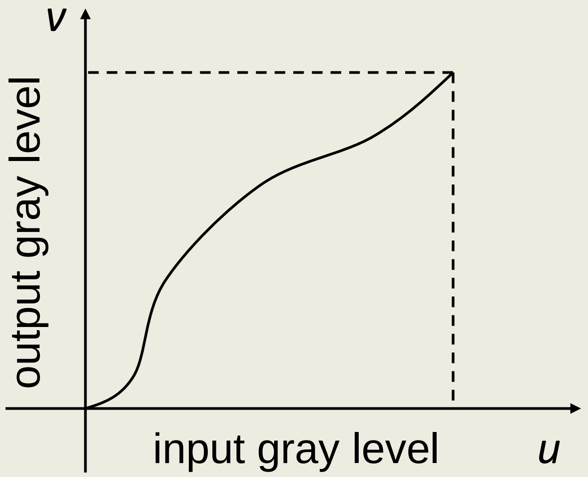
Nana Ramadijanti
Politeknik Elektronika Negeri Surabaya

Materi:

1. Pemrosesan Piksel
2. Histogram Gray Scale
3. Histogram RGB
4. Pemrosesan Piksel melalui Mapping Fungsi
5. Pemrosesan Piksel melalui Look-up Table
6. Brightness
7. Contrass
8. Gamma
9. Histogram Equalisasi
10. Histogram Matching

Operasi Piksel/ Transformasi Intensitas

- Ide dasar :
  - Operasi “Zero memory”
    - Setiap output bergantung pada intensitas input pada piksel yg ada
  - Memetakan warna level gray atau warna level *u* ke level baru *v*, misal: $v = f(u)$
  - Tidak memberikan informasi baru
  - Tapi dapat meningkatkan penampilan visual atau membuat fitur mudah untuk dideteksi



- Contoh-1: Transformasi Warna Koordinat
  - RGB setiap piksel⇨ komponen luminance + chrominance ⇨ dll
- Contoh-2: Kuantisasi scalar
  - Kuantisasi piksel luminance/color dengan jumlah bit yg lebih sedikit

Pemrosesan Piksel

- Pemrosesan piksel adalah mengubah nilai piksel sebagai fungsi dari nilainya sendiri;
- Pemrosesan piksel tidak bergantung pada nilai piksel tetangganya.

Pemrosesan Piksel pada Citra

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

Point Processing

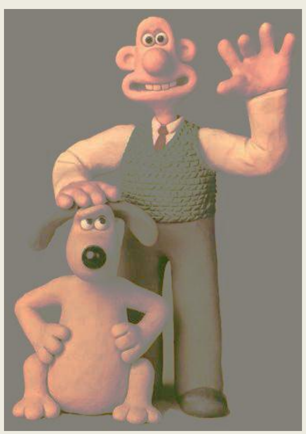
- gamma

- brightness

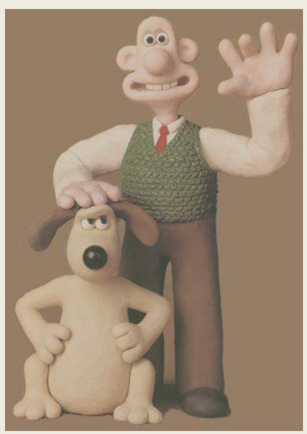
original

+ brightness

+ gamma



histogram mod



- contrast

original

+ contrast

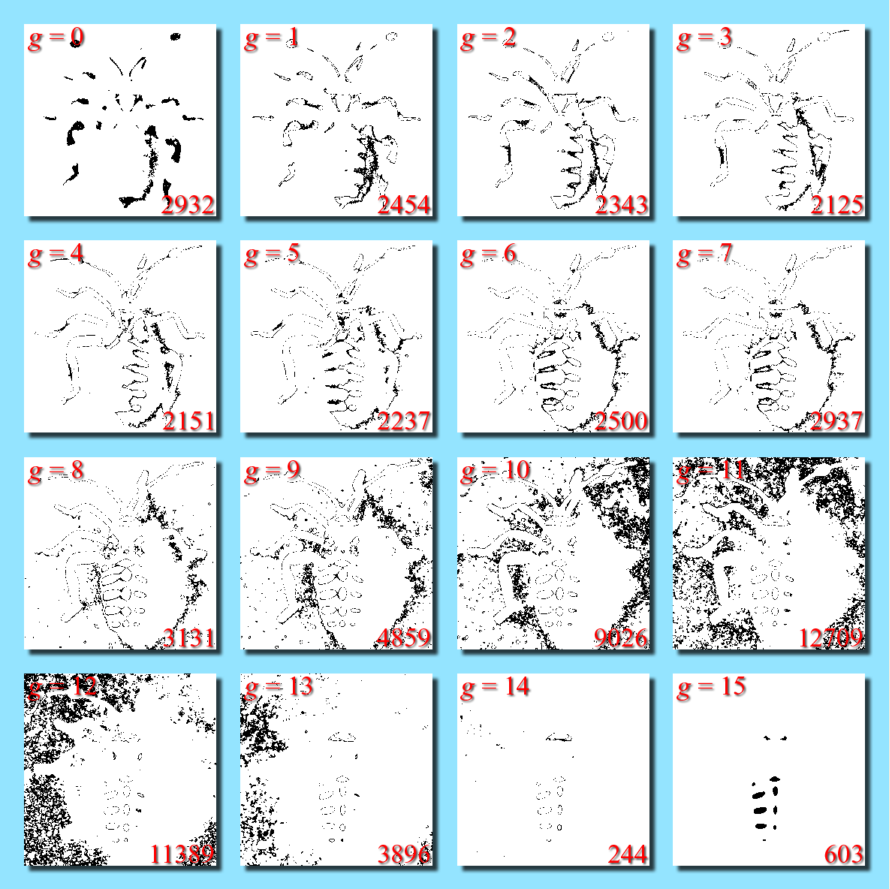
histogram EQ

Histogram Citra Gray Scale

- I adalah citra grayscale.
- $I(r,c)$ adalah 8-bit integer diantara 0 dan 255.
- Histogram, h_I, dari I:
  - 256-element array, h_I
  - $h_I(g)$, untuk g = 0, 1, 2, …, 255, adalah integer
  - $h_I(g)$ = jumlah piksel pada I dengan nilai = g.

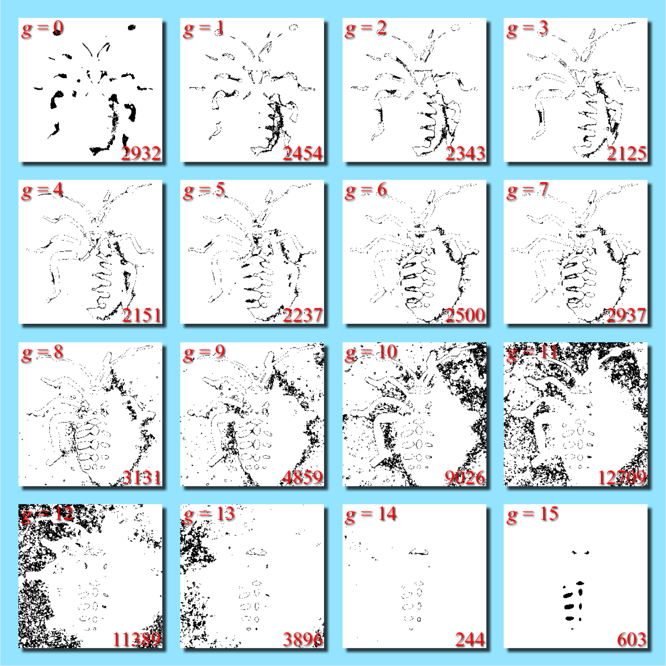
Histogram Citra Gray Scale

16-level (4-bit) citra

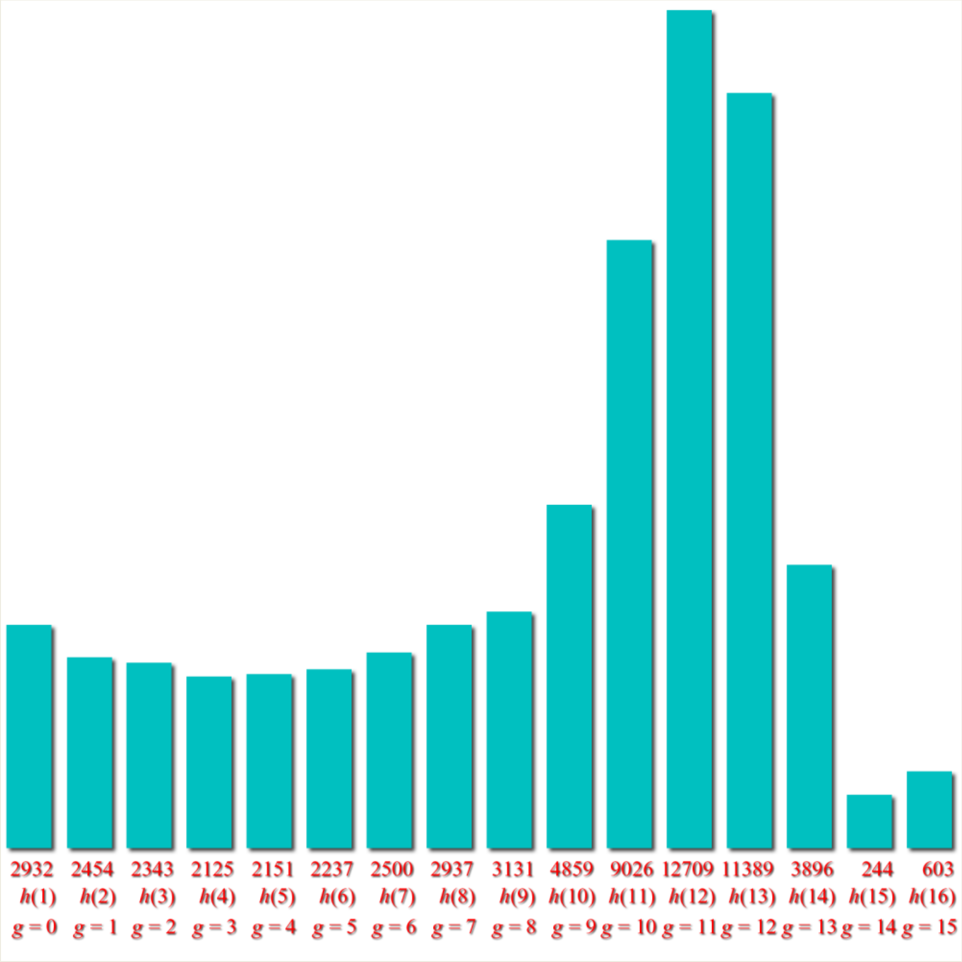


Tanda hitam menunjukkan piksel dengan intensitas g

The Histogram of a Grayscale Image

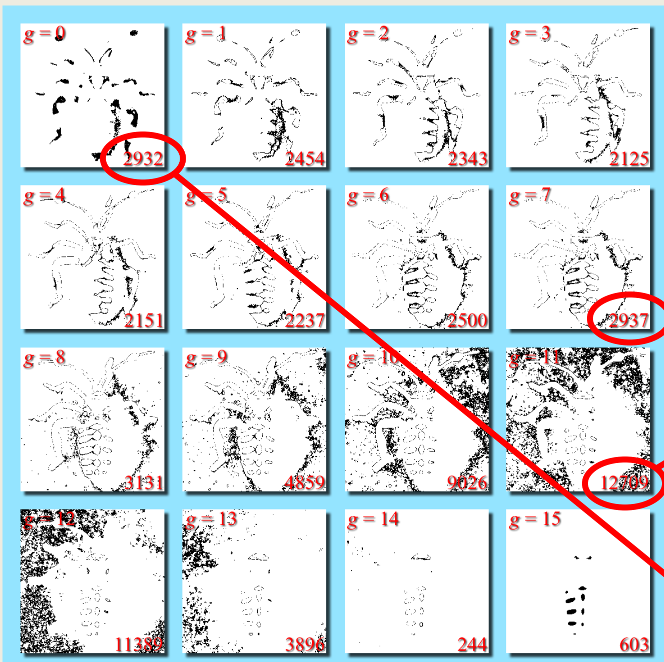


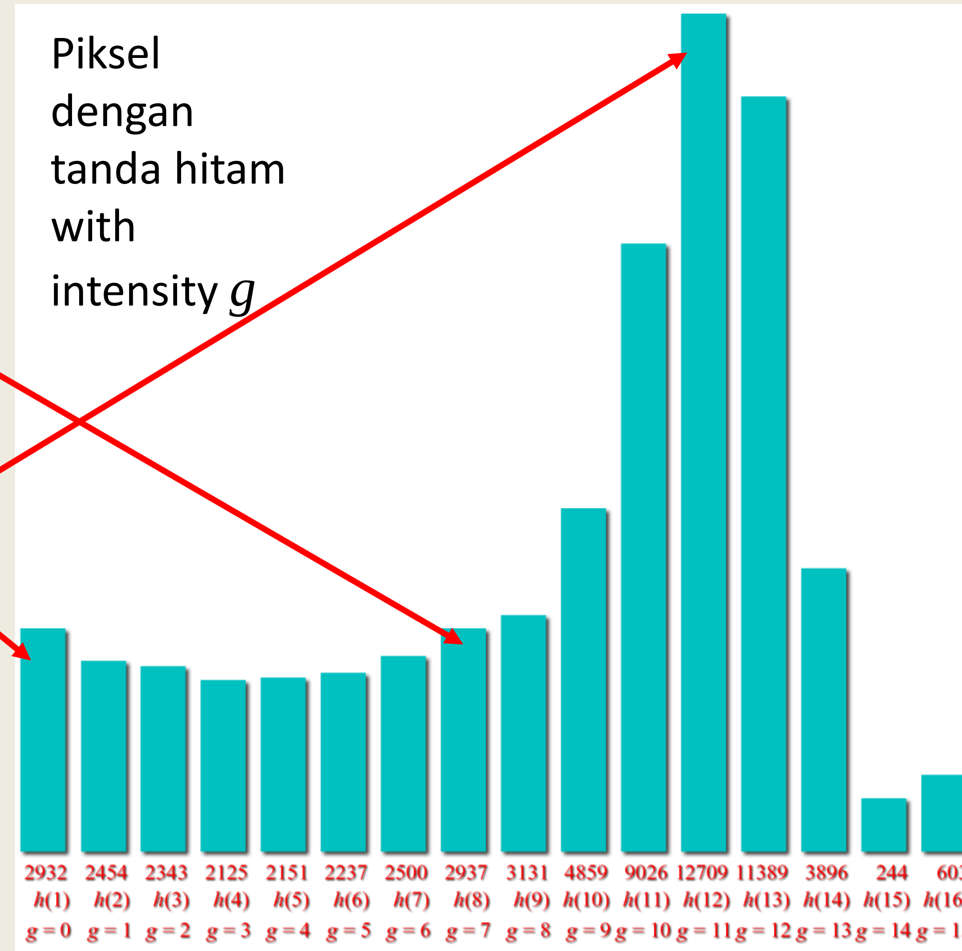
Piksel tanda hitam dengan intensitas g



Histogram :
Jumlah piksel dengan intensitas g

The Histogram of a Grayscale Image

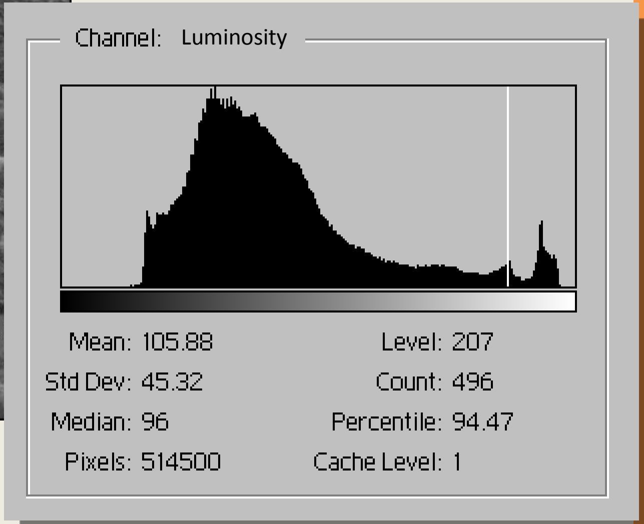




Plot of histogram:
Jumlah piksel dengan intensitas g

The Histogram of a Grayscale Image

$h_I(g+1) =$ the number of pixels in I with graylevel g.

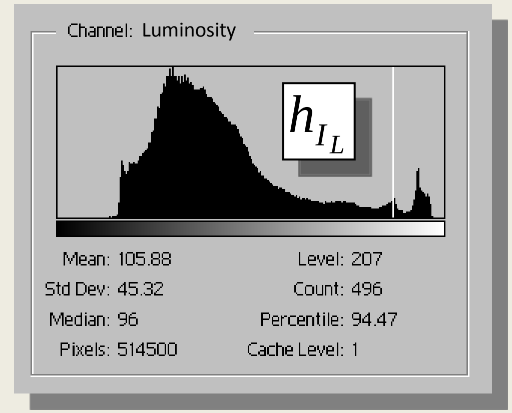


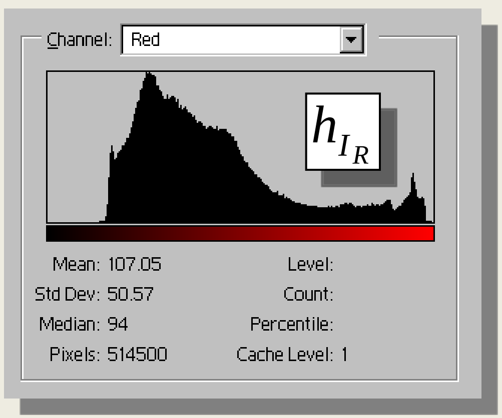
The Histogram of a Color Image

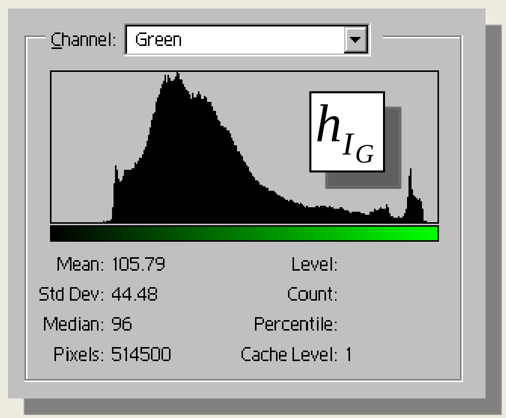
- If I is a 3-band image (truecolor, 24-bit)
- then I(*r*,*c*,*b*) is an integer between 0 and 255.
- Either I has 3 histograms:
  - $h_R(g+1)$ = # of pixels in I(:,:,1) with intensity value g
  - $h_G(g+1)$ = # of pixels in I(:,:,2) with intensity value g
  - $h_B(g+1)$ = # of pixels in I(:,:,3) with intensity value g
- or 1 vector-valued histogram, $h(g,1,b)$ where
  - $h(g+1,1,1)$ = # of pixels in I with red intensity value g
  - $h(g+1,1,2)$ = # of pixels in I with green intensity value g
  - $h(g+1,1,3)$ = # of pixels in I with blue intensity value g

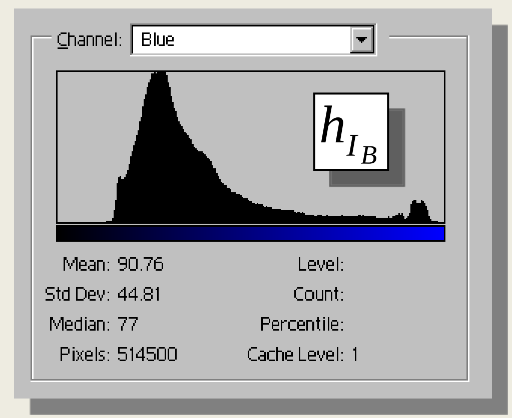
The Histogram of a Color Image

There is one histogram per color band R, G, & B. Luminosity histogram is from 1 band = (R+G+B)/3

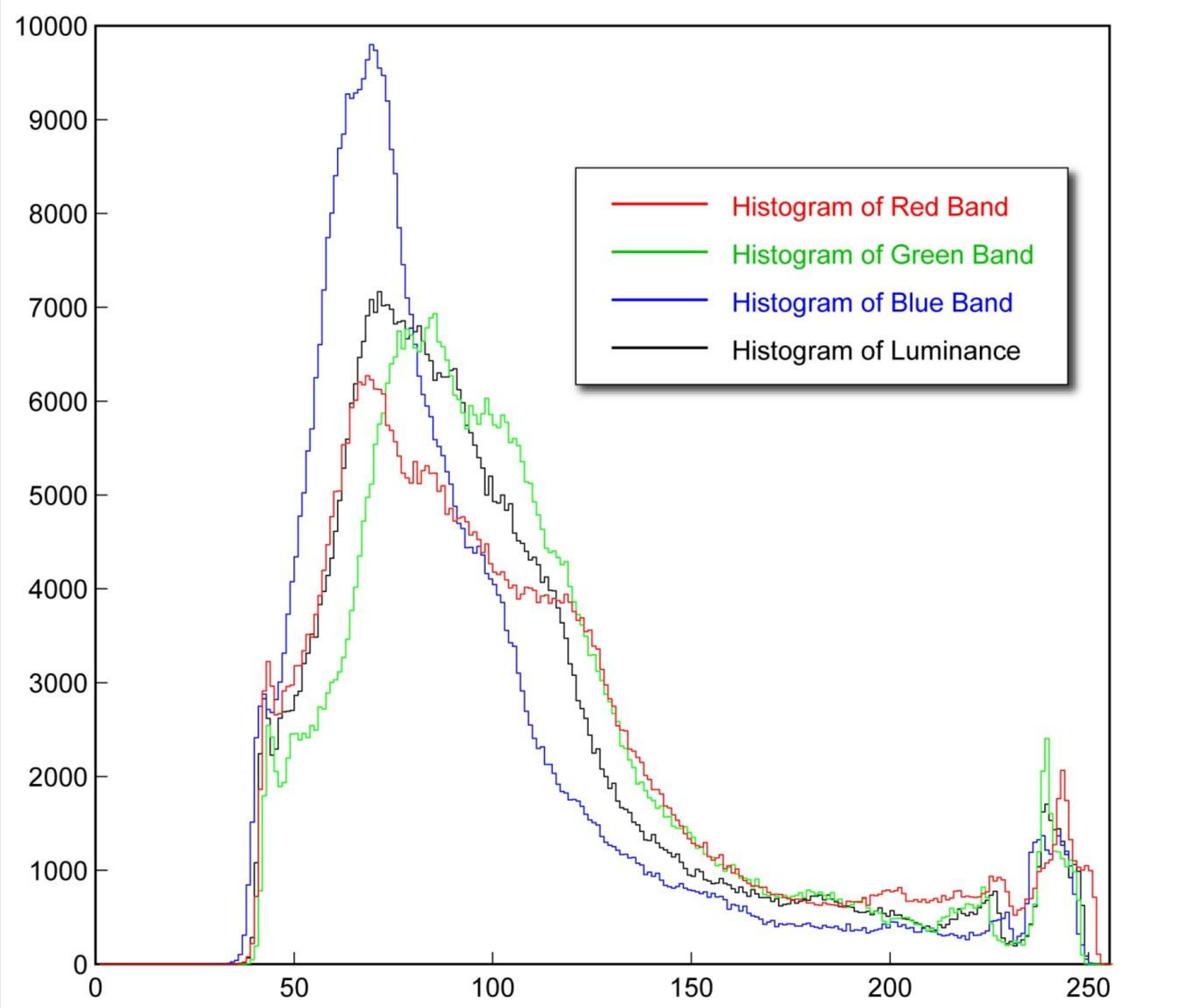








The Histogram of a Color Image



Value or Luminance Histograms

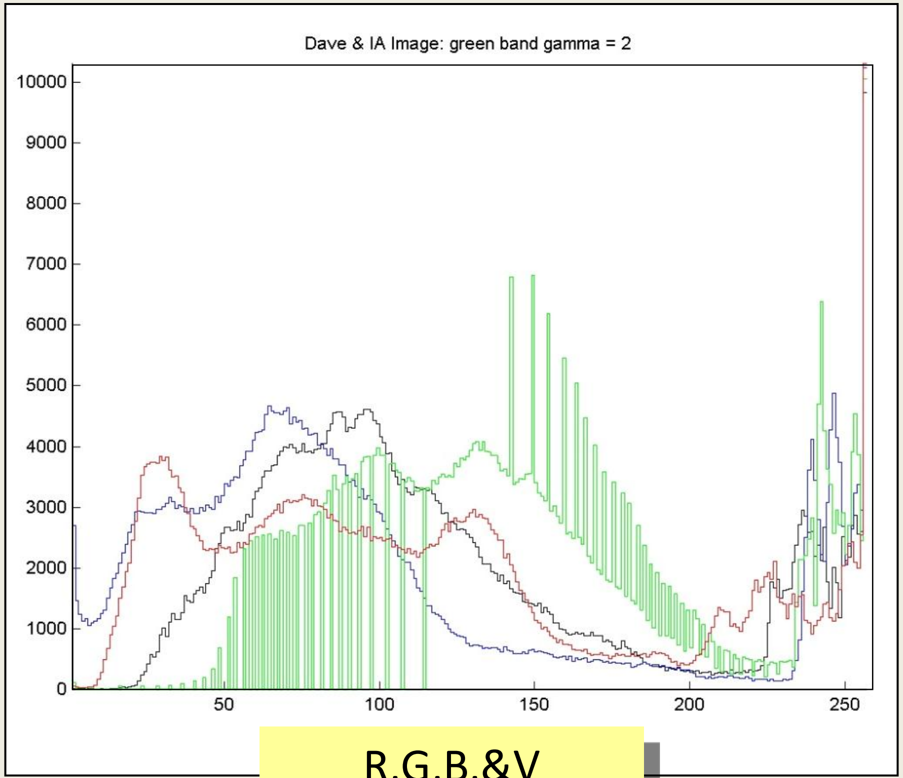
The value histogram of a 3-band (truecolor) image, *I*, is the histogram of the value image,

$$V(r,c)=\frac{1}{3}\left[R(r,c)+G(r,c)+B(r,c)\right]$$

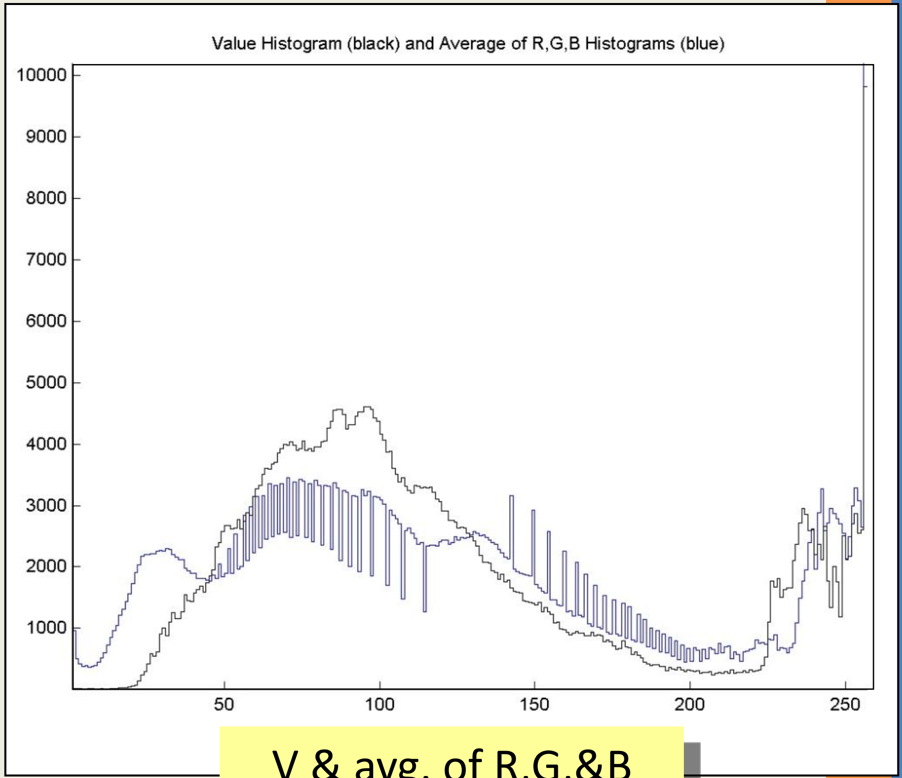
Where *R*, *G*, and *B* are the red, green, and blue bands of *I*.
The luminance histogram of *I* is the histogram of the luminance image,

$$L(r,c)=0.299\cdot R(r,c)+0.587\cdot G(r,c)+0.114\cdot B(r,c)$$

Value Histogram vs. Average of R,G,&B Histograms



R,G,B,&V histograms



V & avg. of R,G,&B histograms

Multi-Band Histogram Calculator in Matlab

```
% Multi-band histogram calculator
function h=histogram(I)

[R C B]=size(I);

% allocate the histogram
h=zeros(256,1,B);

% range through the intensity values
for g=0:255
   h(g+1,1,:) = sum(sum(I==g)); % accumulate
end

return;
```

Multi-Band Histogram Calculator in Matlab

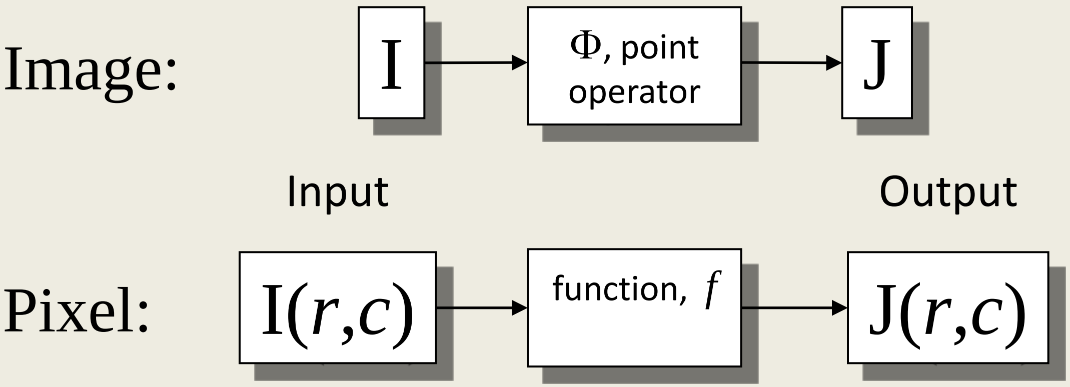
```
% Multi-band histogram calc
function h=histogram(I)

[R C B]=size(I);

% allocate the histogram
h=zeros(256,1,B);

% range through the intensi
for g=0:255
   h(g+1,1,:) = sum(sum(I==g)); % accumulate
end
```

Loop through all intensity levels (0-255)
Tag the elements that have value *g*.
The result is an RxCxB *logical* array that has a 1 wherever $I(r,c,b) = g$ and 0's everywhere else.
Compute the number of ones in each band of the image for intensity *g*.
Store that value in the 256x1xB histogram at $h(g+1,1,b)$.

If B==3, then h(*g*+1,1,:) contains 3 numbers: the number of pixels in bands 1, 2, & 3 that have intensity *g*.

sum(sum(I==*g*)) computes one number for each band in the image.

Point Ops via Functional Mappings



$$J = \Phi[I]$$

If $I(r,c) = g$

and $f(g) = k$

then $J(r,c) = k$.

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*) where *I*(*r*,*c*) = *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

$J(r,c) = f\left(I(r,c)\right),$
for all pixels locations (r,c).

Three-band Image

$J(r,c,b) = f\left(I(r,c,b)\right),$ or
$J(r,c,b) = f_b\left(I(r,c,b)\right),$
for $b = 1,2,3$ and all (r,c).

Point Ops via Functional Mappings

One-band Image

$$J(r,c) = f\big(I(r,c)\big),$$
for all pixels locations (r,c).

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

Three-band Image

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

… each band is mapped through a separate func-tion, f_b.

Point Operations using Look-up Tables

A look-up table (LUT) implements a functional mapping.

If $k = f(g)$,

for $g = 0, \ldots, 255$,

and if k takes on values in $\{0, \ldots, 255\}$, …

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

To remap a 1-band image, I, to J:

$$J = \text{LUT}(I + 1)$$

Point Operations using Look-up Tables

If I is 3-band, then

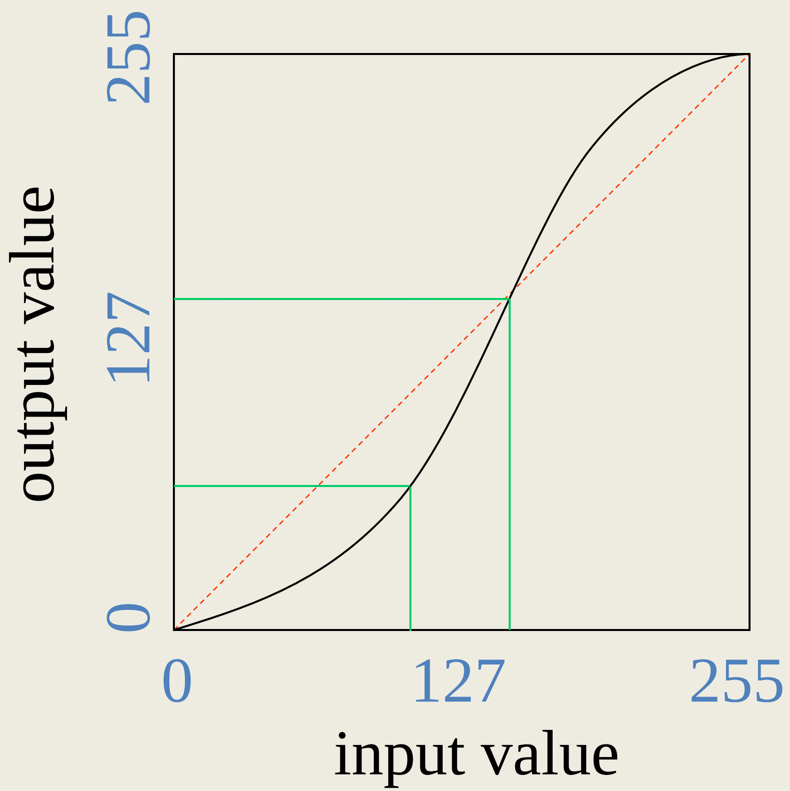
a) each band is mapped separately using the same LUT for each band *or*

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

$$\text{a)}\quad J = \text{LUT}(I+1), \textit{or}$$

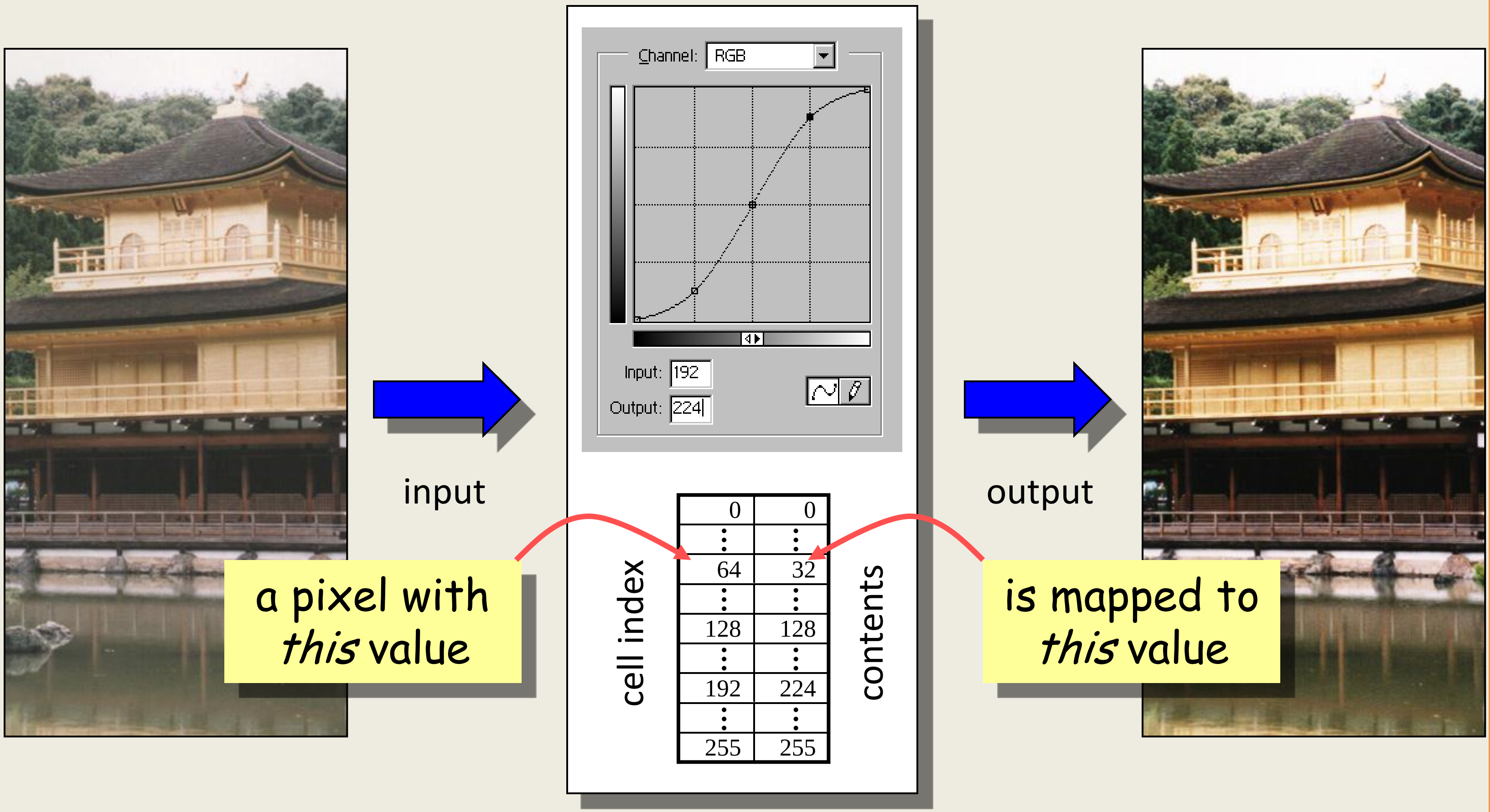
$$\text{b)}\quad J(:,:,b) = \text{LUT}_b\left(I(:,:,b)+1\right) \text{ for } b = 1,2,3.$$

Point Operations = Look-up Table Ops



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

Look-Up Tables



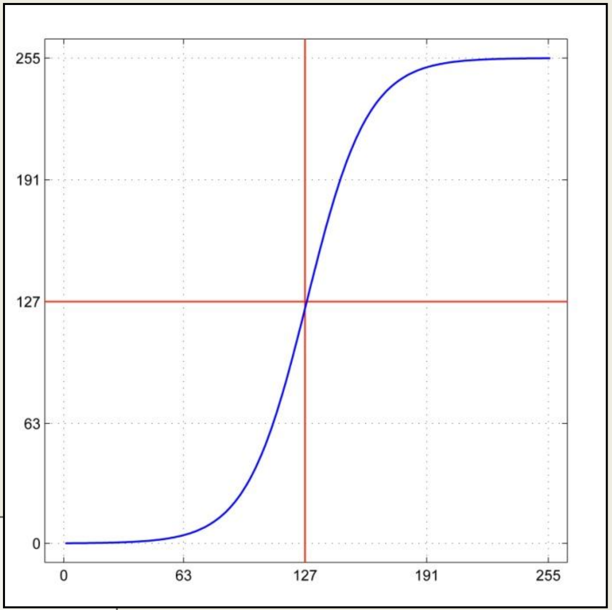
How to Generate a Look-Up Table

For example:

Let $a = 2.$

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

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



Or in Matlab:

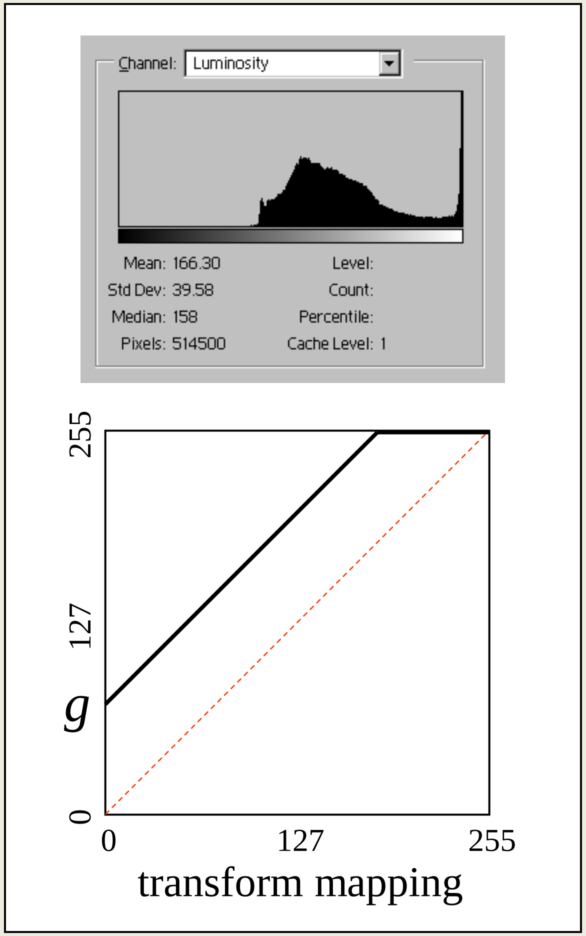
```
a = 2;
x = 0:255;
LUT = 255 ./ (1+exp(-a*(x-127)/32));
```

This is just one example.

Point Processes: Increase Brightness

$$J_k(r,c) = \begin{cases} I_k(r,c) + g, & \text{if } I_k(r,c) + g < 256 \\ 255, & \text{if } I_k(r,c) + g > 255 \end{cases}$$

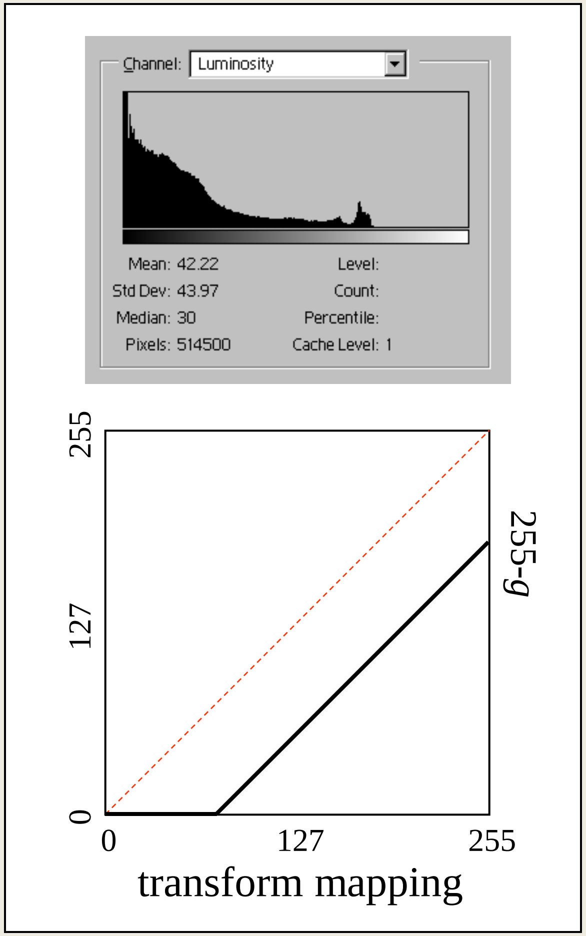
$g \geq 0$ and $k \in \{1,2,3\}$ is the bandindex.



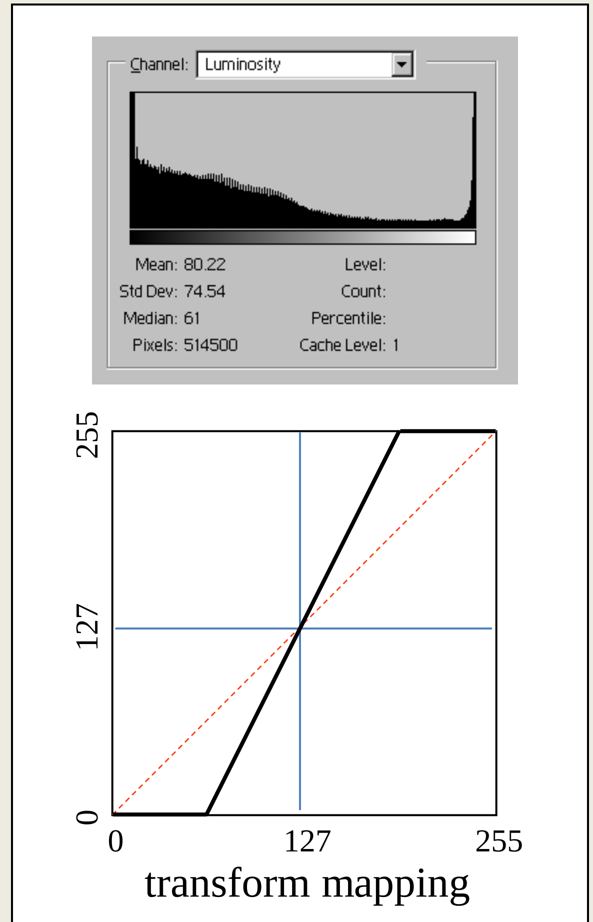
Point Processes: Decrease Brightness

$$J_k(r,c) = \begin{cases} 0, & \text{if } I_k(r,c) - g < 0 \\ I_k(r,c) - g, & \text{if } I_k(r,c) \end{cases}$$

$g \geq 0$ and $k \in \{1,2,3\}$ is the bandindex.



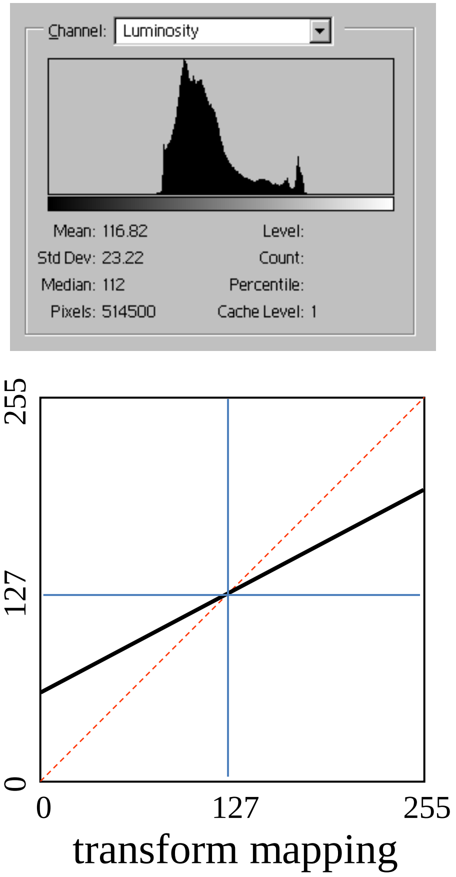
Point Processes: Increase Contrast



Let $T_k(r,c) = a\left[I_k(r,c) - 127\right] + 127,$ where $a > 1.0$

$$J_k(r,c) = \begin{cases} 0, & \text{if } T_k(r,c) < 0, \\ T_k(r,c), & \text{if } 0 \le T_k(r,c) \le 255, \\ 255, & \text{if } T_k(r,c) > 255. \quad k \in \{1,2,3\} \end{cases}$$

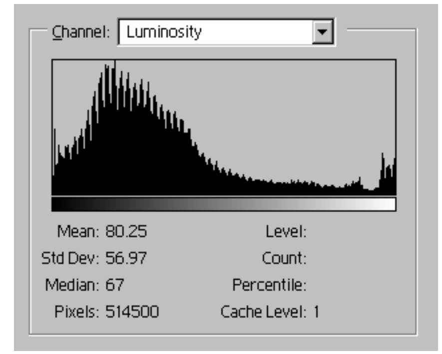
Point Processes: Decrease Contrast

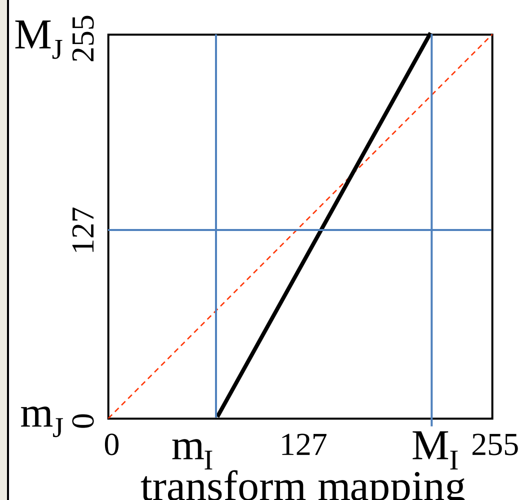


$$T_k(r,c) = a\left[I_k(r,c) - 127\right] + 127,$$

where $0 \le a < 1.0$ and $k \in \{1,2,3\}$.

Point Processes: Contrast Stretch





Let $m_I = \min[I(r,c)], \quad M_I = \max[I(r,c)],$

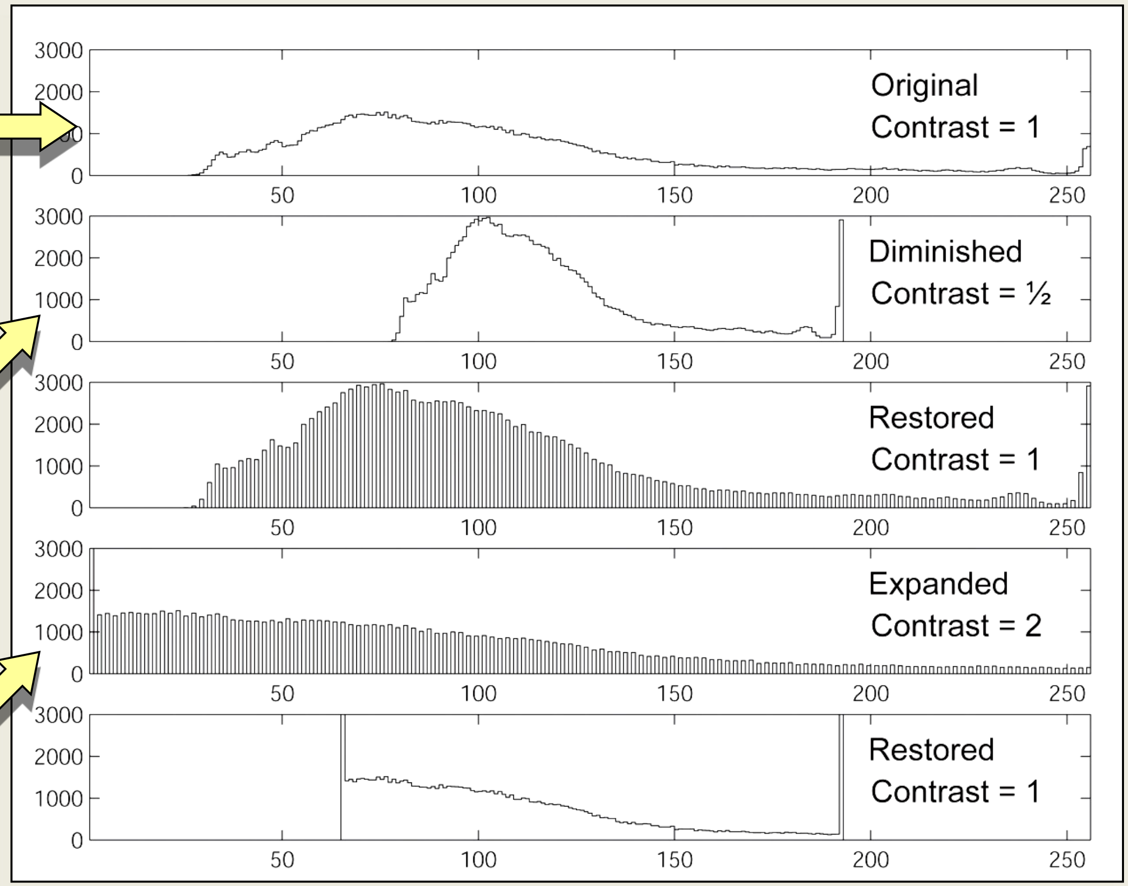
$m_J = \min[J(r,c)], \quad M_J = \max[J(r,c)].$

Then,

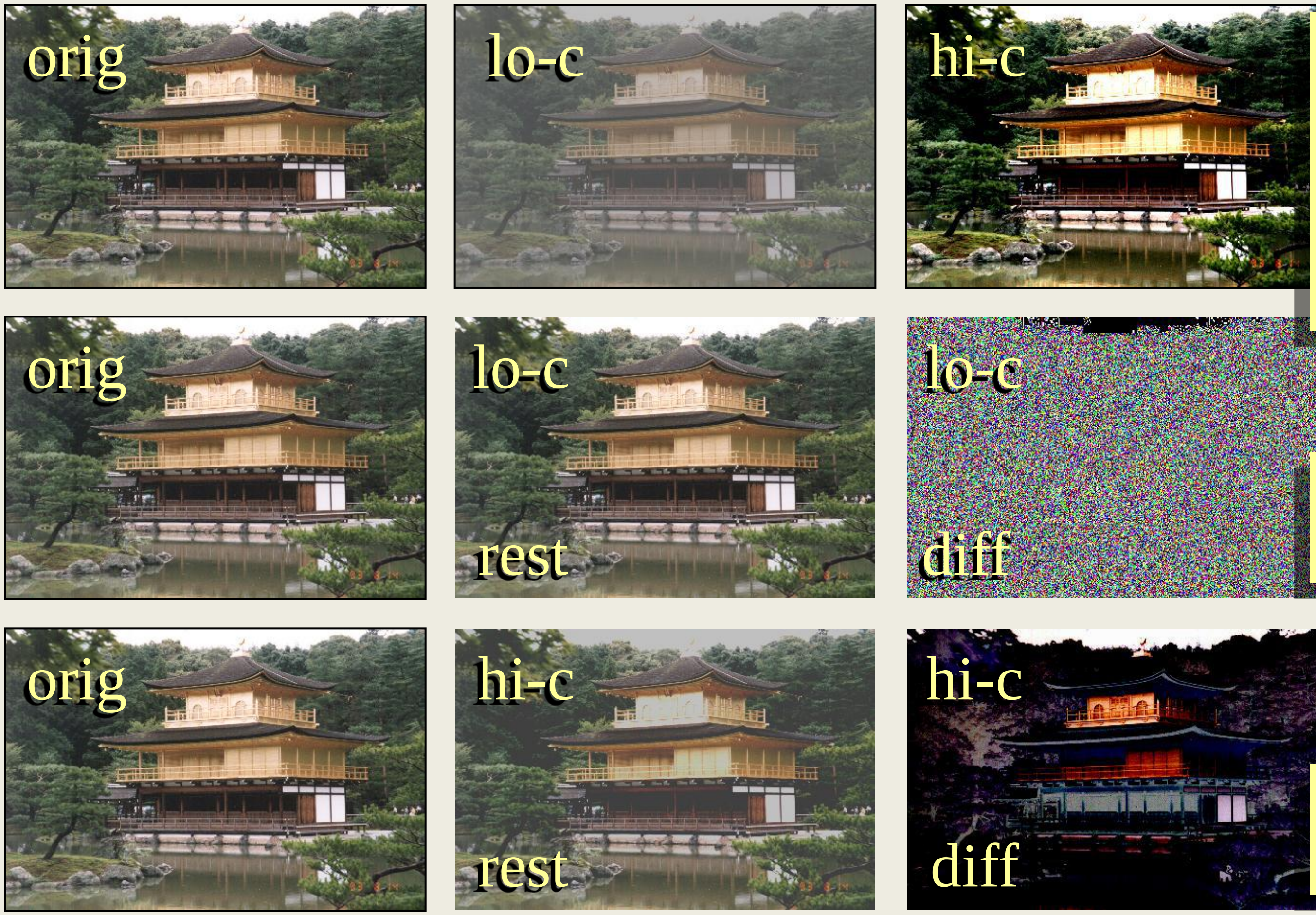
$$J(r,c) = (M_J - m_J)\frac{I(r,c) - m_I}{M_I - m_I} + m_J.$$

histograms

Information Loss from Contrast Adjustment



Information Loss from Contrast Adjustment

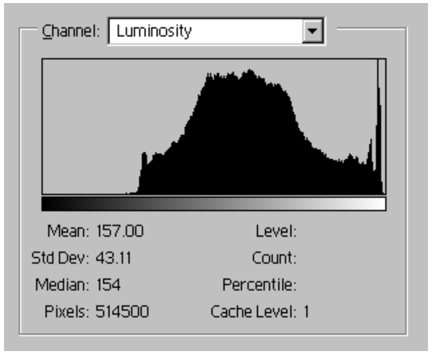


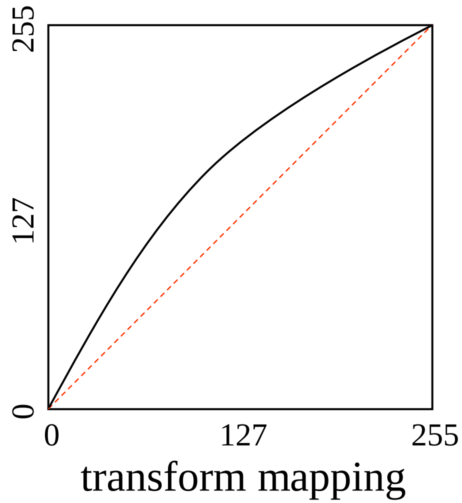
abbreviations:
original
low-contrast
high-contrast
restored
difference

difference between original and restored low-contrast

difference between original and restored high-contrast

Point Processes: Increased Gamma

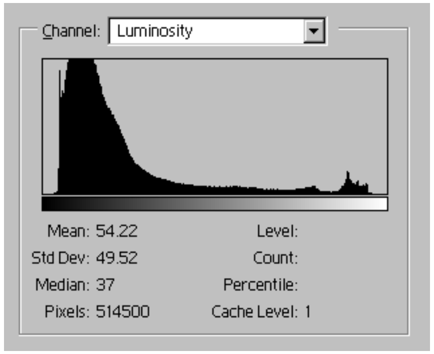


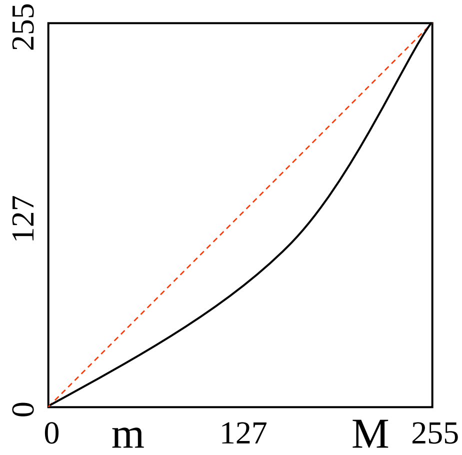


$$J(r,c) = 255 \cdot \left[\frac{I(r,c)}{255} \right]^{1/\gamma} \quad \text{for} \quad \gamma > 1.0$$

Point Processes: Decreased Gamma

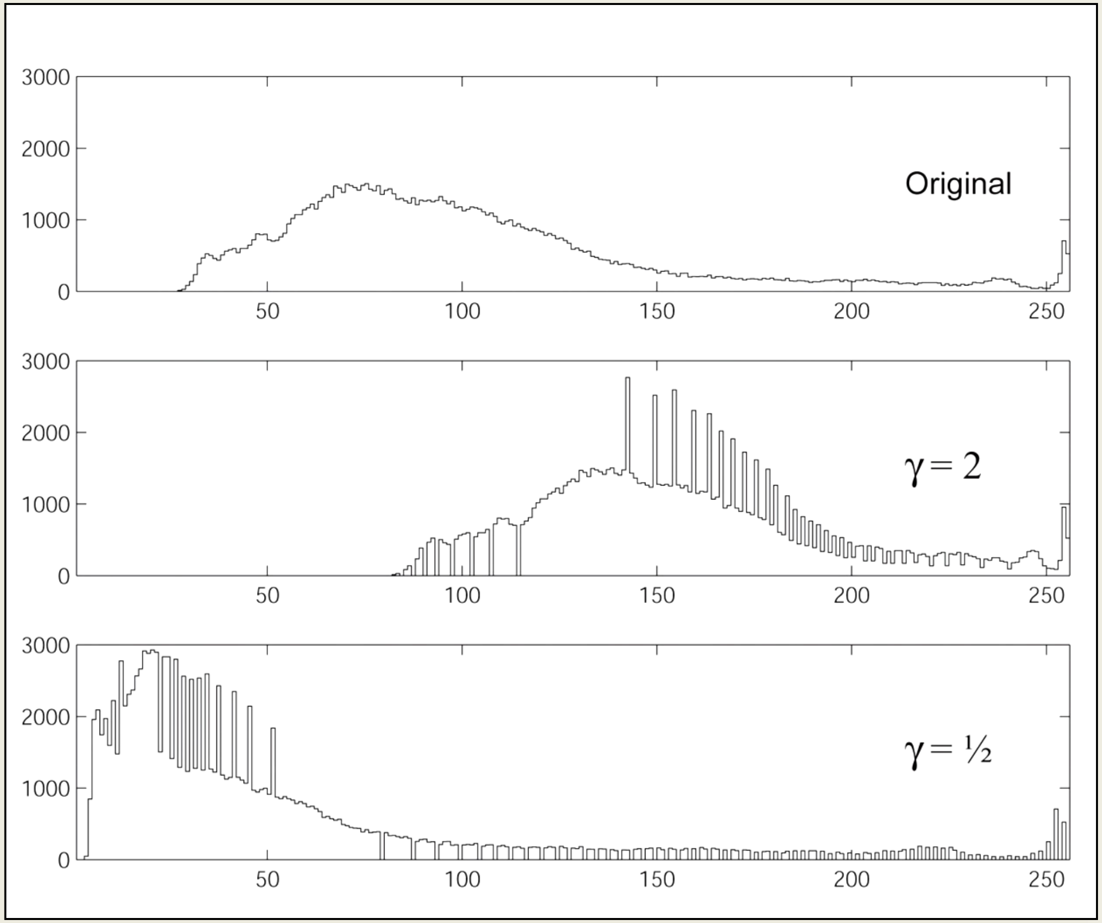
$$J(r,c) = 255 \cdot \left[\frac{I(r,c)}{255} \right]^{1/\gamma} \quad \text{for} \quad \gamma < 1.0$$





transform mapping

Gamma Correction: Effect on Histogram



The Probability Density Function of an Image

Let $A = \sum_{g=0}^{255} h_{I_k}(g+1)$.

pdf
[lower case]

Note that since $h_{I_k}(g+1)$ is the number of pixels in I_k (the kth color band of image I) with value g, A is the number of pixels in I. That is if I is R rows by C columns then $A = R \times C$.

Then,

$$p_{I_k}(g+1) = \frac{1}{A} h_{I_k}(g+1)$$

This is the probability that an arbitrary pixel from I_k has value g.

is the graylevel probability density function of I_k.

The Probability Density Function of an Image

- $p_{\text{band}}(g+1)$ is the fraction of pixels in (a specific band of) an image that have intensity value g.
- $p_{\text{band}}(g+1)$ is the probability that a pixel randomly selected from the given band has intensity value g.
- Whereas the sum of the histogram $h_{\text{band}}(g+1)$ over all g from 1 to 256 is equal to the number of pixels in the image, the sum of $p_{\text{band}}(g+1)$ over all g is 1.
- p_{band} is the normalized histogram of the band.

The Probability Distribution Function of an Image

Let $\mathbf{q} = [q_1\ q_2\ q_3] = I(r,c)$ be the value of a randomly selected pixel from I. Let g be a specific graylevel. The probability that $q_k \leq g$ is given by

$$P_{I_k}(g+1) = \sum_{\gamma=0}^{g} p_{I_k}(\gamma+1) = \frac{1}{A}\sum_{\gamma=0}^{g} h_{I_k}(\gamma+1) = \frac{\sum_{\gamma=0}^{g} h_{I_k}(\gamma+1)}{\sum_{\gamma=0}^{255} h_{I_k}(\gamma+1)},$$

where $h_{Ik}(\gamma+1)$ is the histogram of the kth band of I.

PDF
[upper case]

This is the probability that any given pixel from I_k has value less than or equal to g.

The Probability Distribution Function of an Image

Let $\mathbf{q} = [q_1\ q_2\ q_3] = I(r,c)$ be the value of a randomly selected pixel from I. Let g be a specific graylevel. The probability that $q_k \le g$ is given by

Also called CDF for "Cumulative Distribution Function".

$$P_{I_k}(g+1) = \sum_{\gamma=0}^{g} p_{I_k}(\gamma+1) = \frac{1}{A}\sum_{\gamma=0}^{g} h_{I_k}(\gamma+1) = \frac{\sum_{\gamma=0}^{g} h_{I_k}(\gamma+1)}{\sum_{\gamma=0}^{255} h_{I_k}(\gamma+1)},$$

where $h_{Ik}(\gamma+1)$ is the histogram of the kth band of I.

This is the probability that any given pixel from I_k has value less than or equal to g.

A.k.a. Cumulative Distribution Function.

The Probability Distribution Function of an Image

- $P_{\text{band}}(g+1)$ is the fraction of pixels in (a specific band of) an image that have intensity values less than or equal to g.
- $P_{\text{band}}(g+1)$ is the probability that a pixel randomly selected from the given band has an intensity value less than or equal to g.
- $P_{\text{band}}(g+1)$ is the cumulative (or running) sum of $p_{\text{band}}(g+1)$ from 0 through g inclusive.
- $P_{\text{band}}(1) = \mathrm{p}_{\text{band}}(1)$ and $P_{\text{band}}(256) = 1$; $P_{\text{band}}(g+1)$ is nondecreasing.

Note: the Probability Distribution Function (PDF, capital letters) and the Cumulative Distribution Function (CDF) are exactly the same things. Both PDF and CDF will refer to it. However, pdf (small letters) is the *density* function.

Point Processes: Histogram Equalization

Task: remap image I so that its histogram is as close to constant as possible

Let $P_I(\gamma+1)$

be the cumulative (probability) distribution function of I.

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

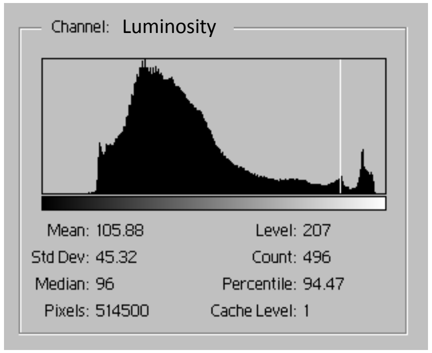
$$J(r,c)=255\cdot P_I\left[I(r,c)+1\right].$$

The CDF itself is used as the LUT.

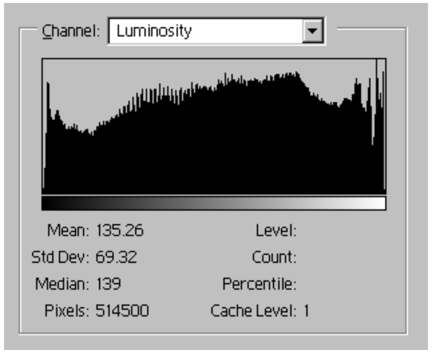
all bands processed similarly

Point Processes: Histogram Equalization

$$J(r,c) = 255 \cdot P_I(g+1)$$

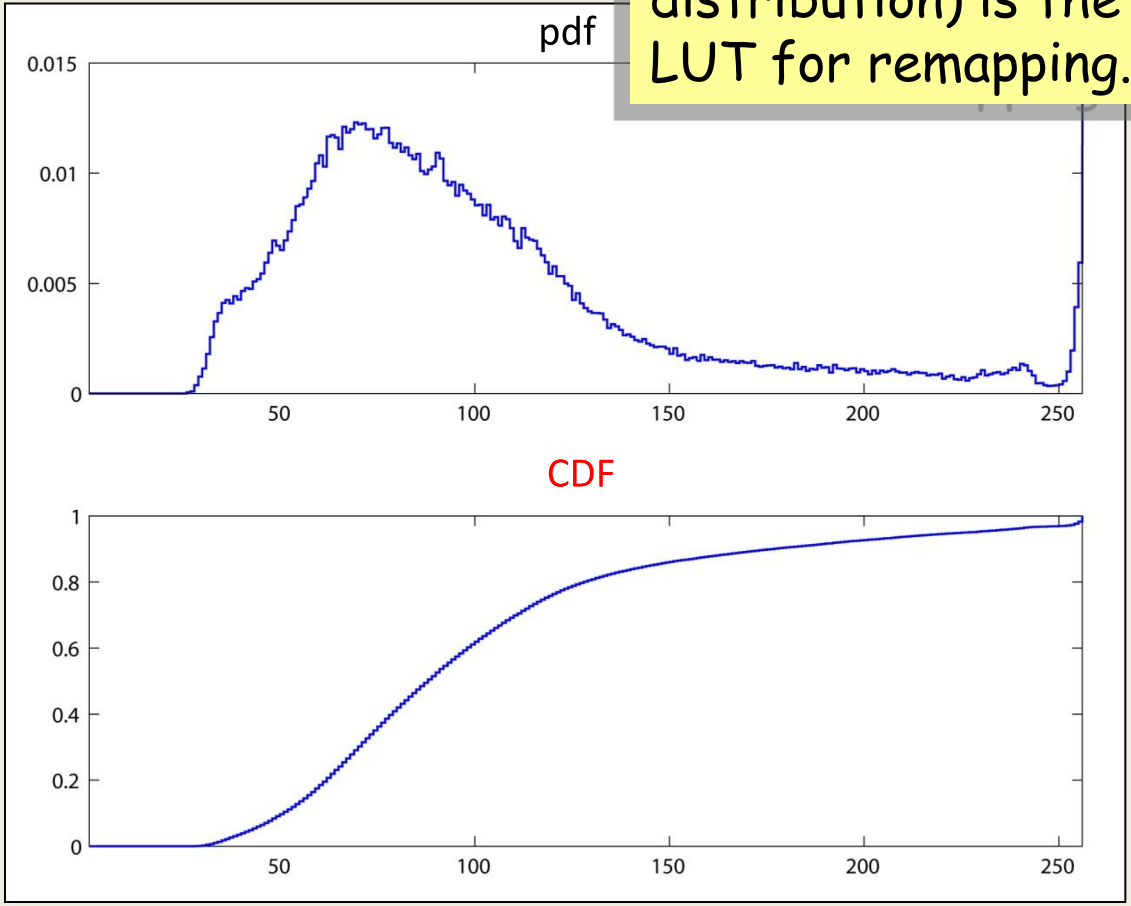


before



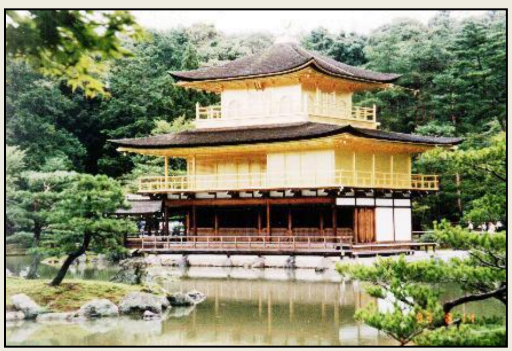
after

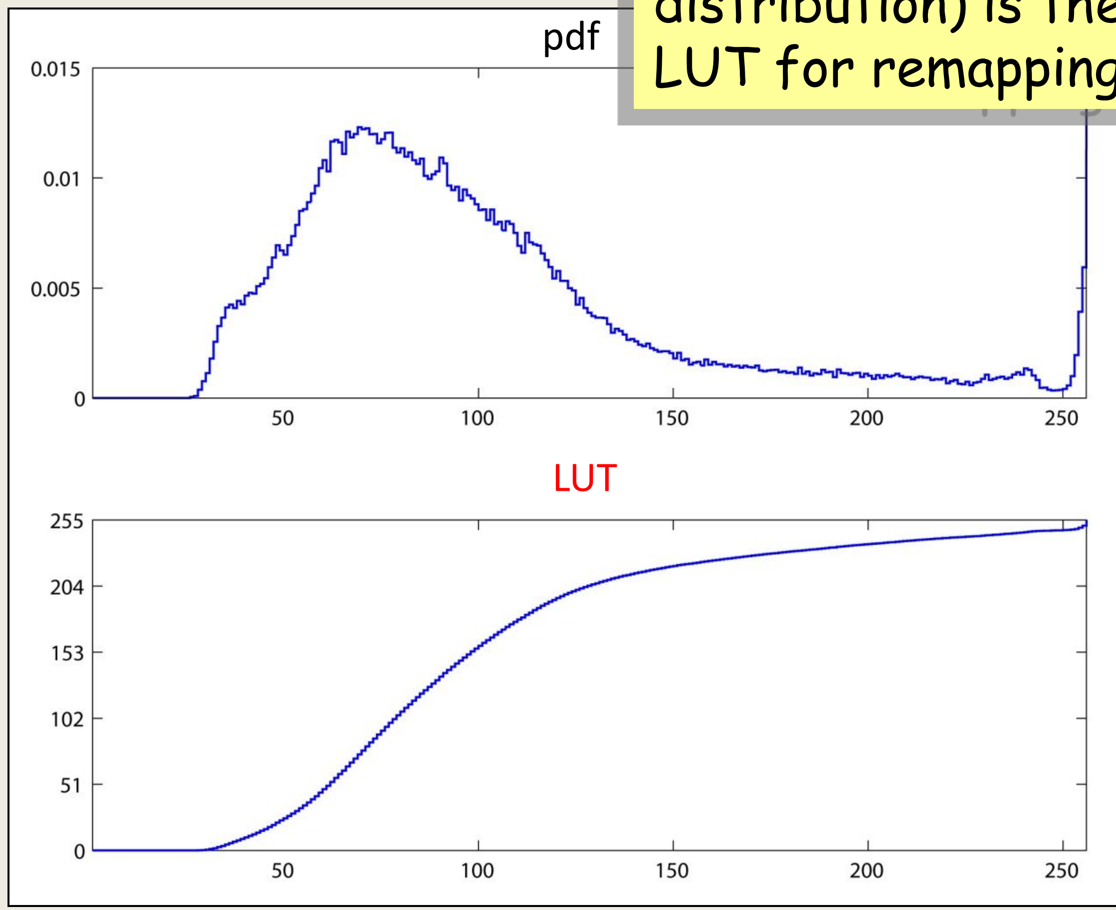
Histogram EQ



The CDF (cummulative distribution) is the LUT for remapping.

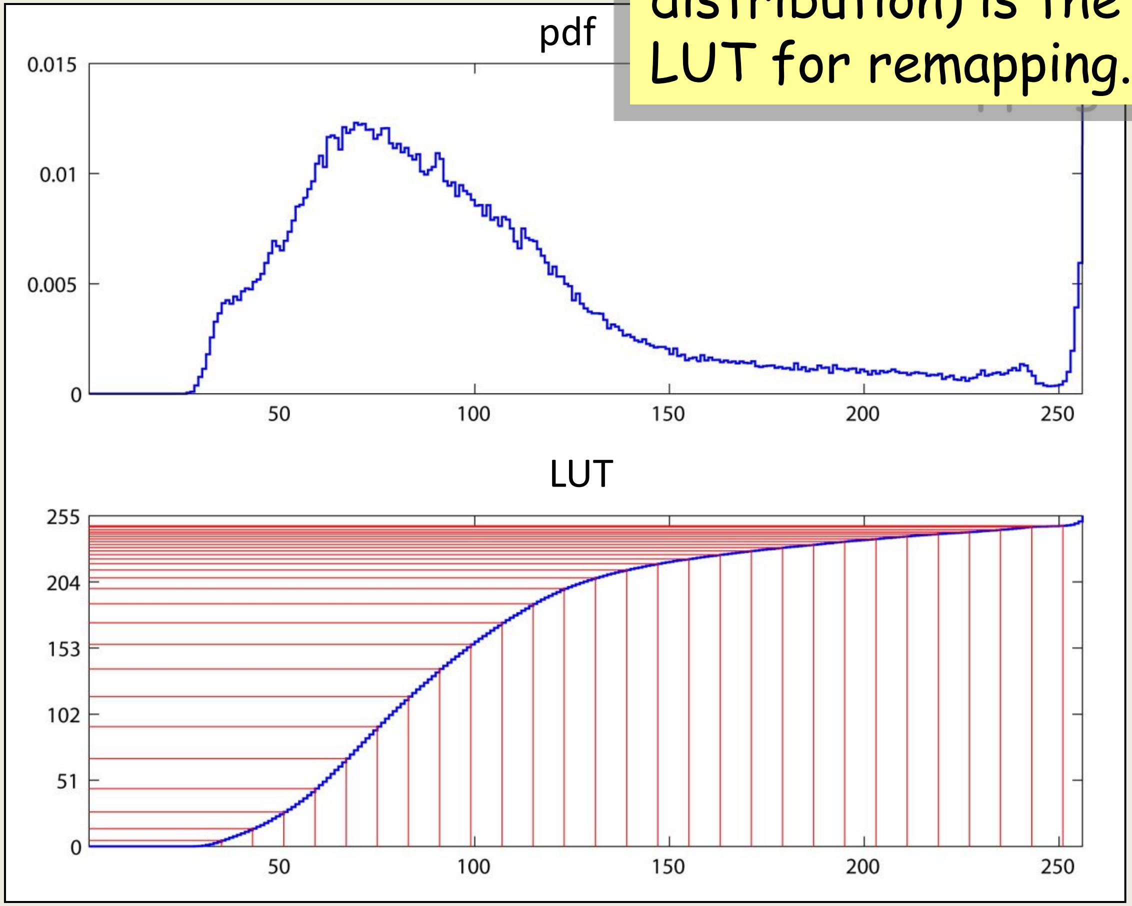
Histogram EQ





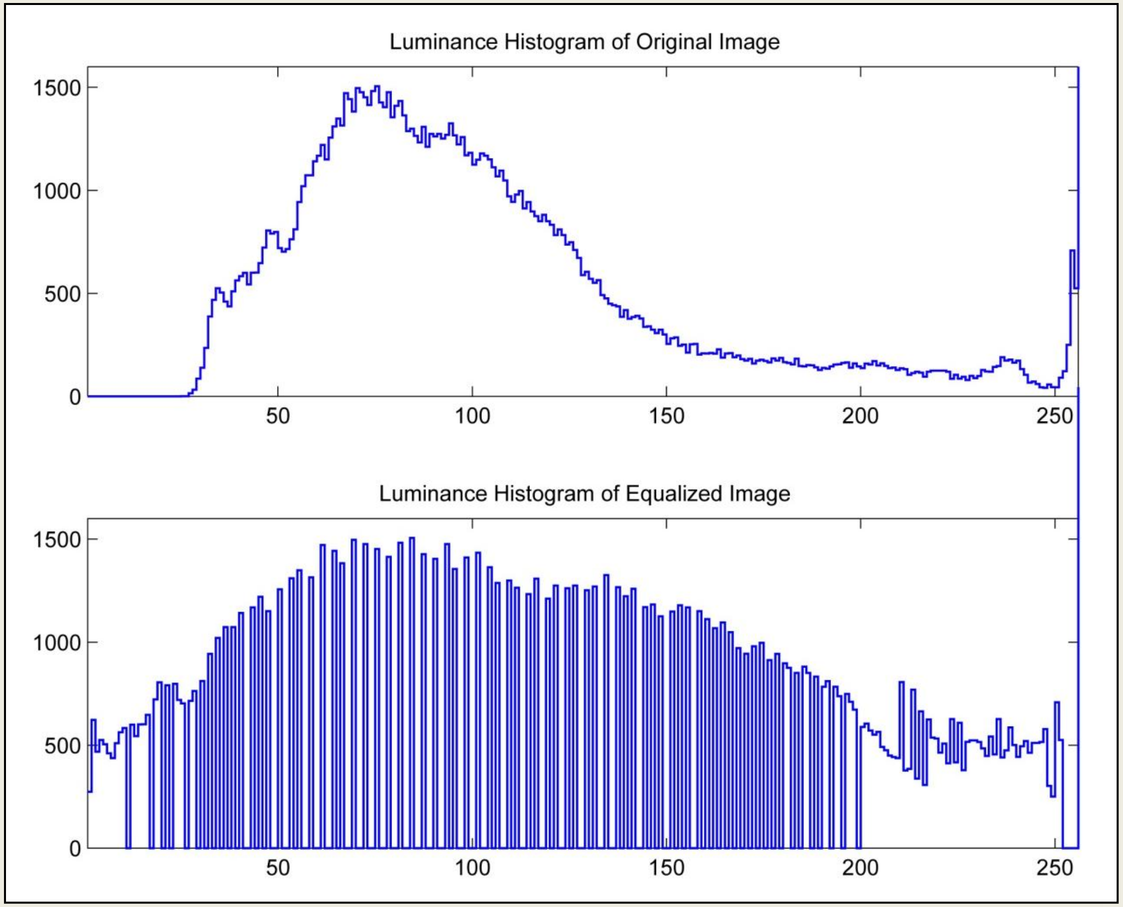
The CDF (cummulative distribution) is the LUT for remapping.

Histogram EQ



The CDF (cummulative distribution) is the LUT for remapping.

Histogram EQ



Point Processes: Histogram Equalization

Task: remap image I with min = m_I and max = M_I so that its histogram is as close to constant as possible and has min = m_J and max = M_J.

Let $P_I(\gamma+1)$
be the cumulative (probability) distribution function of I.

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

Using intensity extrema

$$J(r,c)=(M_J-m_J)\frac{P_I[I(r,c)+1]-P_I(m_I+1)}{1-P_I(m_I+1)}+m_J.$$

Point Processes: Histogram Matching

Task: remap image I so that it has, as closely as possible, the same histogram as image J.

Because the images are digital it is not, in general, possible to make $h_I \equiv h_J$. Therefore, $p_I \not\equiv p_J$.

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

Matching Percentiles

… assuming a 1-band image or a single band of a color image.

Recall:

- The CDF of image I is such that $0 \leq P_I(g_I) \leq 1$.
- $P_I(g_I + 1) = c$ means that c is the fraction of pixels in I that have a value less than or equal to g_I.
- $100c$ is the *percentile* of pixels in I that are less than or equal to g_I.

To match percentiles, replace all occurrences of value g_I in image I with the value, g_J, from image J whose percentile in J most closely matches the percentile of g_I in image I.

Matching Percentiles

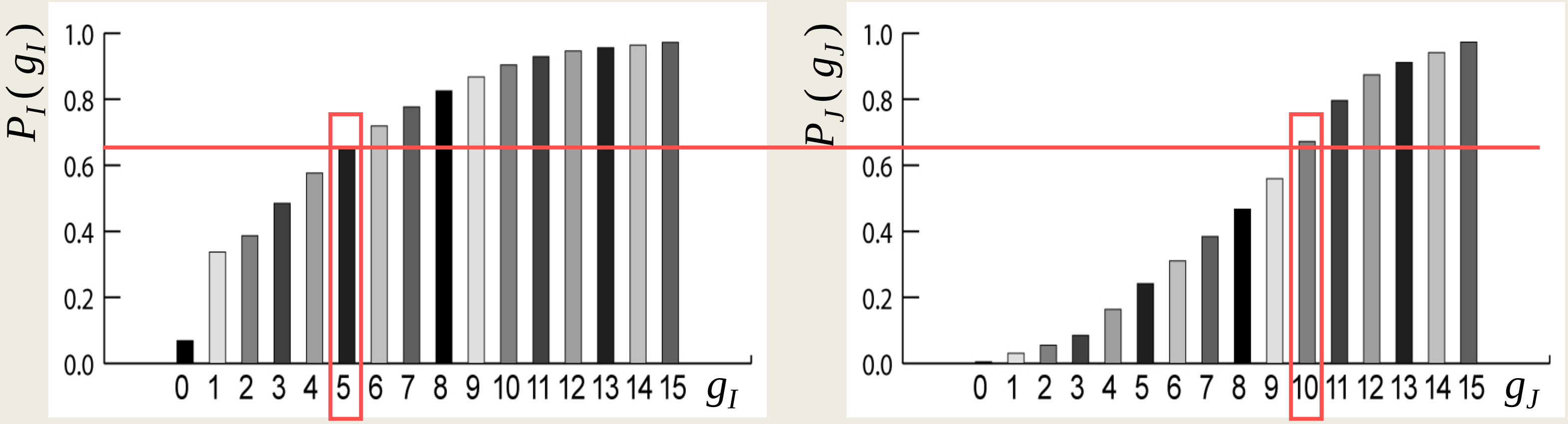
… assuming a 1-band image or a single band of a color image.

So, to create an image, K, from image I such that K has nearly the same CDF as image J do the following:

If $I(r,c) = g_I$ then let $K(r,c) = g_J$ where g_J is such that

$$P_I(g_I) > P_J(g_J - 1) \text{ AND } P_I(g_I) \leq P_J(g_J).$$

Example:
$I(r,c) = 5$
$P_I(5) = 0.65$
$P_J(9) = 0.56$
$P_J(10) = 0.67$
$K(r,c) = 10$



Histogram Matching Algorithm

… assuming a 1-band image or a single band of a color image.

```
[R,C] = size(I);
K = zeros(R,C);
g_J = m_J;
for g_I = m_I to M_I
    while g_J < 255 AND P_I(g_I + 1) < 1 AND
            P_J(g_J + 1) < P_I(g_I + 1)
        g_J = g_J + 1;
    end
    K = K + [g_J × (I == g_I)]
end
```

This directly matches image I to image J.

$P_I(g_I + 1)$: CDF of I

$P_J(g_J + 1)$: CDF of J.

$m_J = \min J$,

$M_J = \max J$,

$m_I = \min I$,

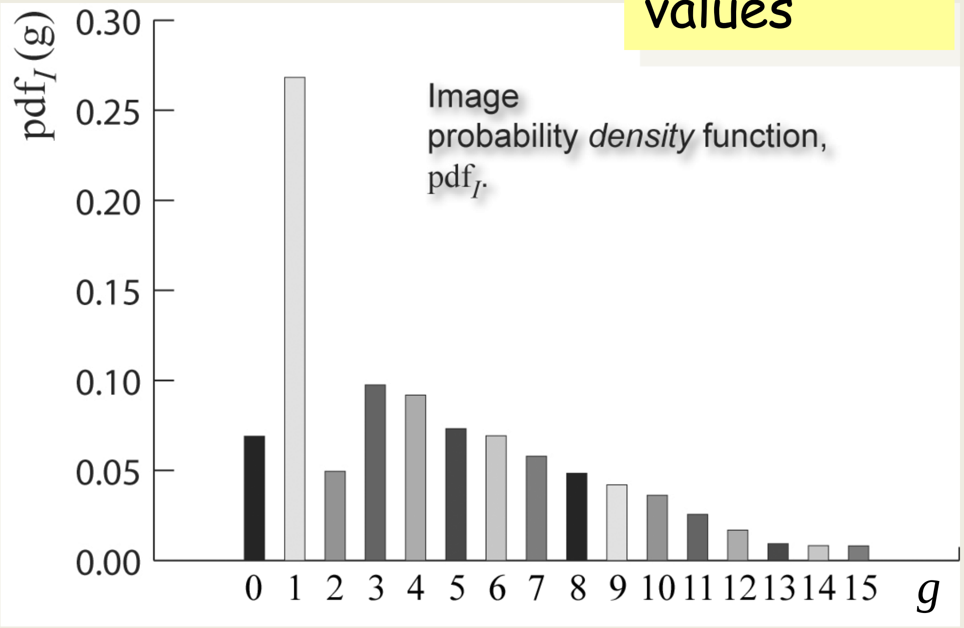
$M_I = \max I$.

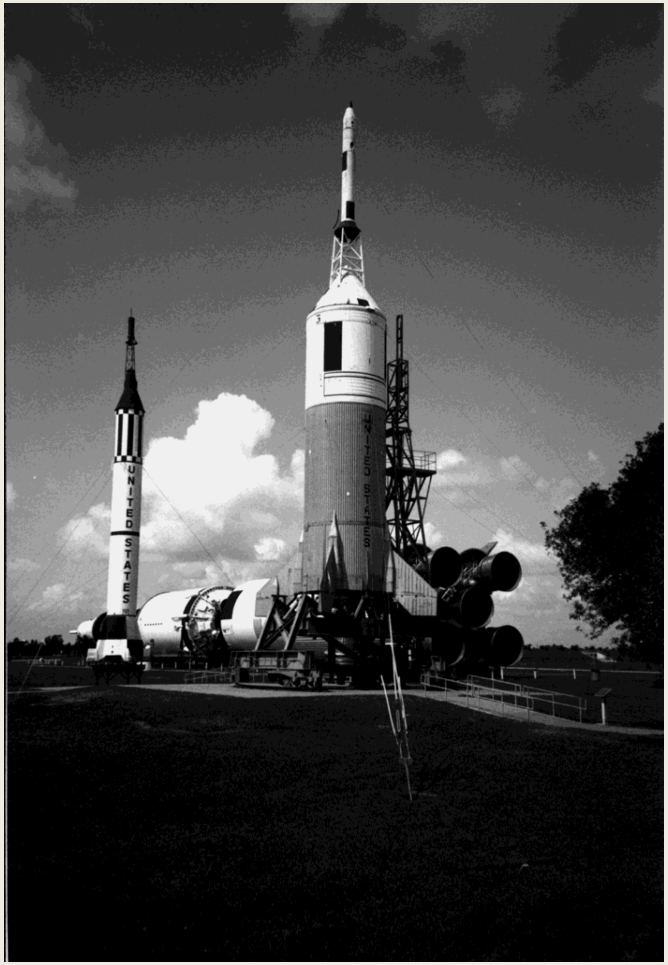
Better to use a LUT. See slide 54.

Example: Histogram Matching

Image pdf

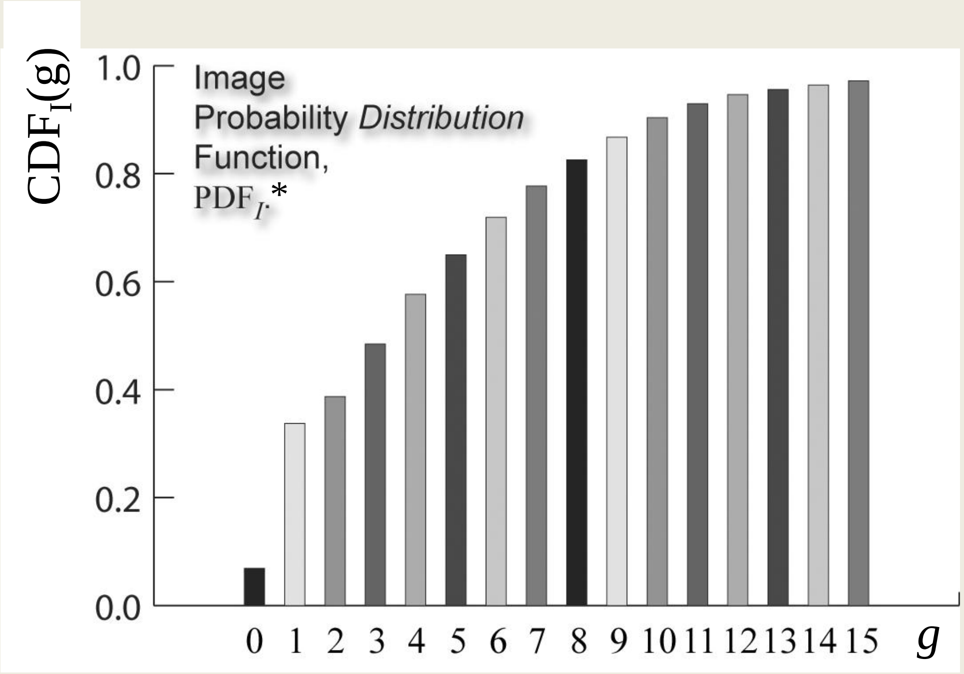
Image with 16 intensity values



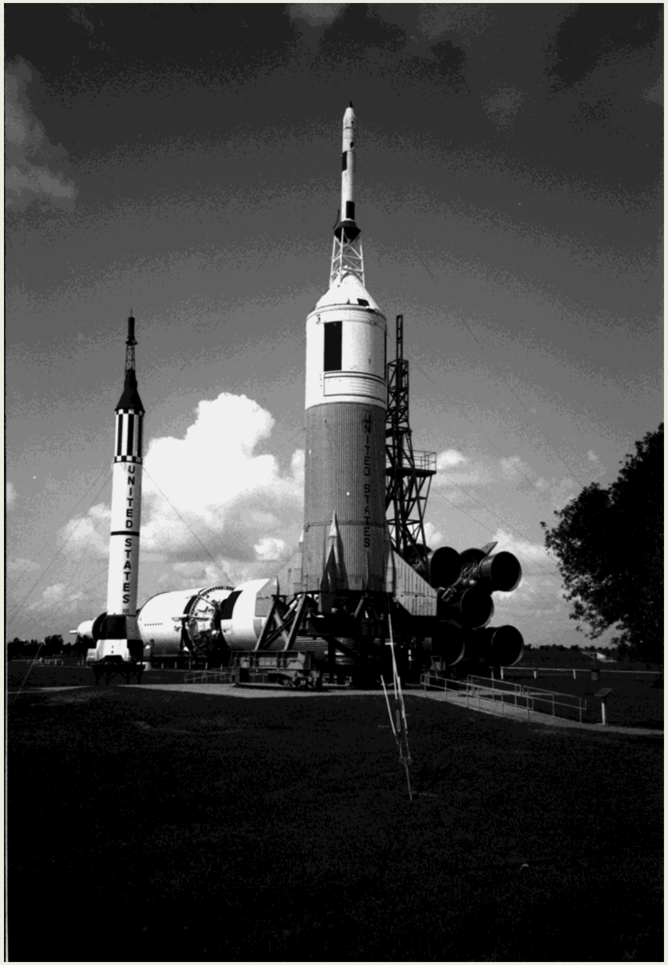


Example: Histogram Matching

Image CDF



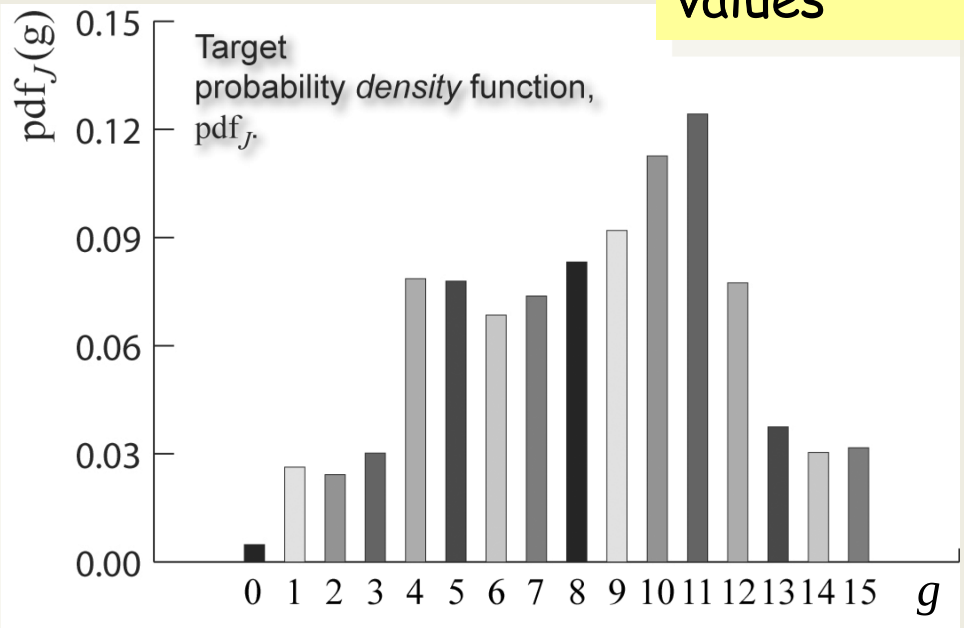
*a.k.a Cumulative Distribution Function, CDF_I.

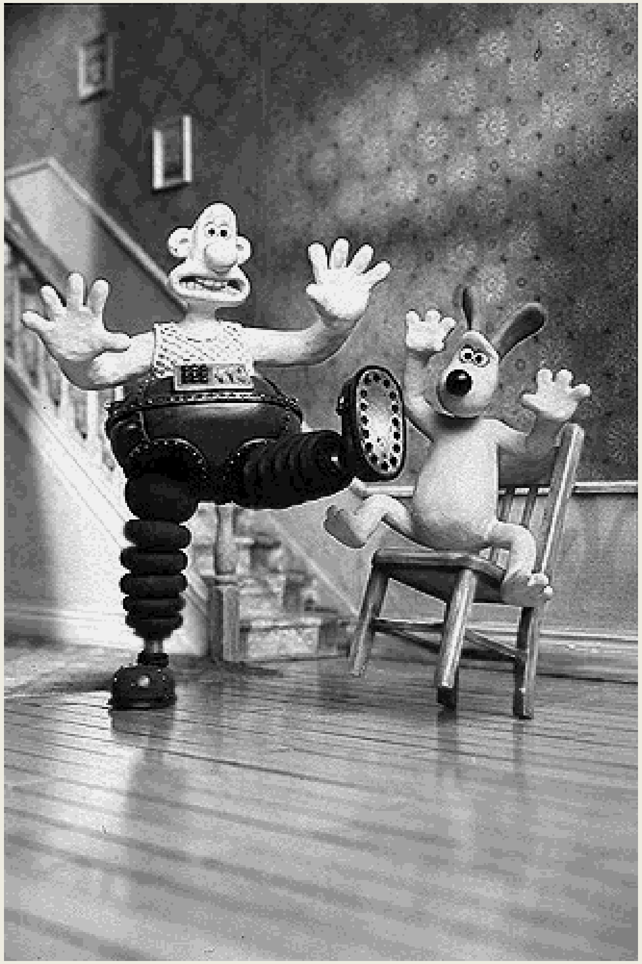


Example: Histogram Matching

Target pdf

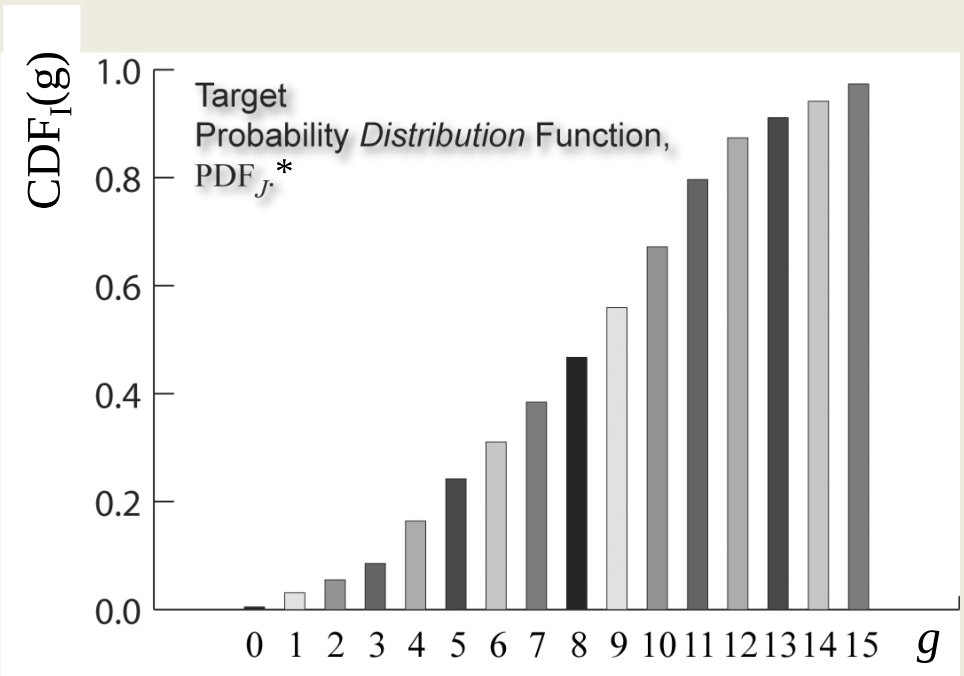
Target with 16 intensity values



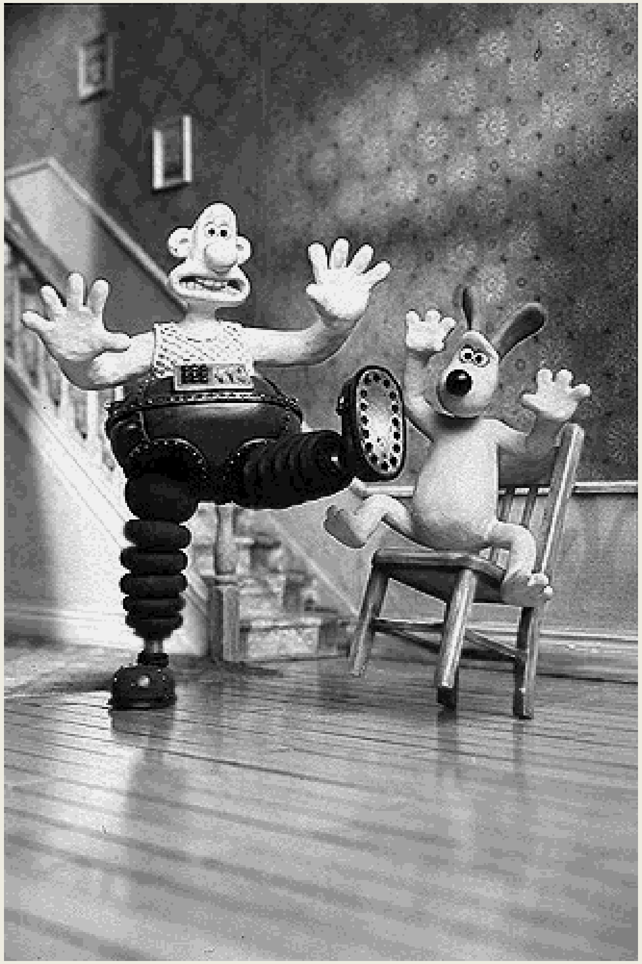


Example: Histogram Matching

Target CDF



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



Histogram Matching with a Lookup Table

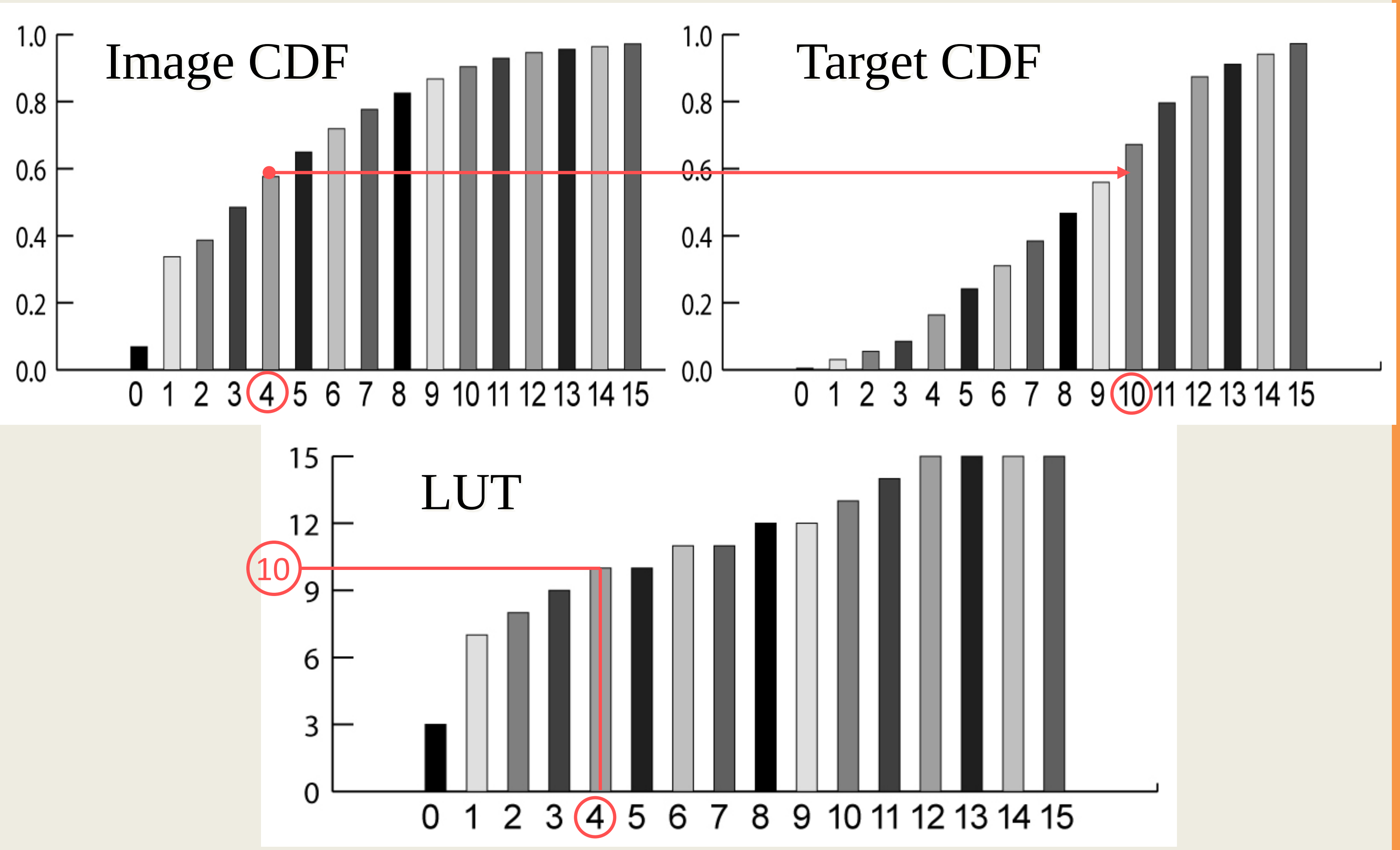
The algorithm on slide 50 matches one image to another directly. Often it is faster or more versatile to use a lookup table (LUT). Rather than remapping each pixel in the image separately, one can create a table that indicates to which target value each input value should be mapped. Then

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

In *Matlab* if the LUT is a 256 × 1 matrix with values from 0 to 255 and if image *I* is of type **`uint8`**, it can be remapped with the following code:

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

LUT Creation



Look Up Table for Histogram Matching

```
LUT = zeros(256,1);
g_J = 0;
for g_I = 0 to 255
    while P_J(g_J + 1) < P_I(g_I + 1) AND g_J < 255
        g_J = g_J + 1;
    end
    LUT(g_I + 1) = g_J;
end
```

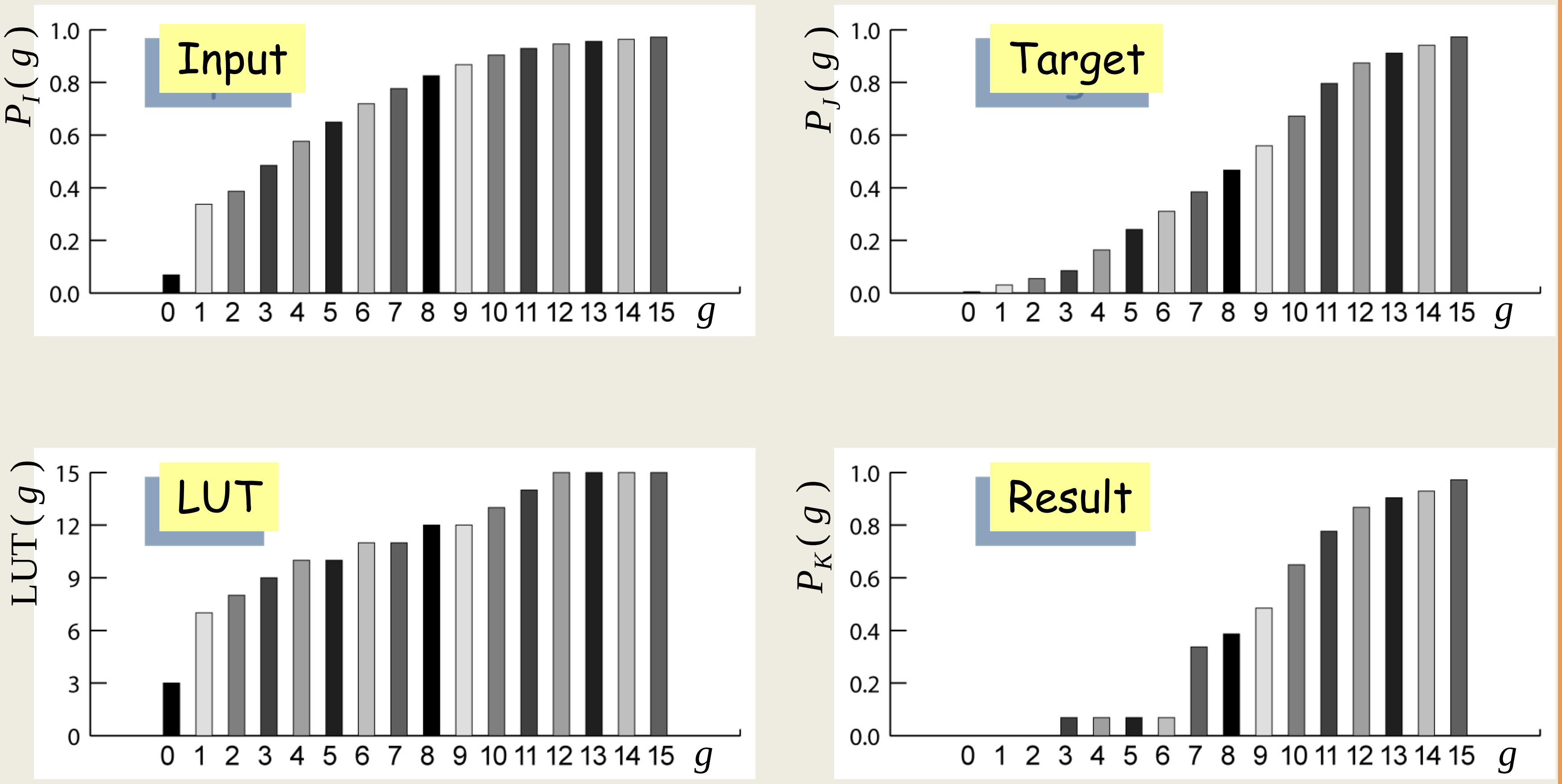
This creates a look-up table which can then be used to remap the image.

$P_I(g_I + 1)$: CDF of I ,

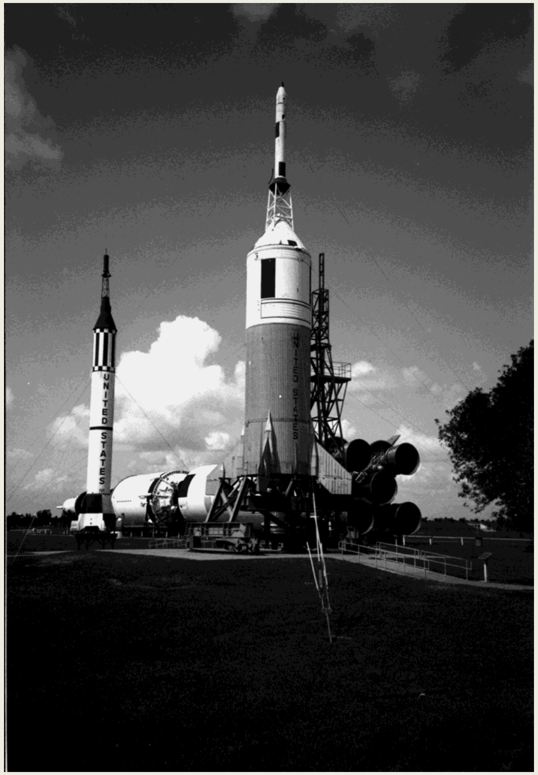
$P_J(g_J + 1)$: CDF of J,

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

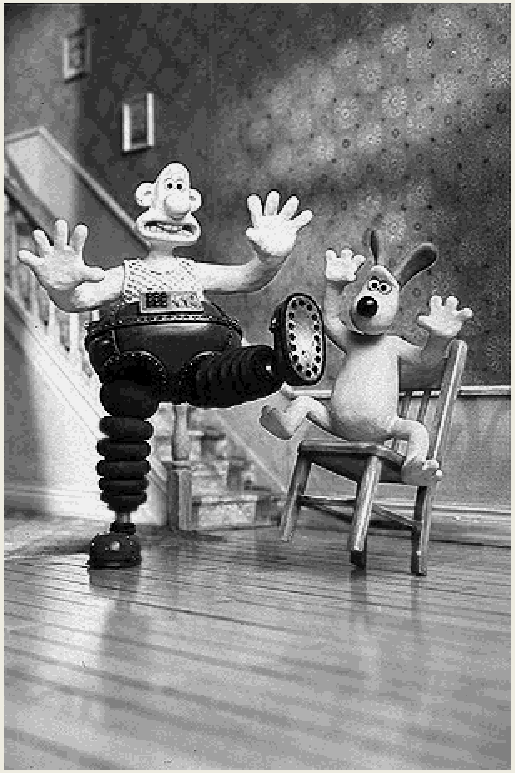
Input & Target CDFs, LUT and Resultant CDF



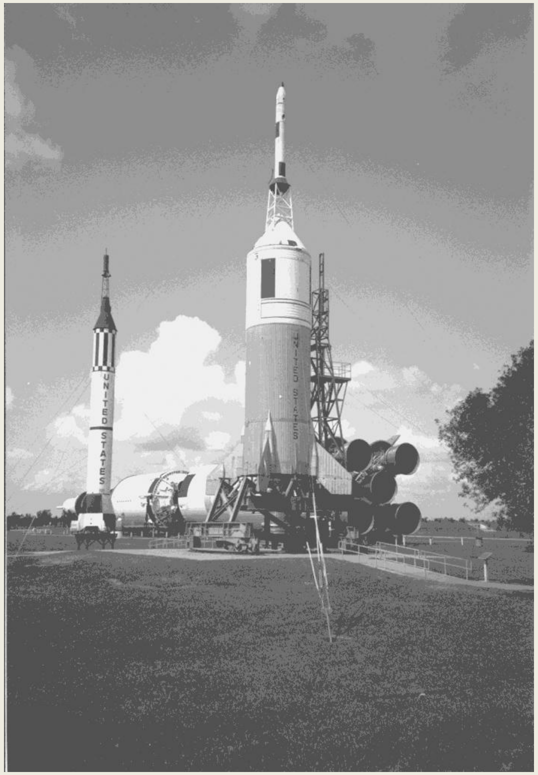
Example: Histogram Matching



original

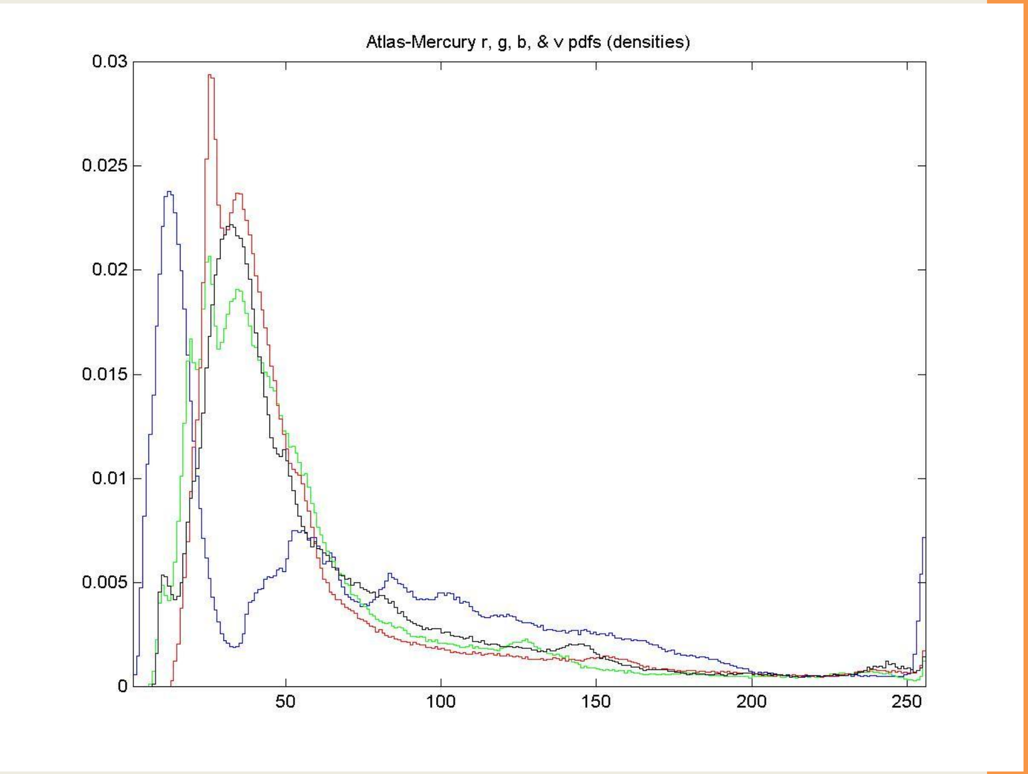


target



remapped

Probability Density Functions of a Color Image



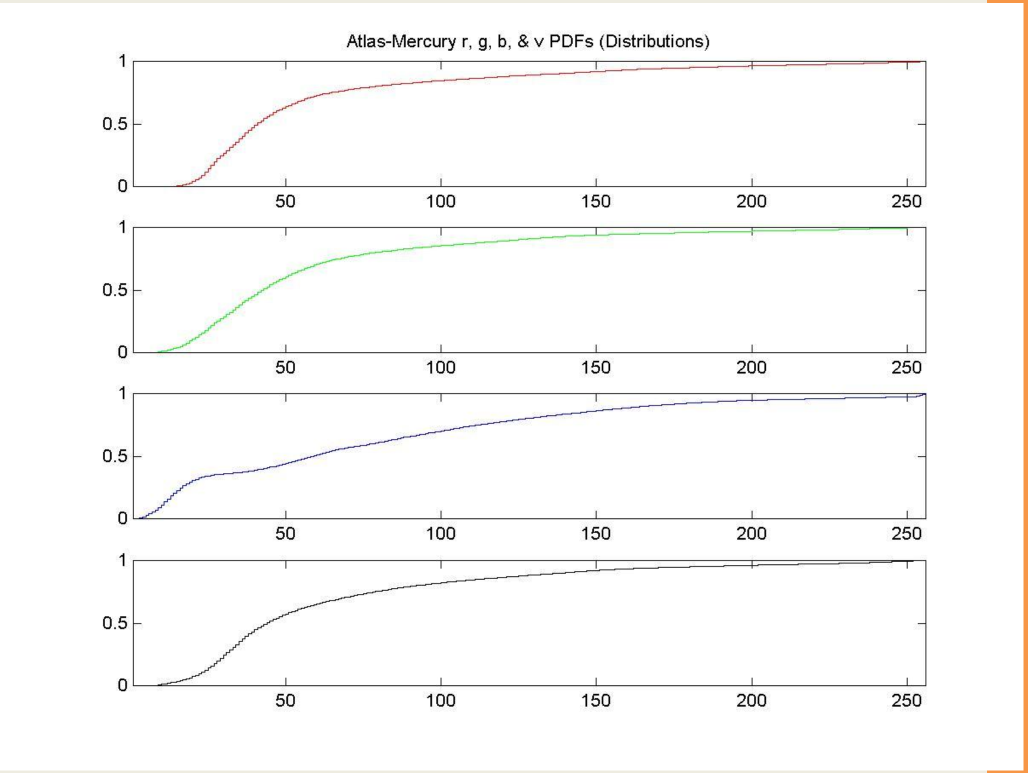
red pdf

green pdf

blue pdf

luminosity pdf

Cumulative Distribution Functions (CDF)



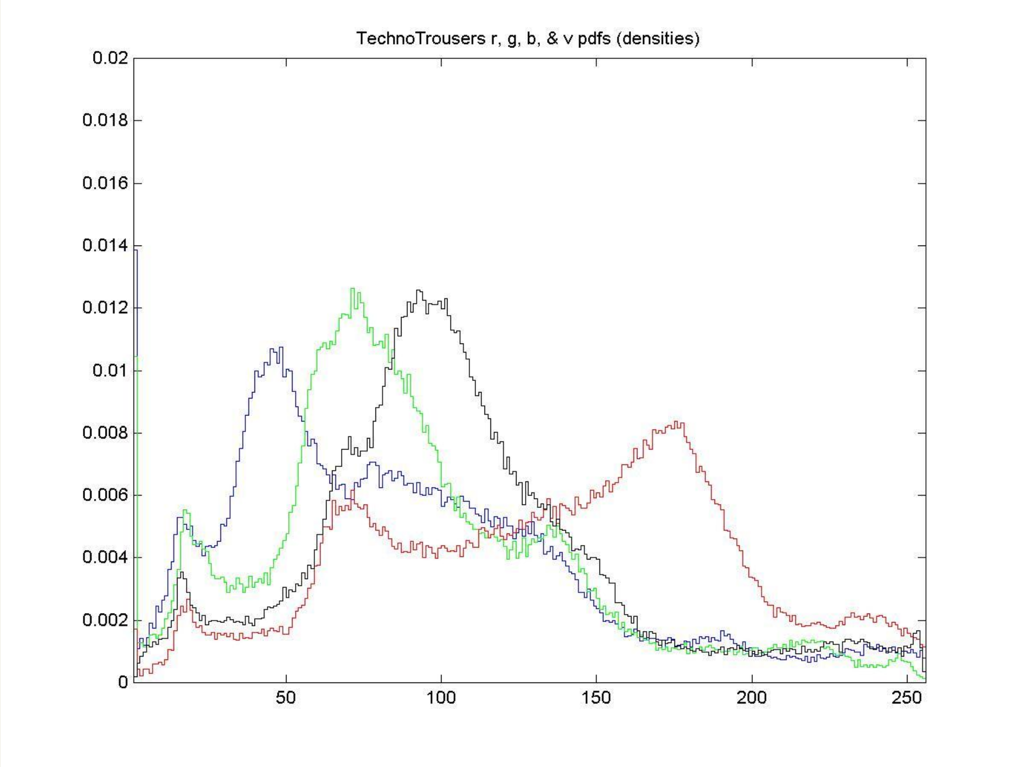
red CDF

green CDF

blue CDF

luminosity CDF

Probability Density Functions of a Color Image



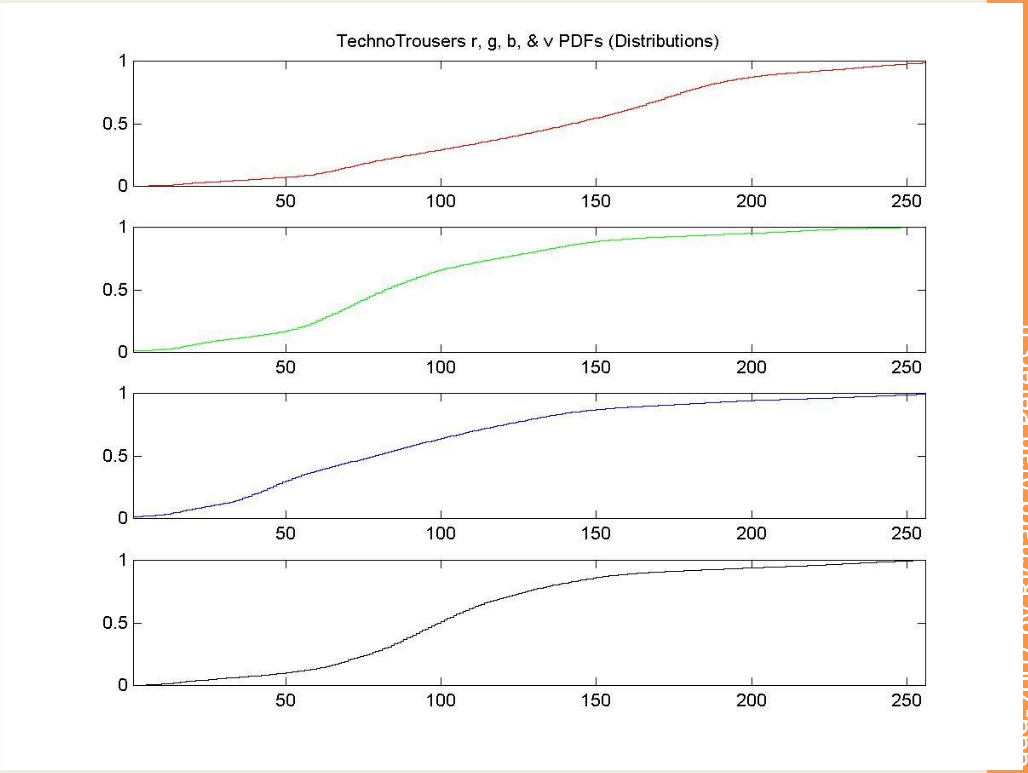
red pdf

green pdf

blue pdf

luminosity pdf

Cumulative Distribution Functions (CDF)



red CDF
green CDF
blue CDF
luminosity CDF

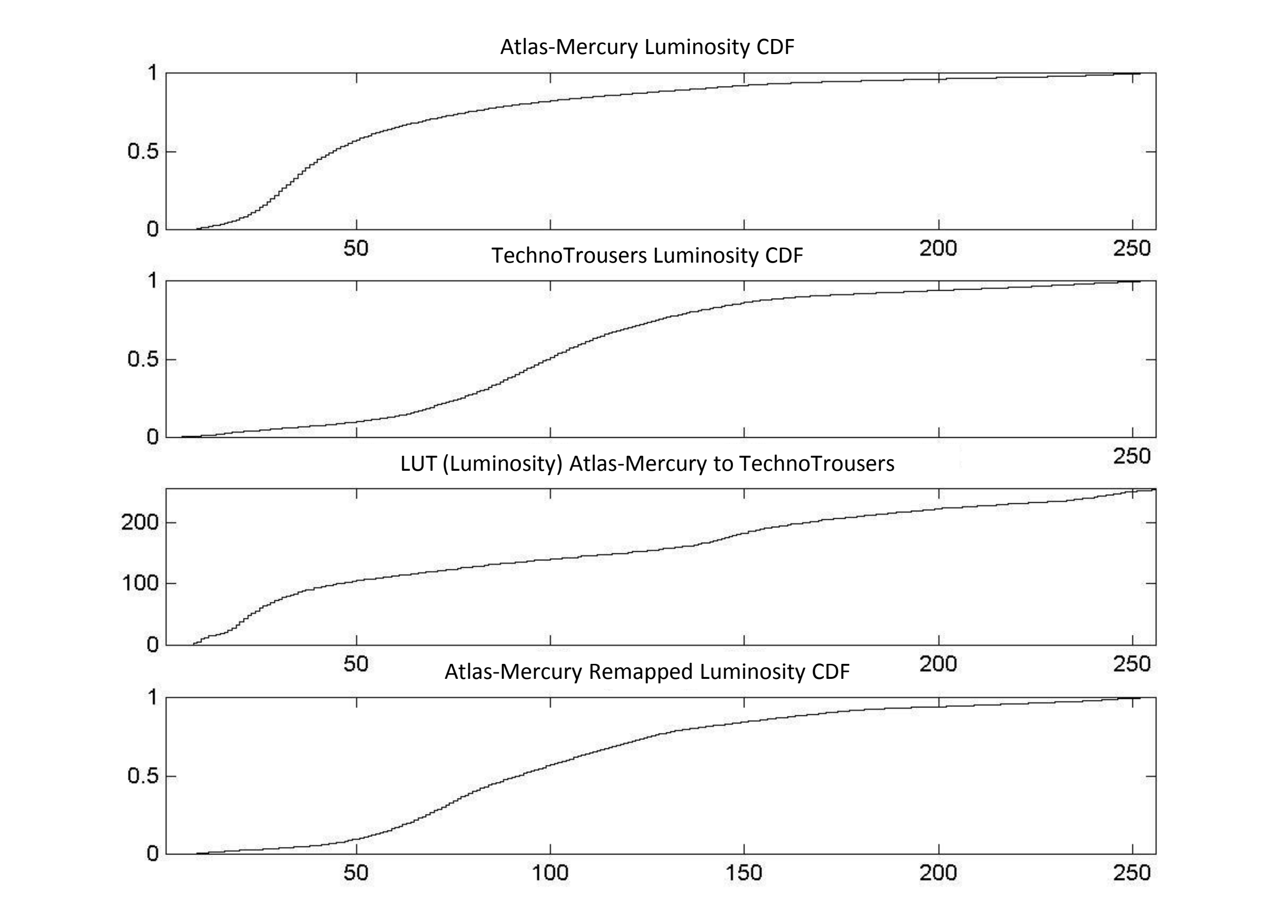
Remap an Image to have the Lum. CDF of Another

original

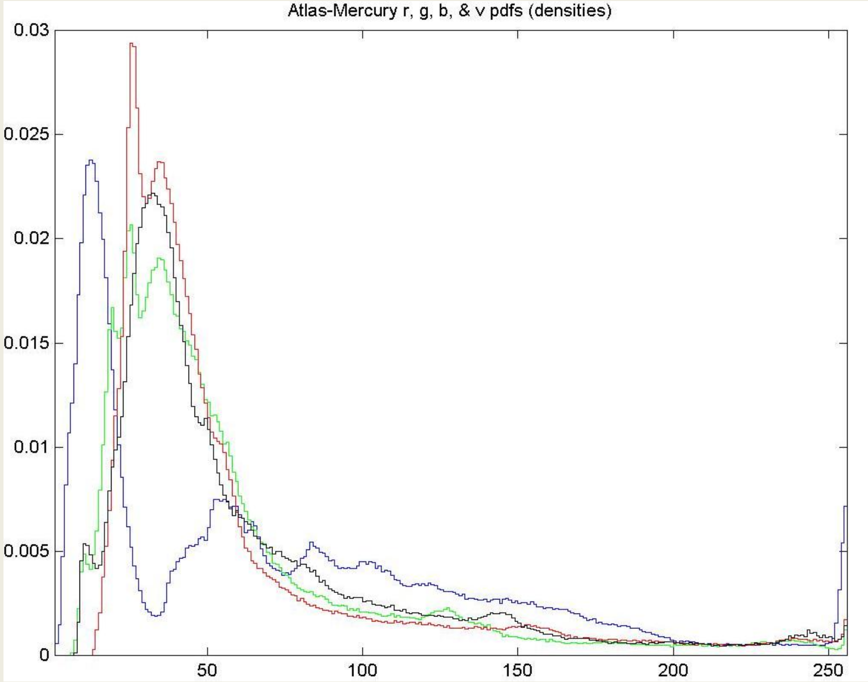
target

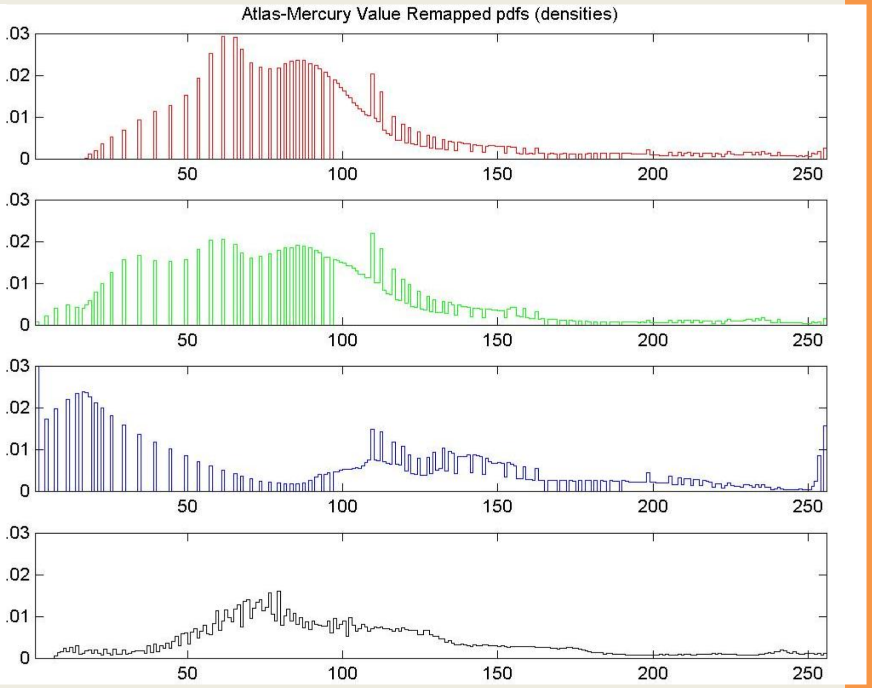
luminosity remapped

CDFs and the LUT



Effects of Luminance Remapping on pdfs

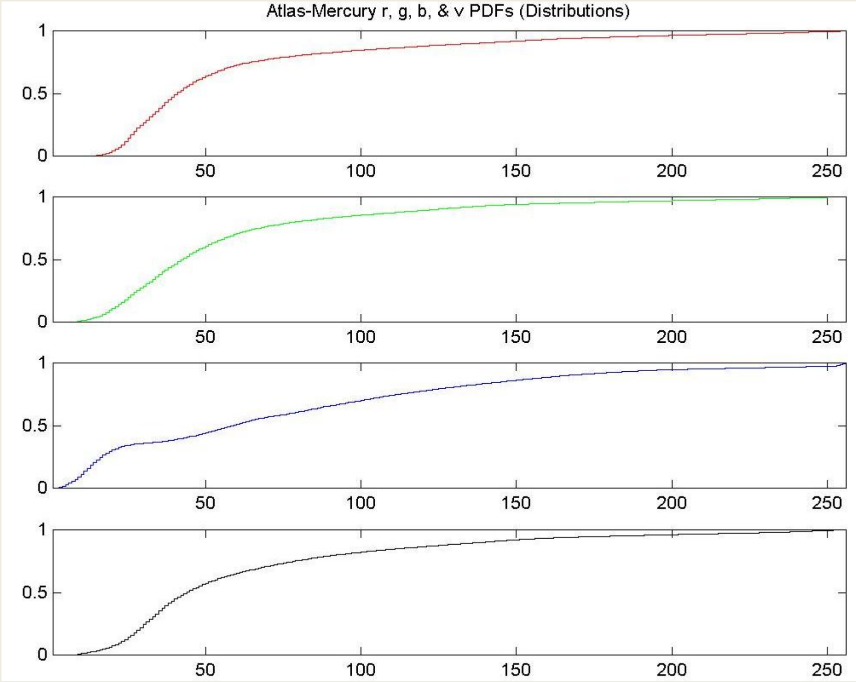




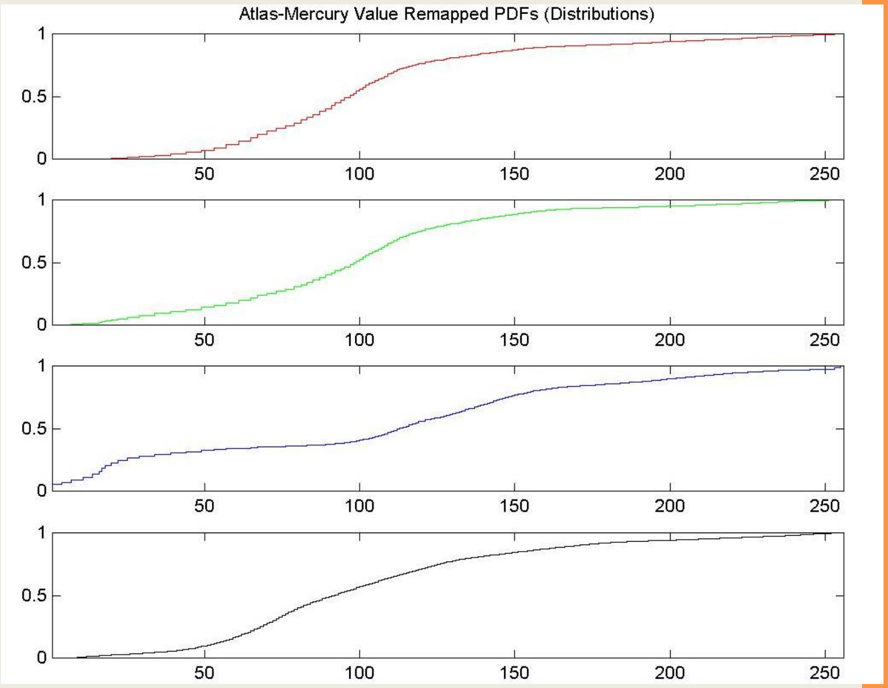
Before

After

Effects of Luminance Remapping on CDFs



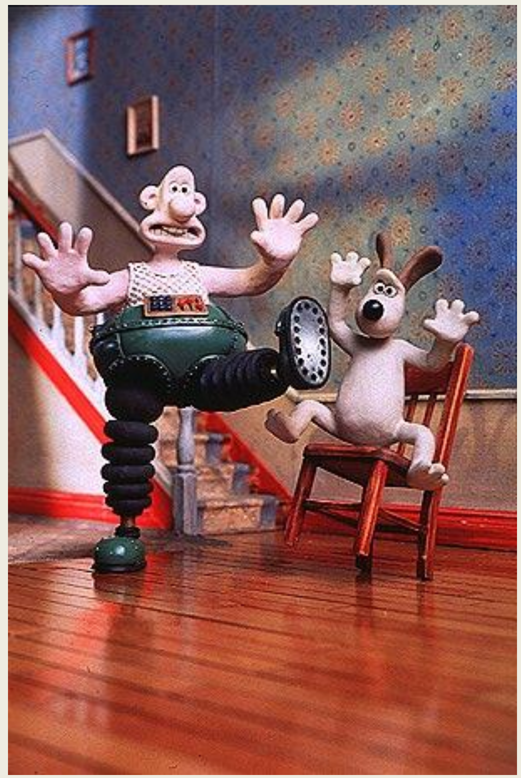
Before



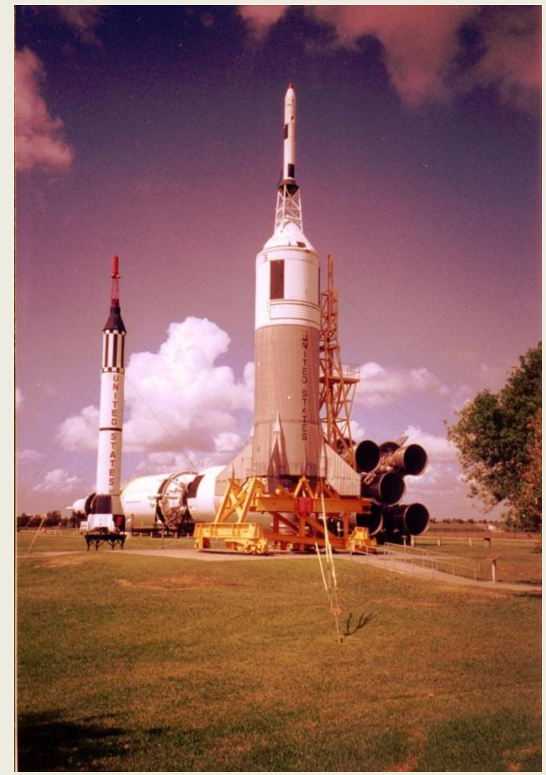
After

Remap an Image to have the rgb CDF of Another

original

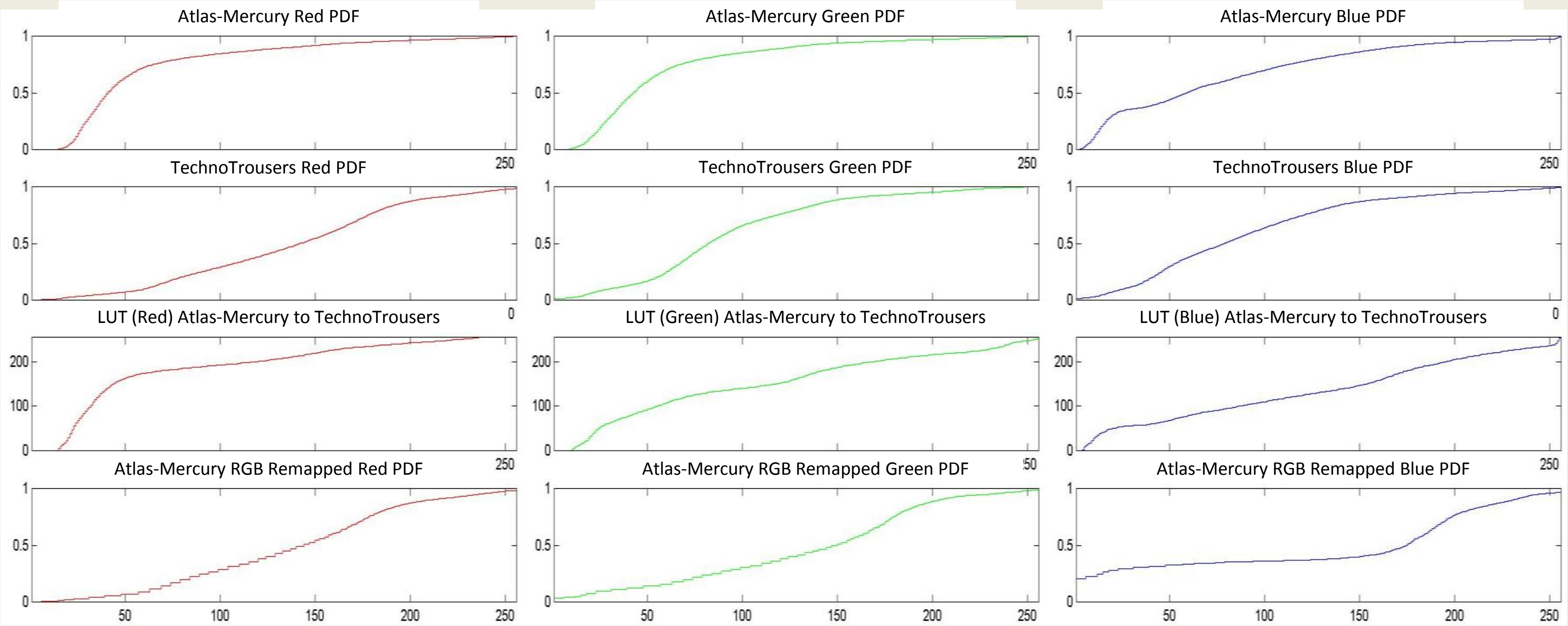


target

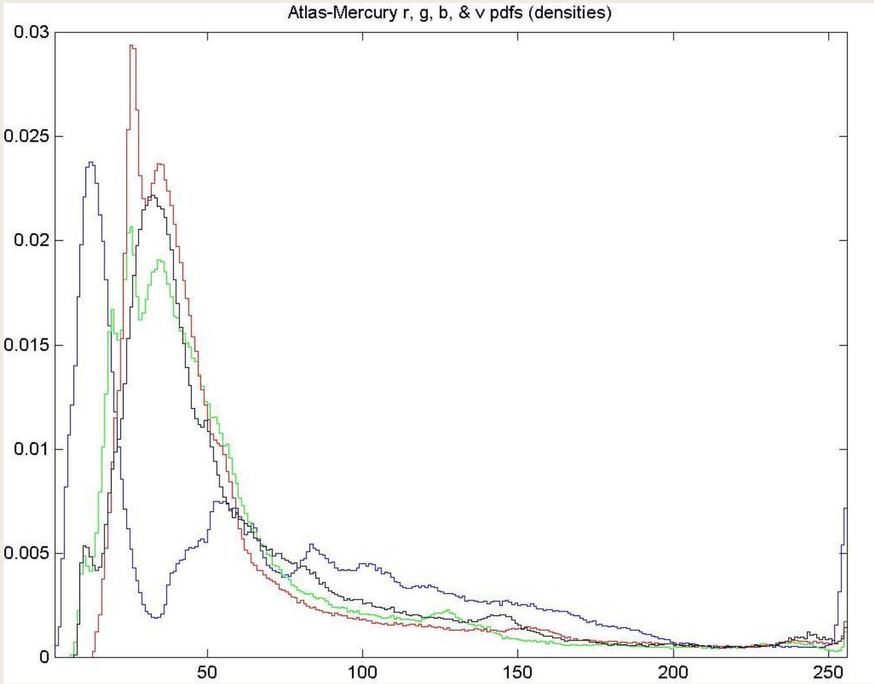


R, G, & B remapped

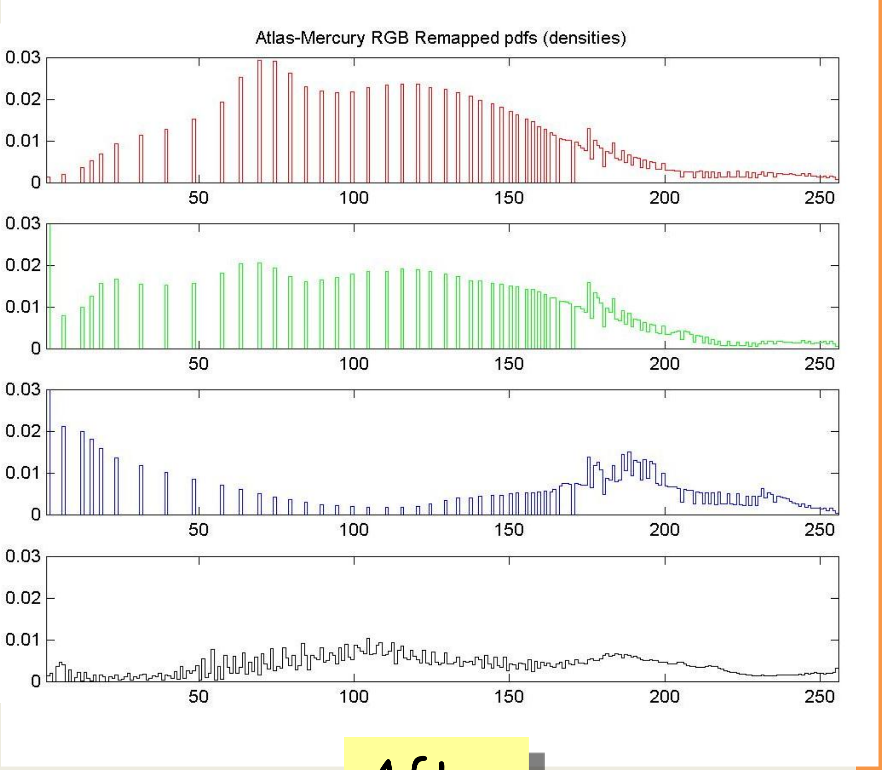
CDFs and the LUTs



Effects of RGB Remapping on pdfs

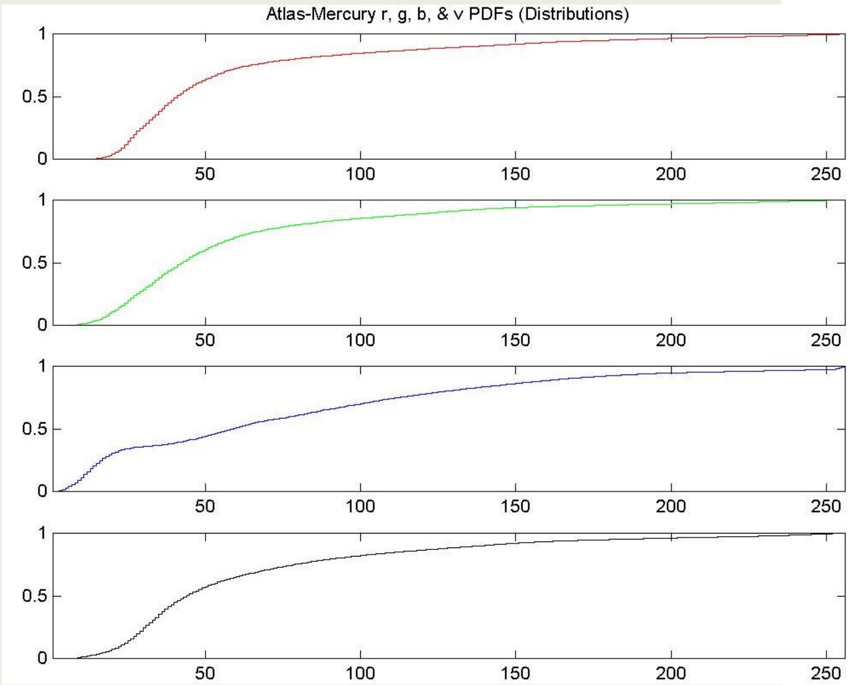


Before

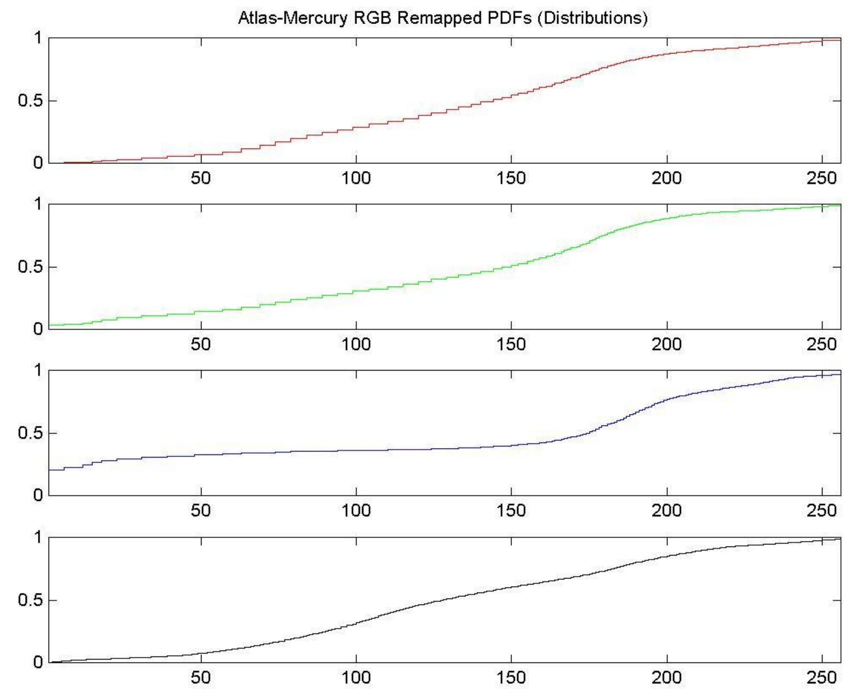


After

Effects of RGB Remapping on CDFs



Before



After

Remap an Image:

To Have Two of its Color pdfs Match the Third

original

G & B ← R

B & R ← G

R & G ← B