Digital Image Processing

Point Processing(点处理)

Point Processing of Images

- In a digital image, point = 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

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

Point Processing

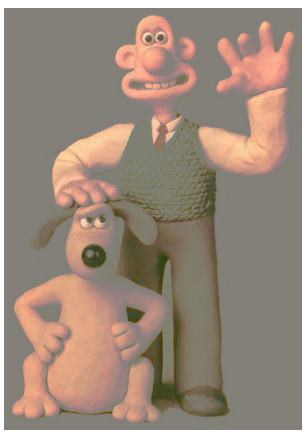
- gamma

- brightness

original

+ brightness

+ gamma



histogram mod

- contrast

original

+ contrast

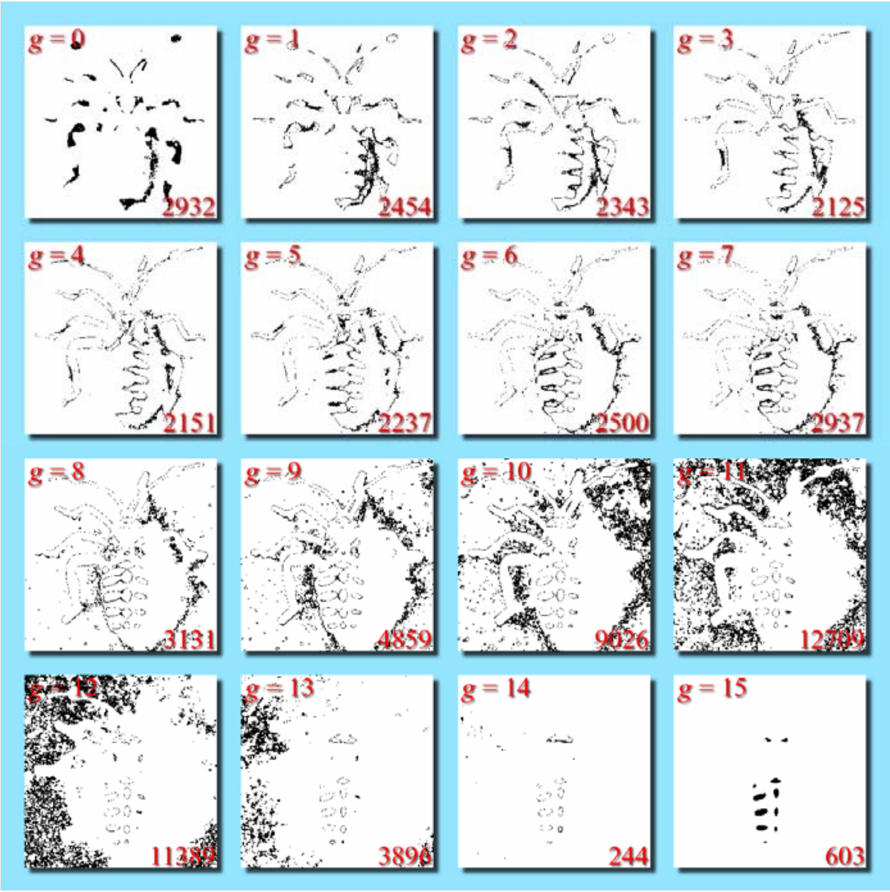
histogram EQ

The Histogram of a Grayscale Image 灰度直方图

- Let I be a 1-band (grayscale) image.
- $I(r,c)$ is an 8-bit integer between 0 and 255.
- Histogram, h_I, of I:
  - a 256-element array, h_I
  - $h_I(g)$, for g = 1, 2, 3, …, 256, is an integer
  - $h_I(g)$ = number of pixels in I that have value g-1.

The Histogram of a Grayscale Image
灰度直方图

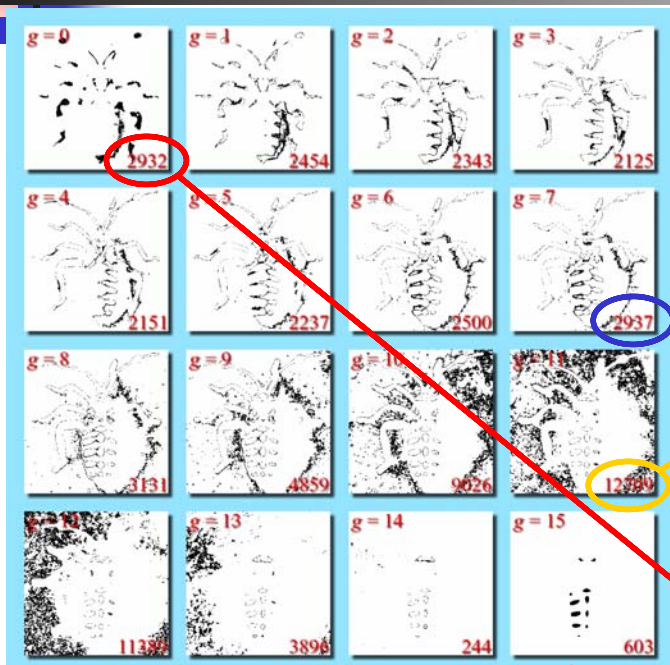
16-level (4-bit) image
Resolution: 256*256



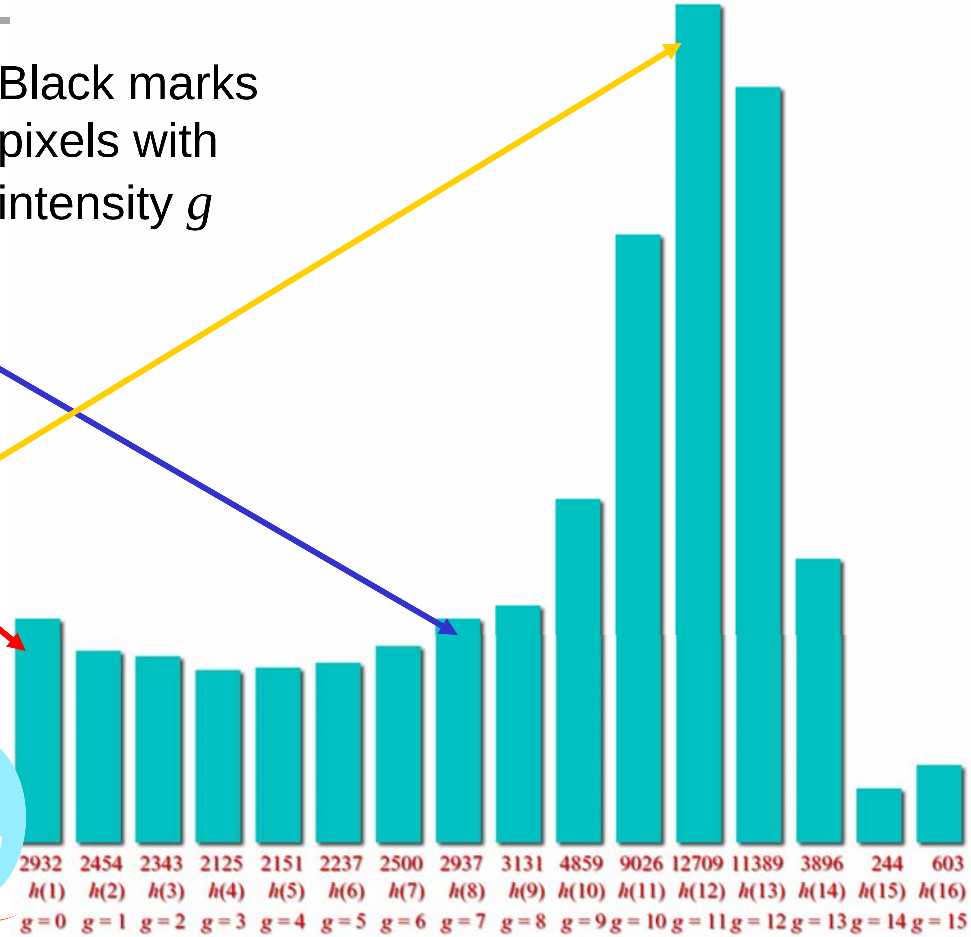
lower subscript: number of pixels with intensity g

black regions mark pixels with intensity g

The Histogram of a Grayscale Image



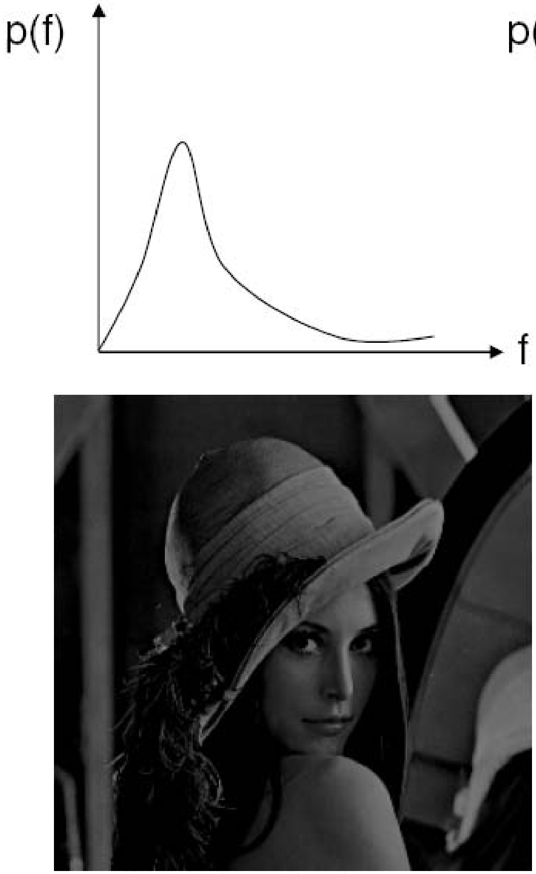
Black marks pixels with intensity g



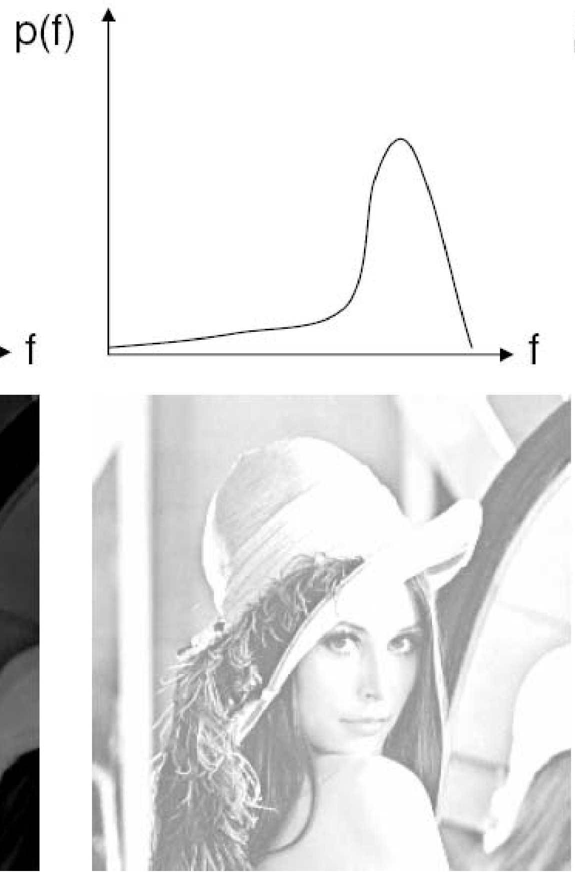
Plot of histogram:
number of pixels with intensity g

Sum(h(i))=?

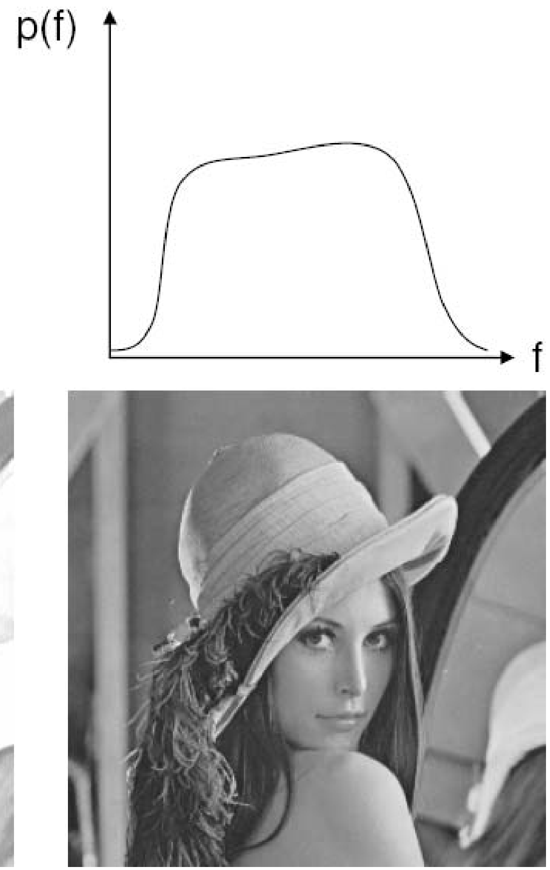
Examples of Histograms



(a) Too dark

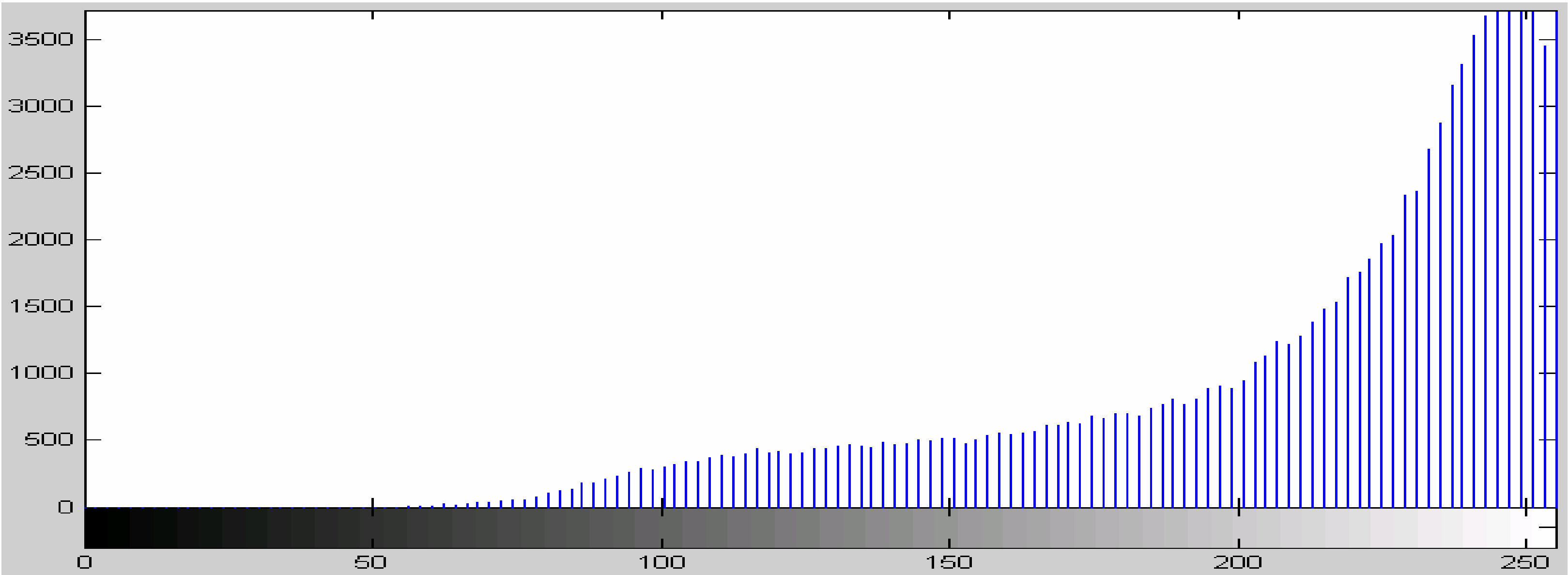


(b) Too bright

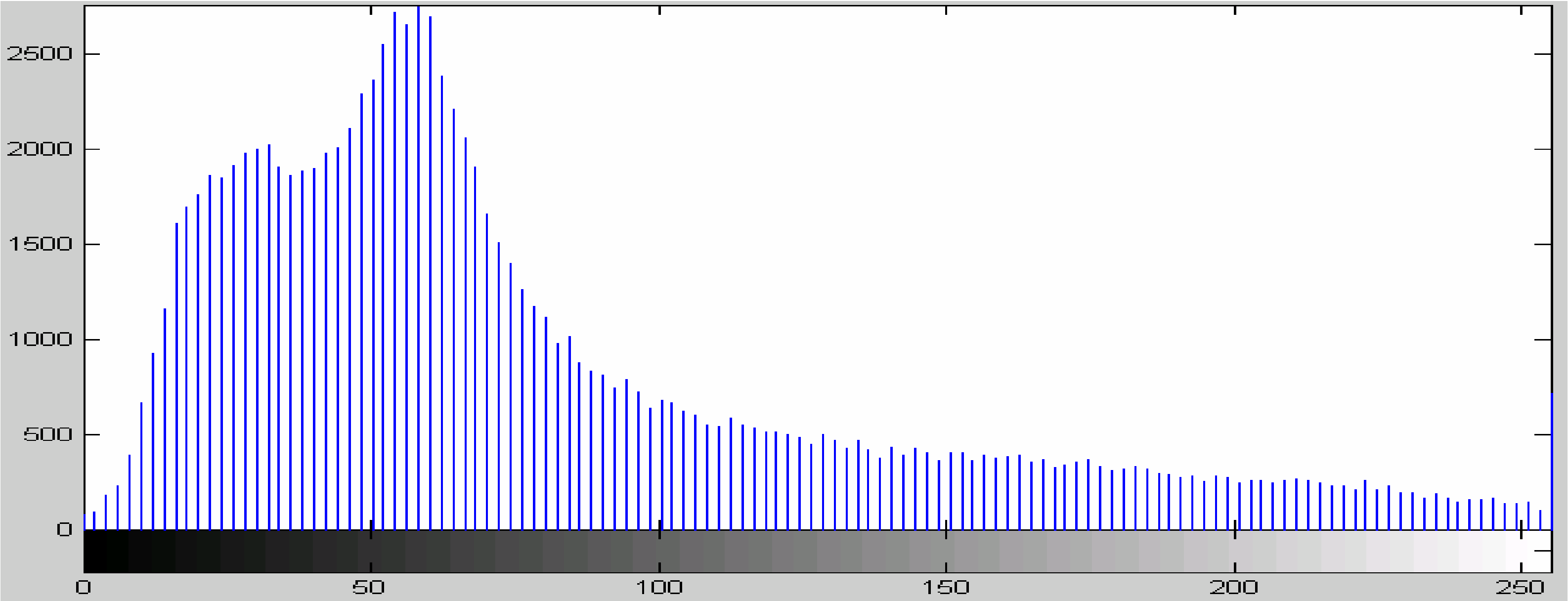


(c) Well balanced

Histogram Examples

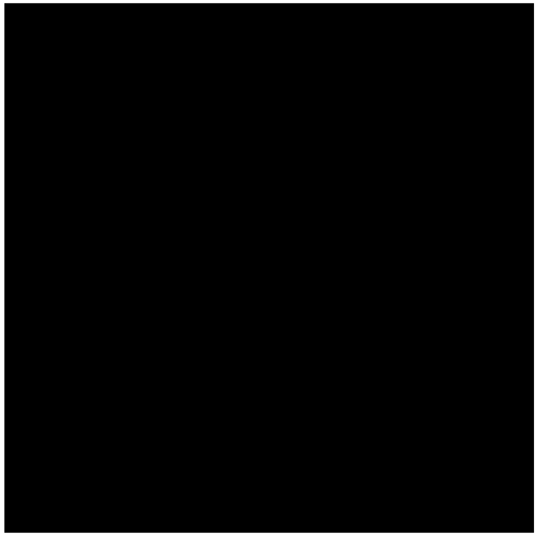


Histogram Examples



How is the histogram useful

- It is an easy method to statistically compare the contents of two images

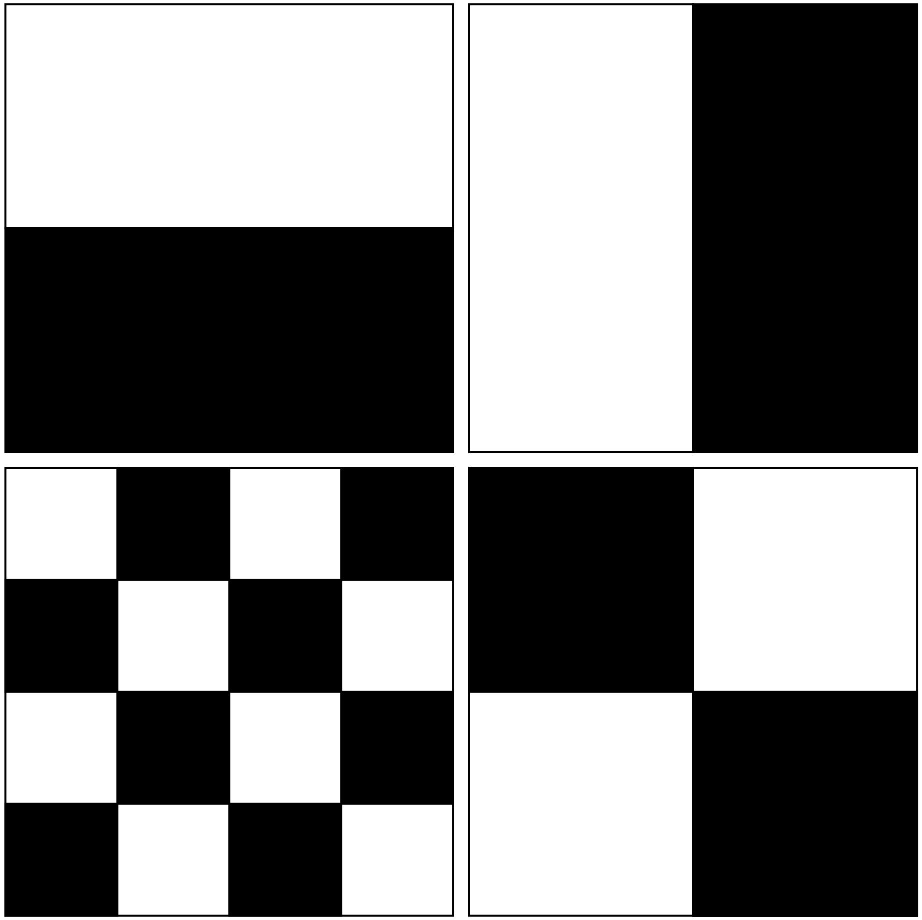


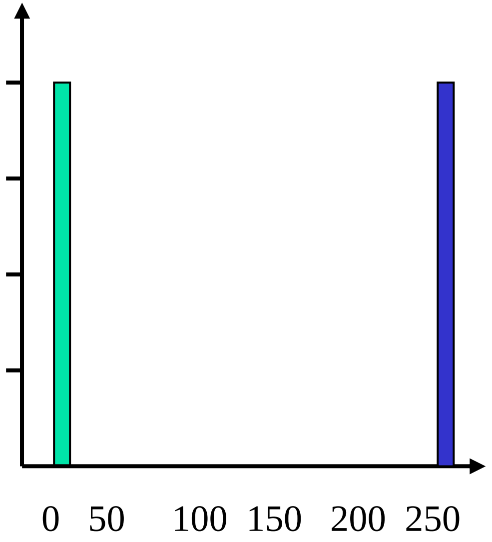
Pixelwise : ×

Histogram : √

- But……

Very different images may have same histograms

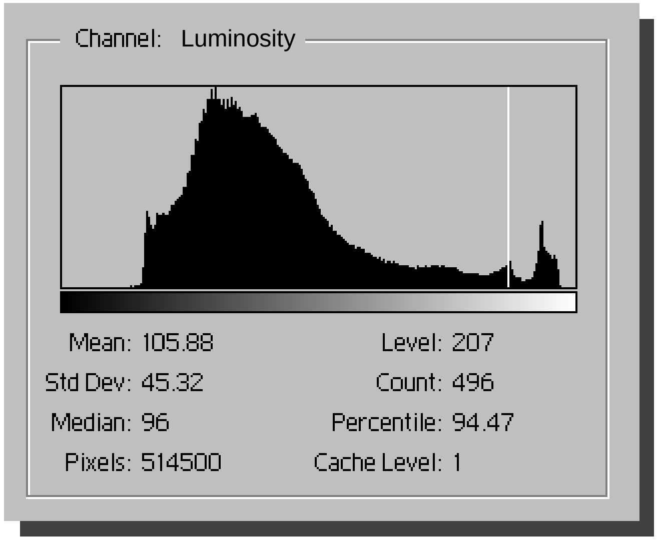




- Histogram reflects the pixel intensity distribution, not the spatial distribution!

The Histogram of a Grayscale Image

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

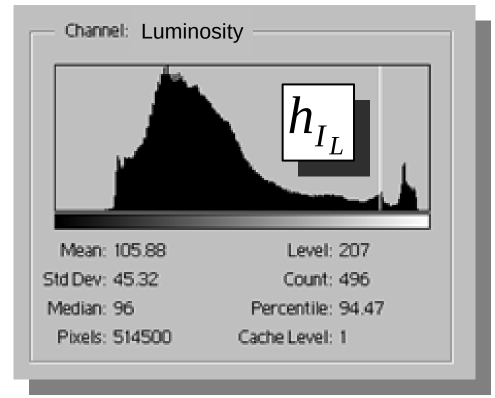


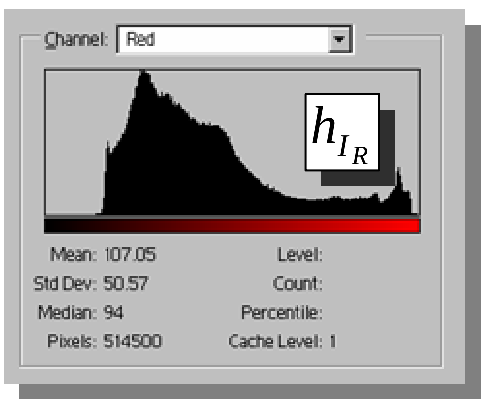
The Histogram of a Color Image
彩色直方图

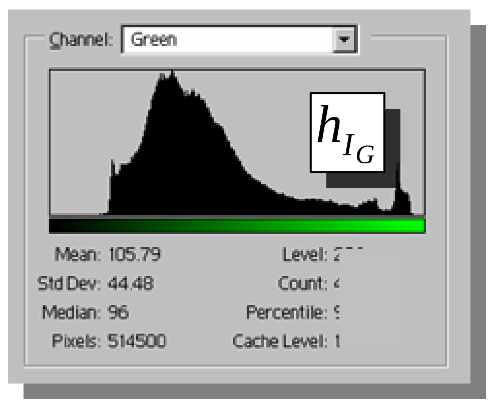
- If I is a 3-band(波段) RGB image (truecolor, 24-bit).
- Each of the 3 bands is an integer between 0 and 255.
- Then I has 3 histograms, for R,G,B components respectively:
  - $h_R(g+1)$ = # of pixels in I(:,:,1) with intensity value g
  - $h_G(g+1)$ = # of pixels in I(:,:,2) with intensity value g
  - $h_B(g+1)$ = # of pixels in I(:,:,3) with intensity value g

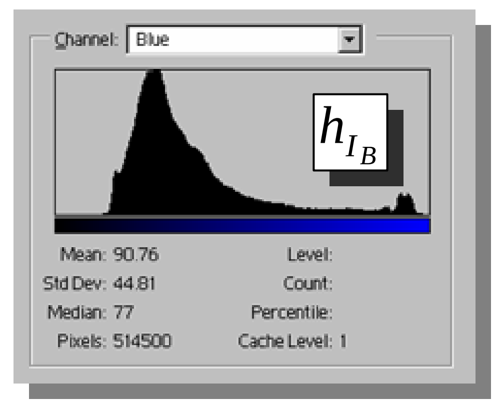
The Histogram of a Color Image

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



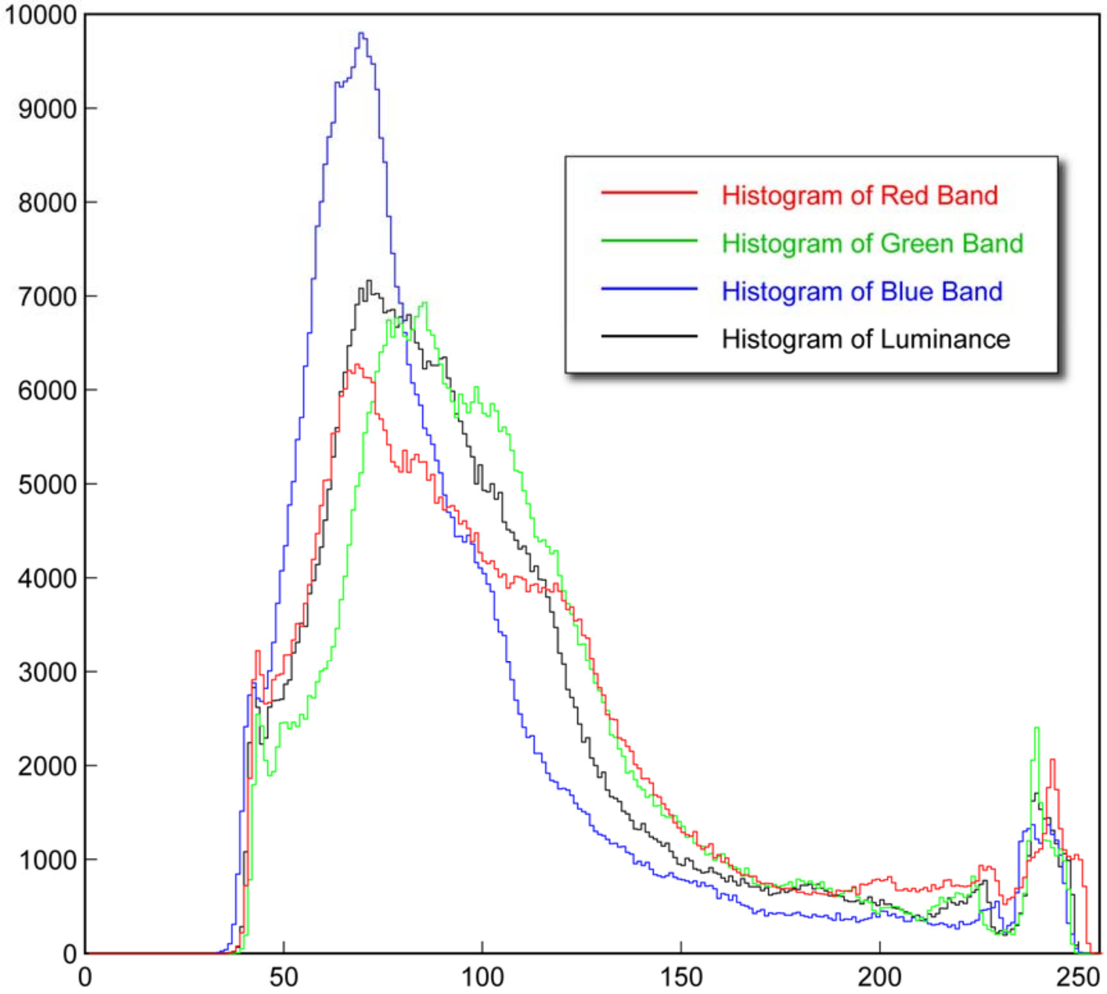






Luminosity: 光度

The Histogram of a Color Image



Value or Luminance Histograms 光度直方图和亮度直方图

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

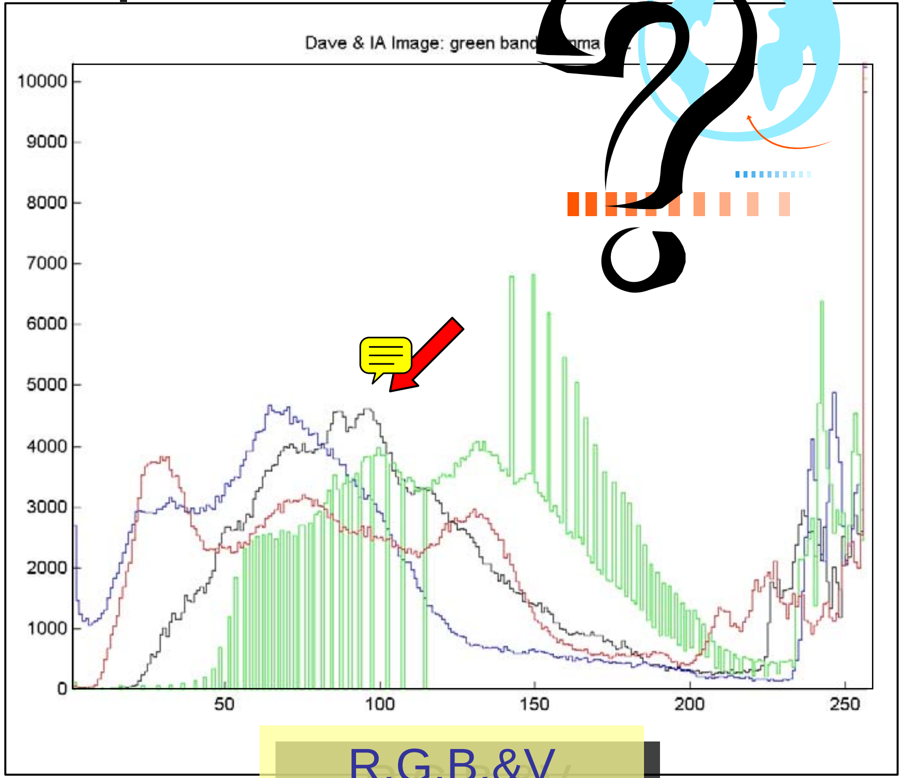
$$\boxed{V(r,c)=\frac{1}{3}\left[R(r,c)+G(r,c)+B(r,c)\right]}$$

Where *R*, *G*, and *B* are the red, green, and blue bands of *I*.

The luminance histogram(亮度直方图) of *I* is the histogram of the luminance image,

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

Value Histogram vs.
Avg. of R,G,B Histograms



R,G,B,&V histograms

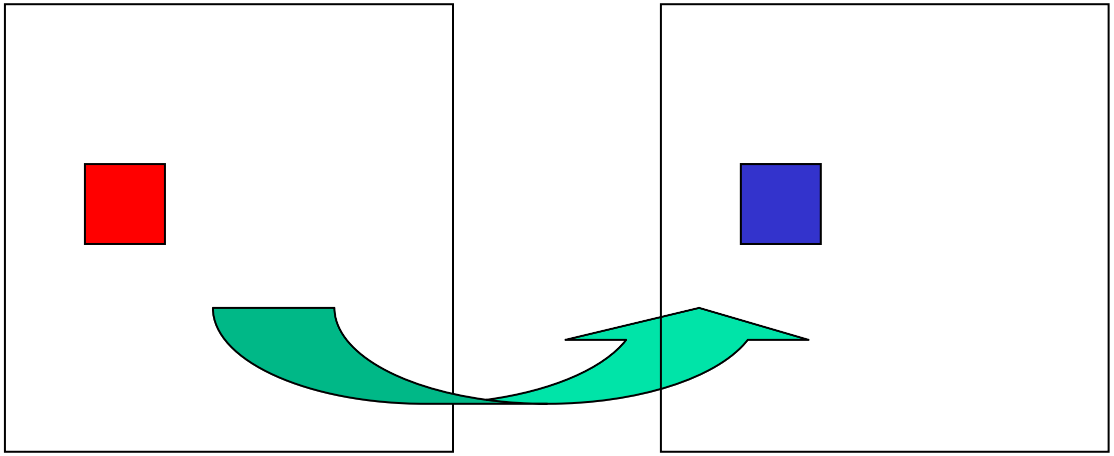
Value histogram : average followed by statistics;
Avg. of R,G,B histograms: statistics followed by average.

Multi-Band Histogram Calculator in Matlab

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

[R C B]=size(I);     % Row/Column/Band size

% 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;
```

Point Processing(点处理)

- $s=T(r)$
- r: gray-level at (x,y) in original image f(x,y)
- s: gray-level at (x,y) in processed image g(x,y)
- T is called gray-level transformation(变换) or mapping(映射)



Point Operations via Functional Mappings

Image: I → Φ, point operator → J

$$J = \Phi\left[I\right]$$

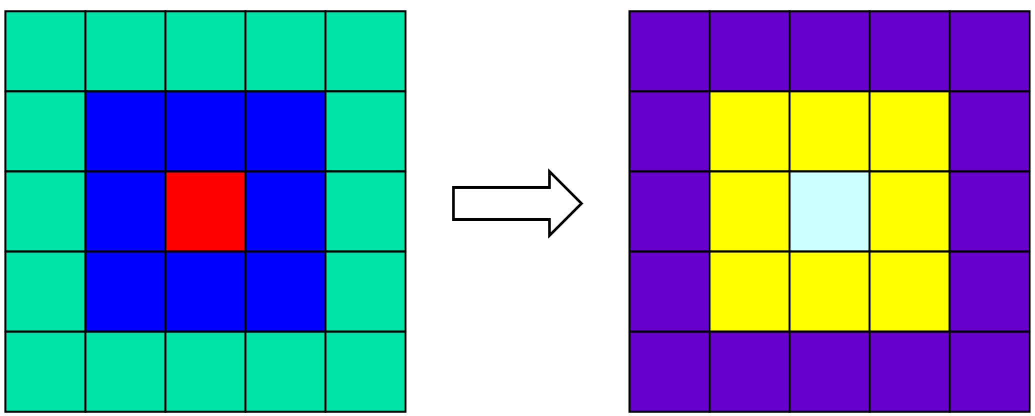
Input | Output

Pixel: I(r,c) → function, f → J(r,c)

If $I(r,c) = g$

and $f(g) = k$

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



Point Operations via Functional Mappings

One-band Image

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

Three-band Image

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

Point Operations using Look-up Tables(对照表)

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

If $k = f(g)$,

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

and if k takes on

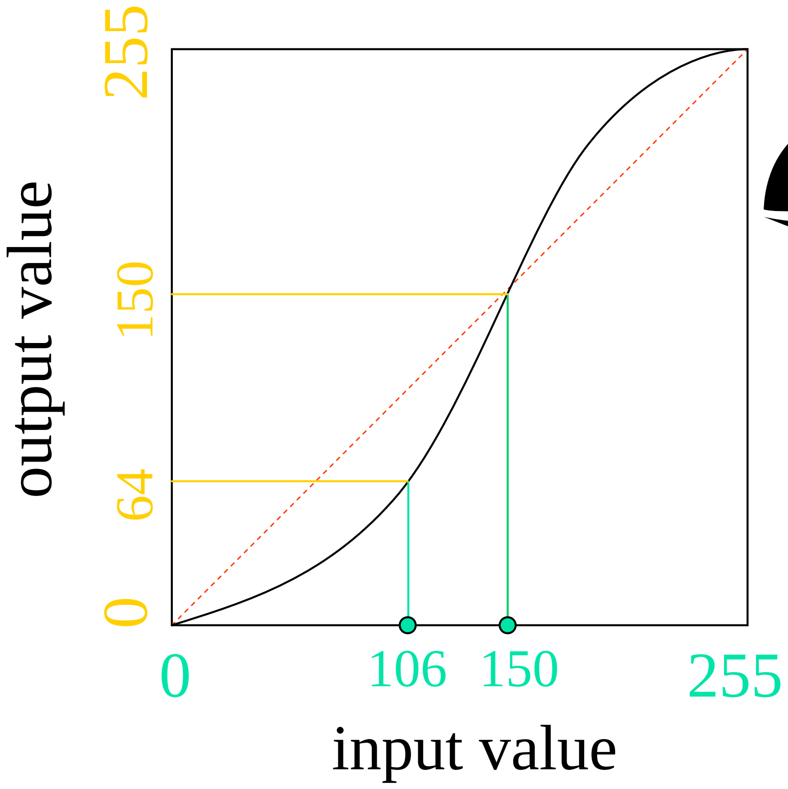
values in $\{0, \ldots, 255\}$, …

… 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 = Look-up Table Ops



E.g.:	index	value
	...	...
	101	57
	102	61
	103	62
	104	63
	105	63
	106	64
	...	...
	input	output

Point Operations using Look-up Tables

If I is 3-band, then

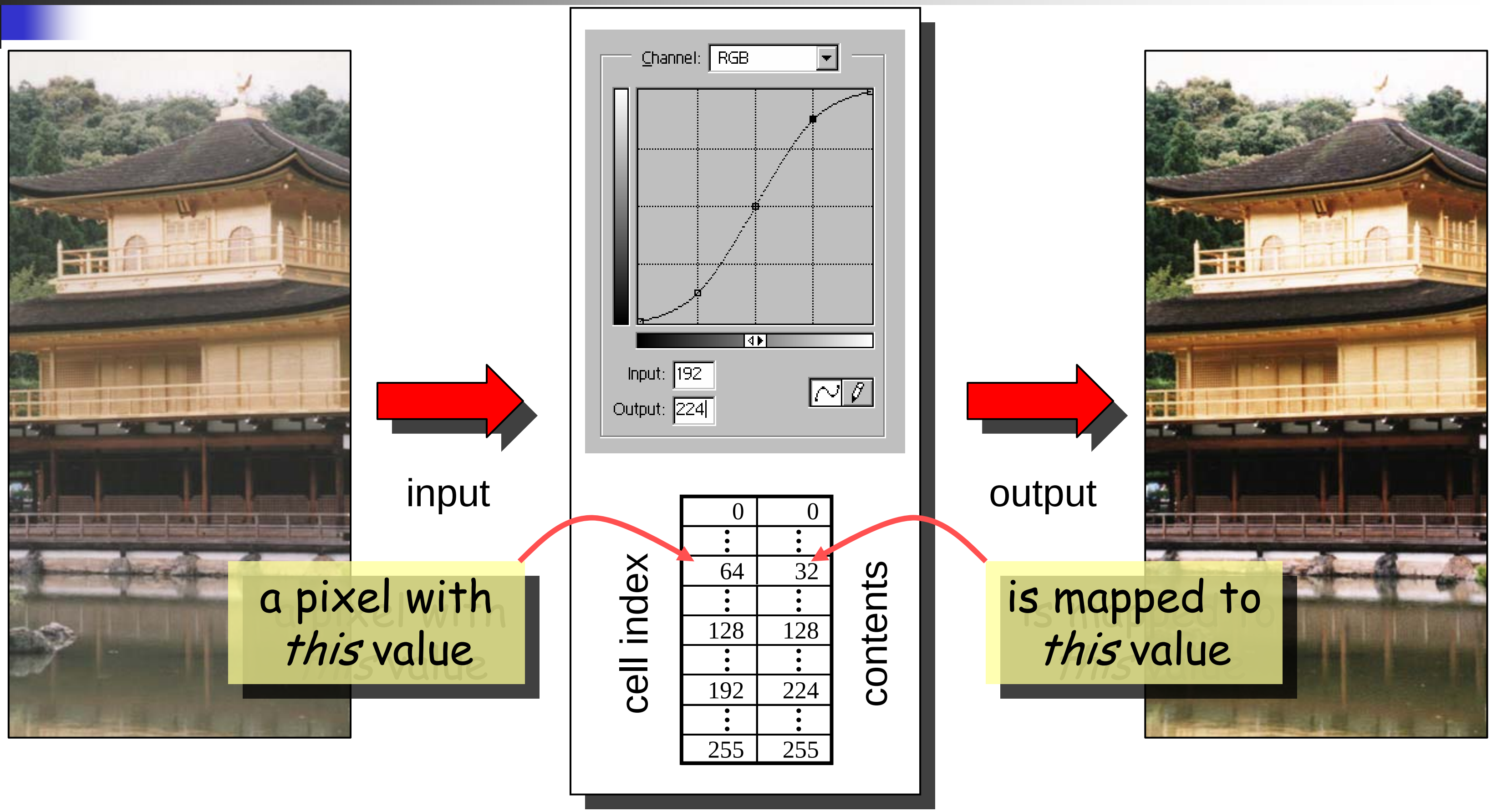
a) each band is mapped separately using the same LUT for each band *or*

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

a) $J = \text{LUT}\left(I + 1\right)$, *or*

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

Look-Up Tables(对照表)



How to Generate a Look-Up Table

For example:

$$\text{Let } a = 2.$$

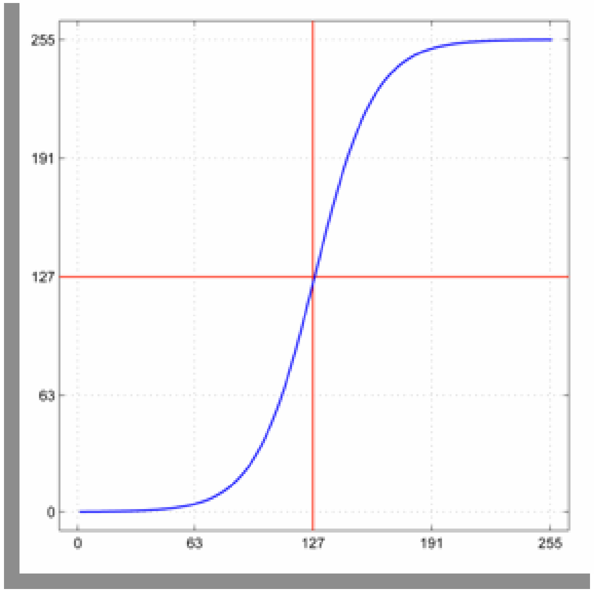
$$\text{Let } x \in \{0,\ldots,255\}$$

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

Or in Matlab:

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

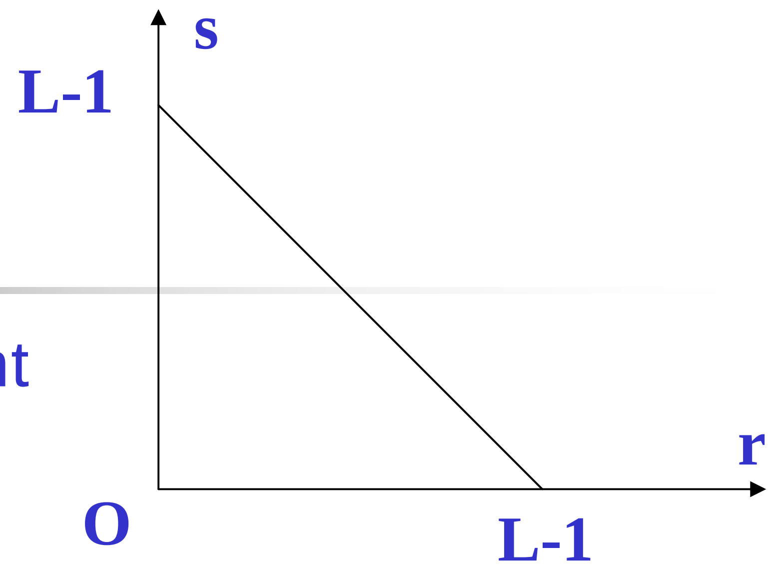
This is just one example.



- Digital inverse: s=L-1-r
- Power-law transformation: $s=cr^{\gamma}$
- Log transformation: s=clog(1+r)

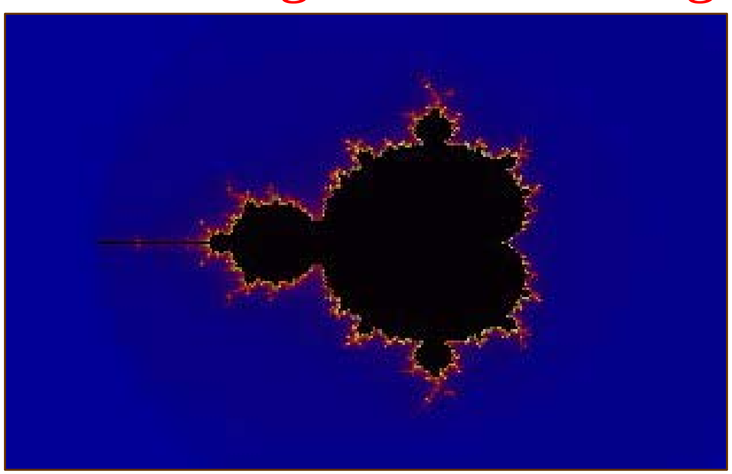
Image Negative

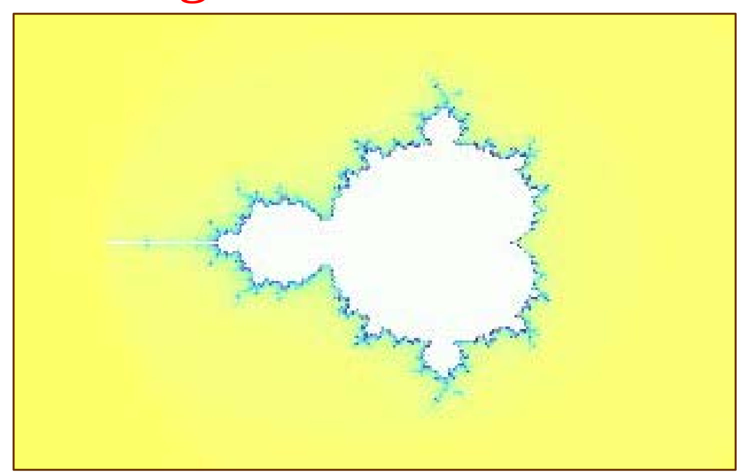
- Convert the color to its complement



$$s=(L-1)-r$$

What is the relationship between the histogram of the original and the negative image?





Power-law transformations(指数型变换)

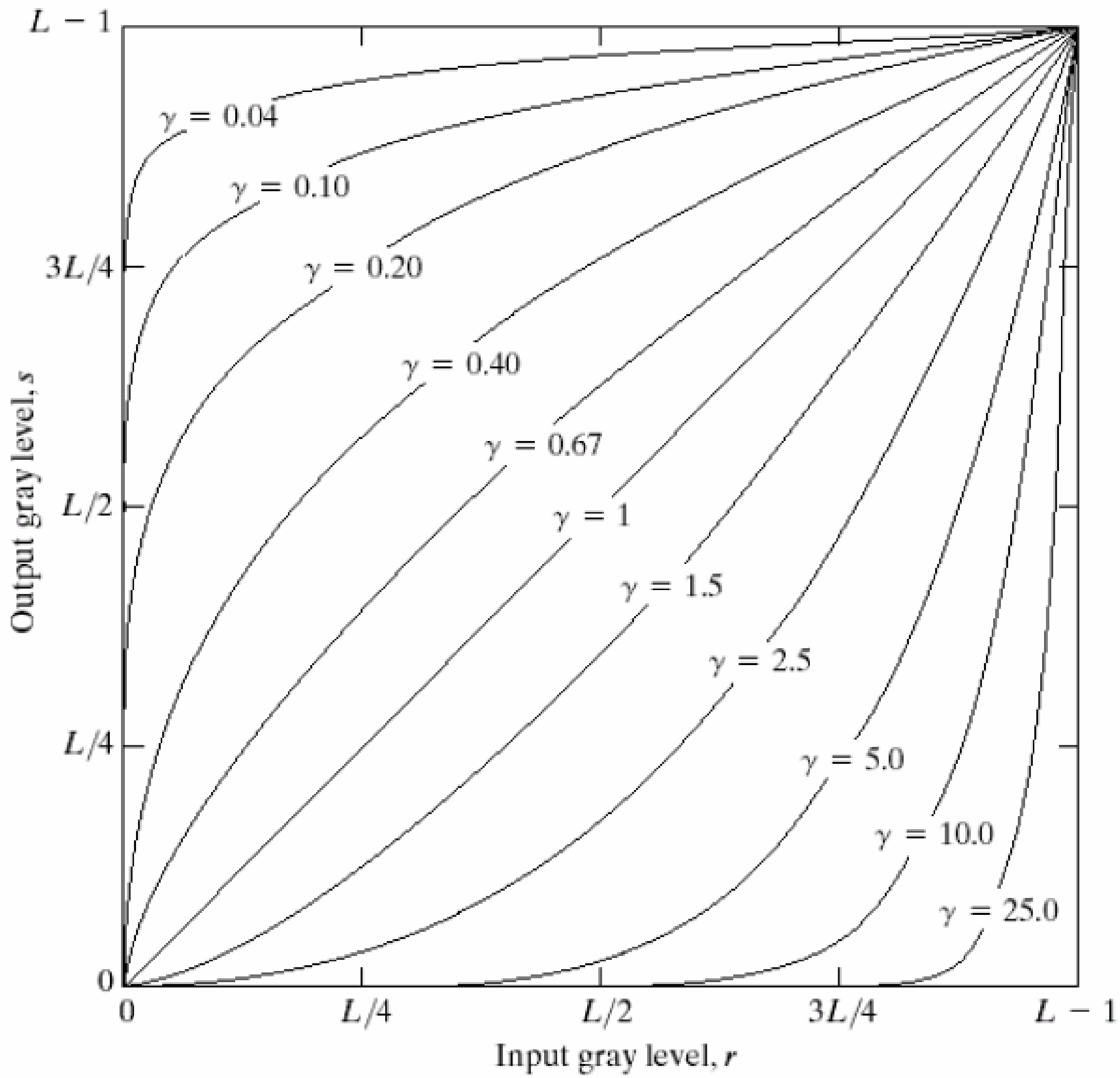
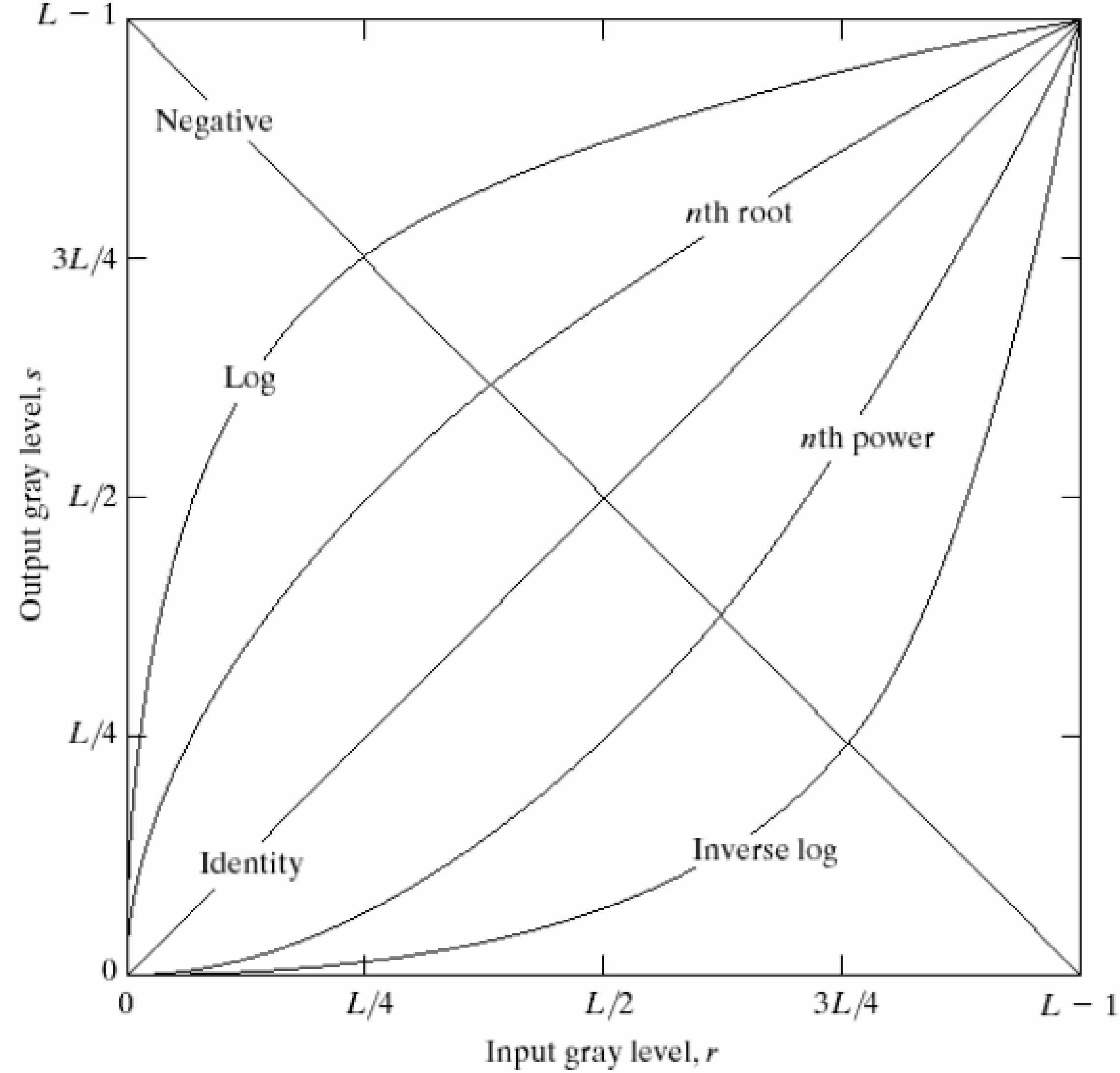


FIGURE 3.6 Plots of the equation $s = cr^{\gamma}$ for various values of γ ($c = 1$ in all cases).

Power-law transformation: $s=cr^{\gamma}$

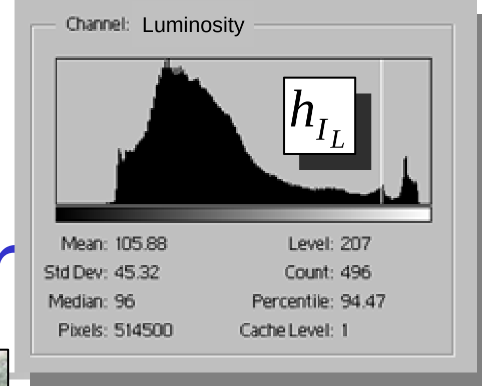
Log transformations(对数型变换)

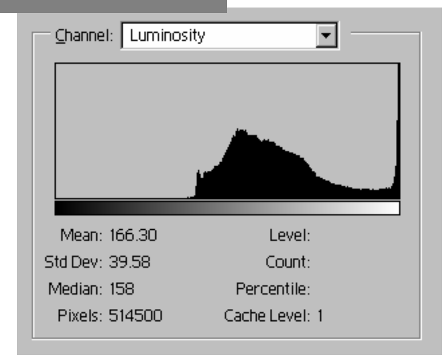
FIGURE 3.3 Some basic gray-level transformation functions used for image enhancement.

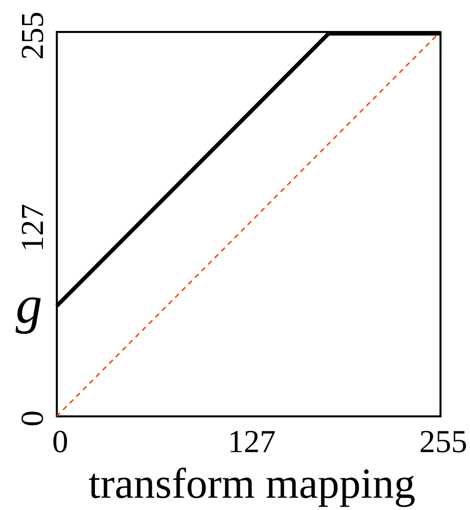


Log transformation: s=clog(1+r)

Point Processes: Increase Brigh



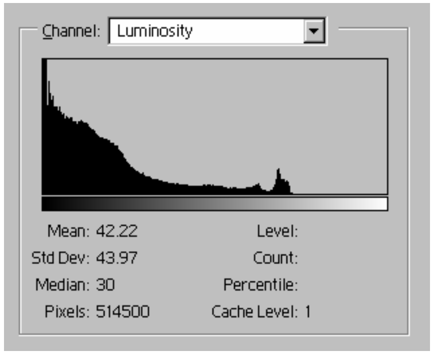


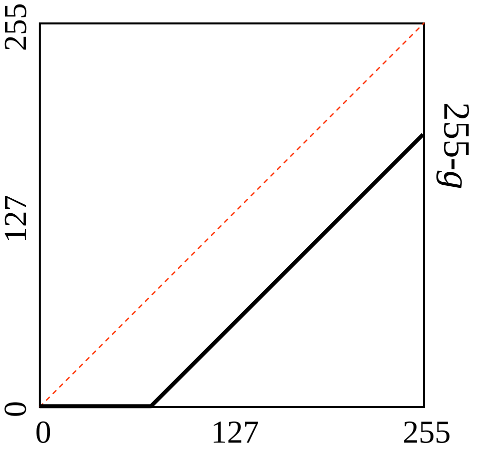


$$J_k(r,c) = \begin{cases} I_k(r,c) + g, & \text{if } I_k(r,c) + g < 256 \\ 255, & \text{if } I_k(r,c) + g > 255 \end{cases}$$

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

Point Processes: Decrease Brightness





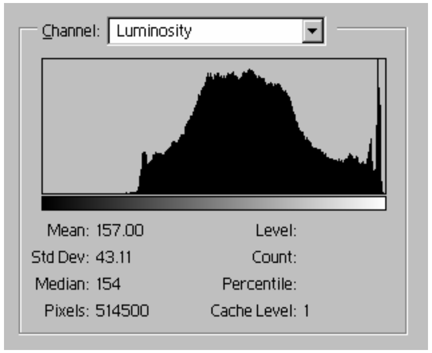
transform mapping

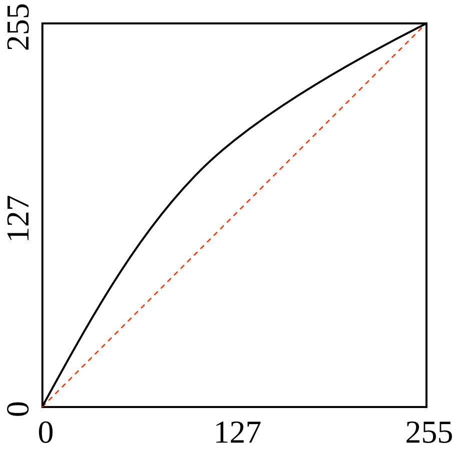
$$J_k(r,c) = \begin{cases} 0, & \text{if } I_k(r,c) - g < 0 \\ I_k(r,c) - g, & \text{if } I_k(r,c) \end{cases}$$

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

Point Processes: Increased Gamma

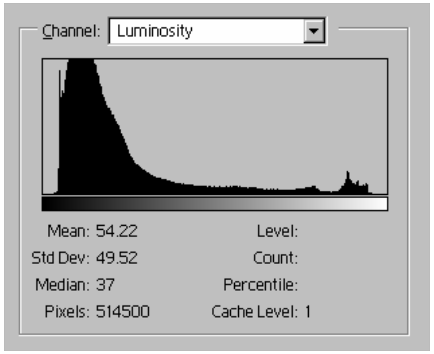
$$J(r,c) = 255 \cdot \left[\frac{I(r,c)}{255} \right]^{1/\gamma} \quad \text{for} \quad \gamma > 1.0$$

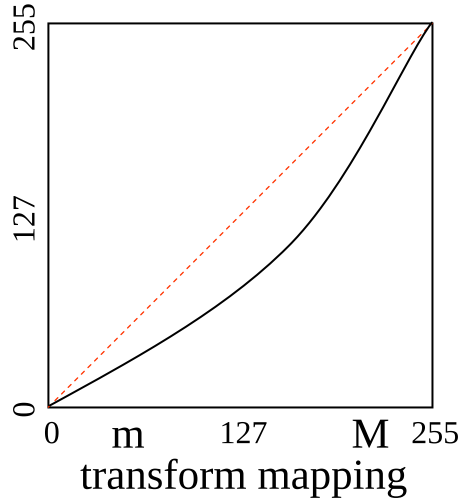




transform mapping

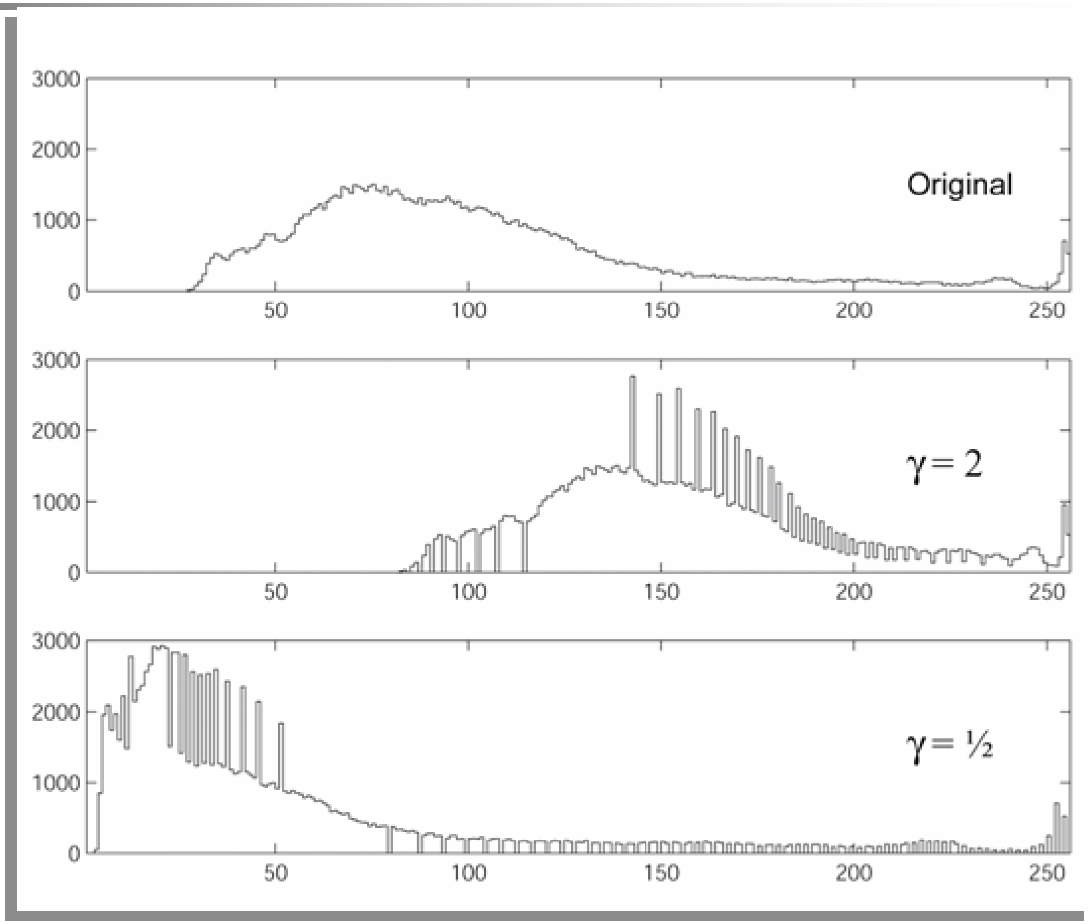
Point Processes: Decreased Gamma



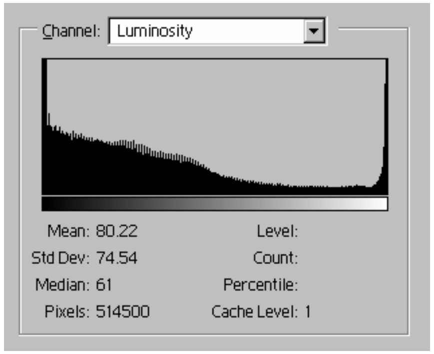


$$J(r,c) = 255 \cdot \left[\frac{I(r,c)}{255}\right]^{1/\gamma} \quad \text{for} \quad \gamma < 1.0$$

Gamma Correction: Effect on Histogram

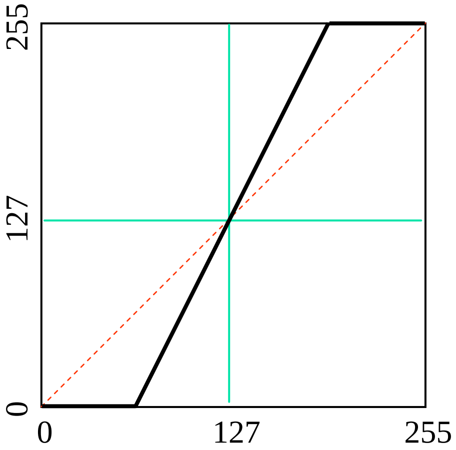


Point Processes: Increase Contrast



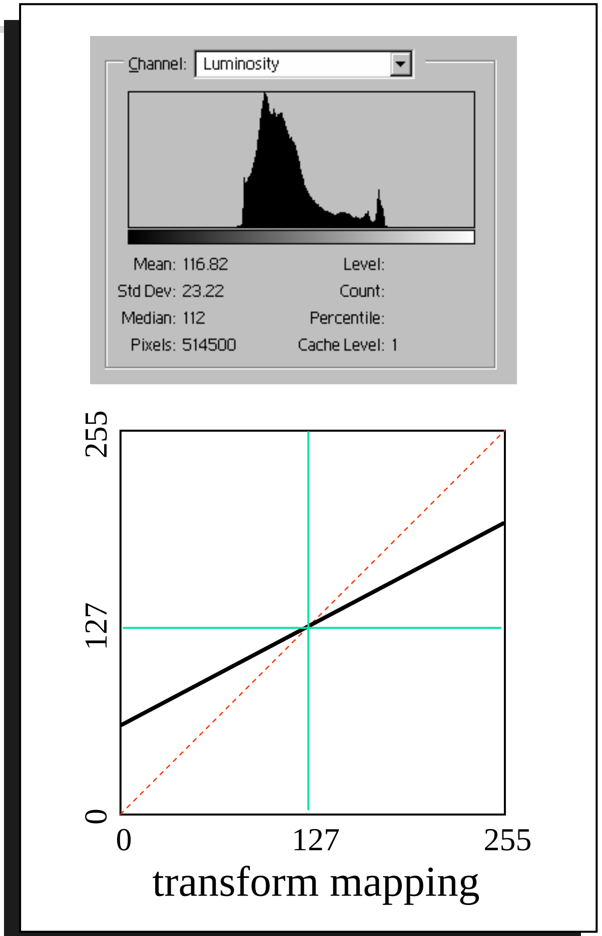
Let $T_k(r,c) = a\left[I_k(r,c) - 127\right] + 127,$ where $a > 1.0$

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



transform mapping

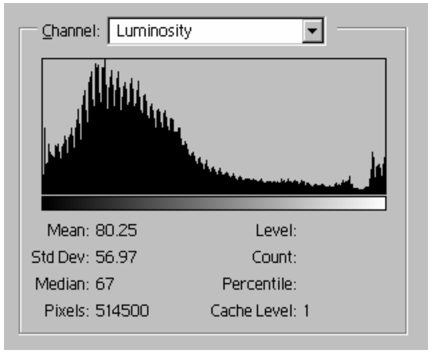
Point Processes: Decrease Contrast



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

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

Point Processes: Contrast Stretch

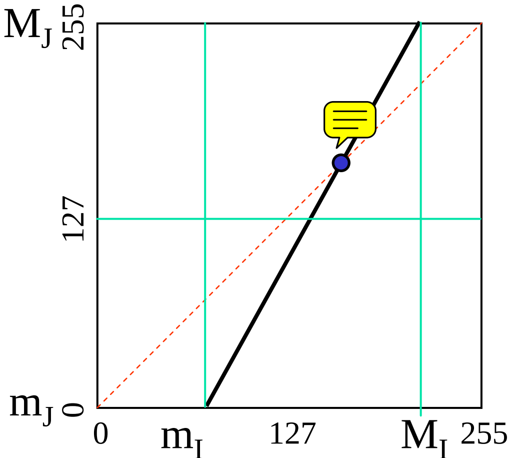


Let $m_I = \min[I(r,c)], \quad M_I = \max[I(r,c)],$

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

Then,

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



The watershed value m:

The gray levels below m are darkened and the levels above m are brightened.

Contrast stretch

- Contrast Stretching: to get an image with higher contrast than the original image

Original

Enhanced

Contrast stretch

- Limiting case: produces a binary image (two level) from the input image

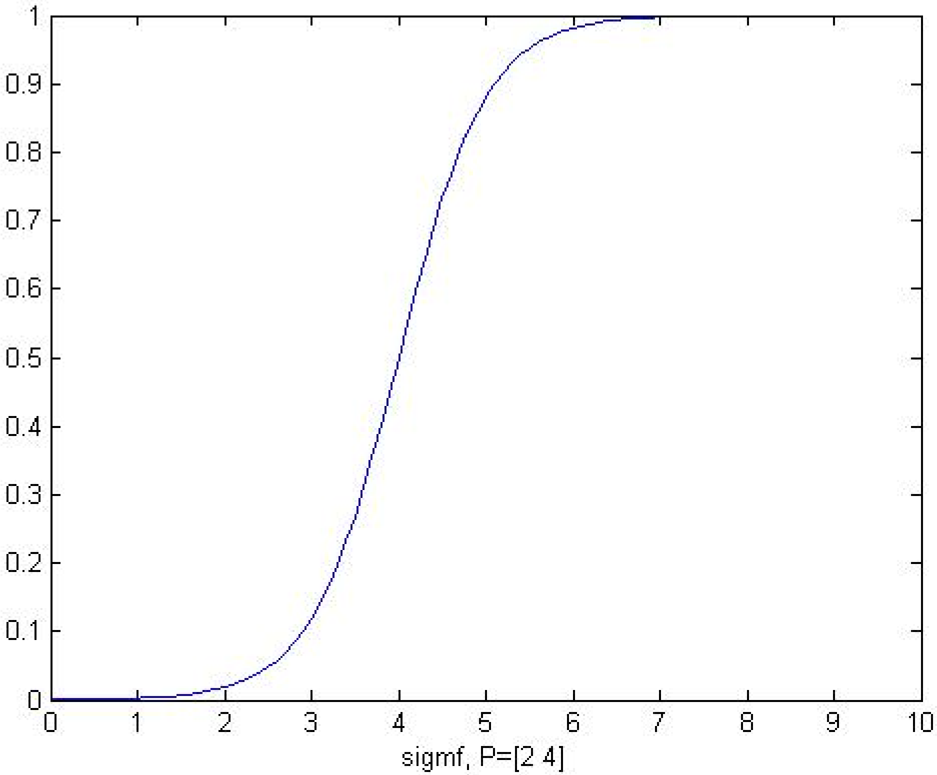
Original **Enhanced**

How to realize the S-curve for constrast stretch

- Strategy 1: sine function $y = \sin(x)$
- Strategy 2: inverse tangent function $y = \arctan(x)$
- Strategy 3: inverse function of $y=x^3$
- Strategy 4: sigmoid function $y = \frac{1}{1+e^{-a(x-c)}}$

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

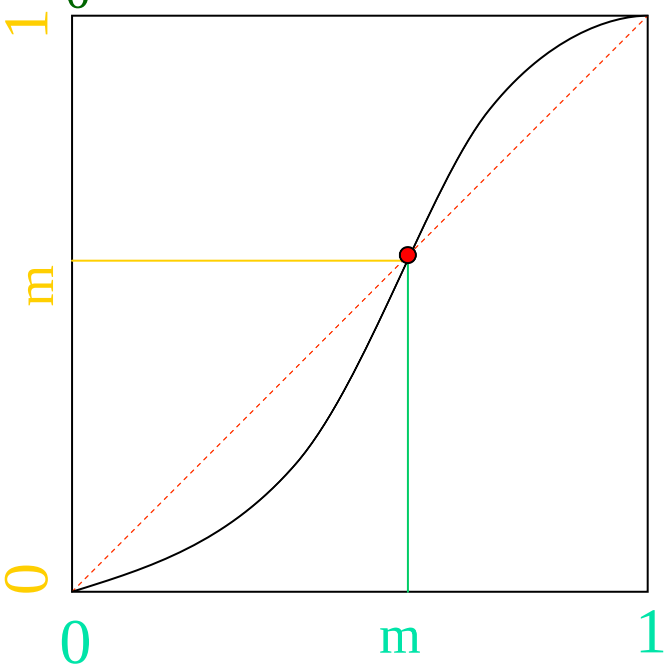
```
x=0:0.1:10;
y=sigmf(x,[2 4]);
plot(x,y)
xlabel('sigmf, P=[2 4]')
```



How to realize Contrast stretch

Can you design a polynomial S-curve as the contrast stretch mapping function?

e.g. $f(x)=a_n x^n+a_{n-1}x^{n-1}+\ldots+a_2x^2+a_1x+a_0$



Requirement:

$f(0)=0$

$f(1)=1$

$f'(0)=0$

$f'(1)=0;$

The Probability Density Function(pdf) of an Image(概率密度函数)

pdf
[lower case]

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

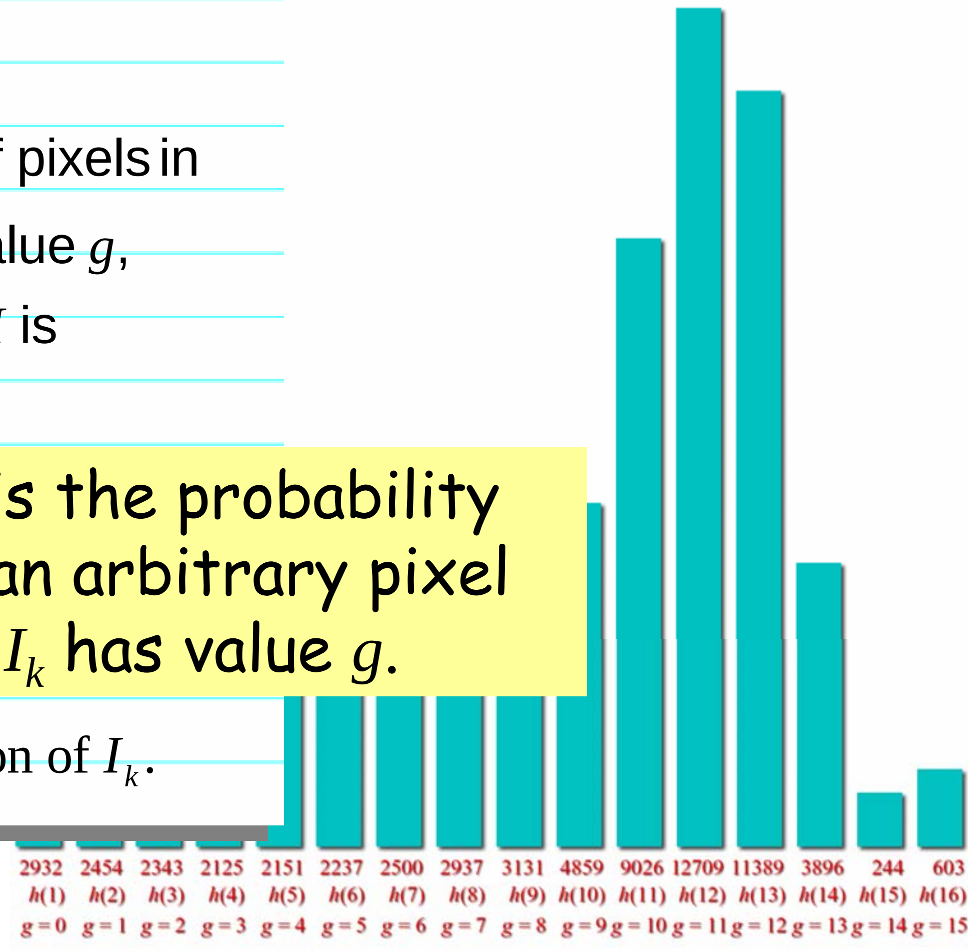
Note that since $h_{I_k}(g+1)$ is the number of pixels in I_k (the kth color band of image I) with value g, A is the number of pixels in I. That is if I is R rows by C columns then $A = R \times C$.

Then,

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

is the graylevel probability density function of I_k.

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



The Probability Density Function(pdf) of an Image

- $p_{\mathrm{band}}(g+1)$ is the fraction of pixels in (a specific band of) an image that have intensity value g.
- $p_{\mathrm{band}}(g+1)$ is the probability that a pixel randomly selected from the given band has intensity value g.
- the sum of the histogram $h_{\mathrm{band}}(g+1)$ over all g from 0 to 255 is equal to the number of pixels in the image,
- the sum of $p_{\mathrm{band}}(g+1)$ over all g is 1.
- p_{band} is the normalized histogram(归一化直方图) of the band.

The Cumulative Distribution Function(cdf) of an Image(累积概率分布函数)

cdf
[lower case]

Let **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 $q_k \leqslant g$ 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 *k*th band of *I*.

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

The Cumulative Distribution Function(cdf) of an Image

a.k.a. Probability Distribution Function(PDF).

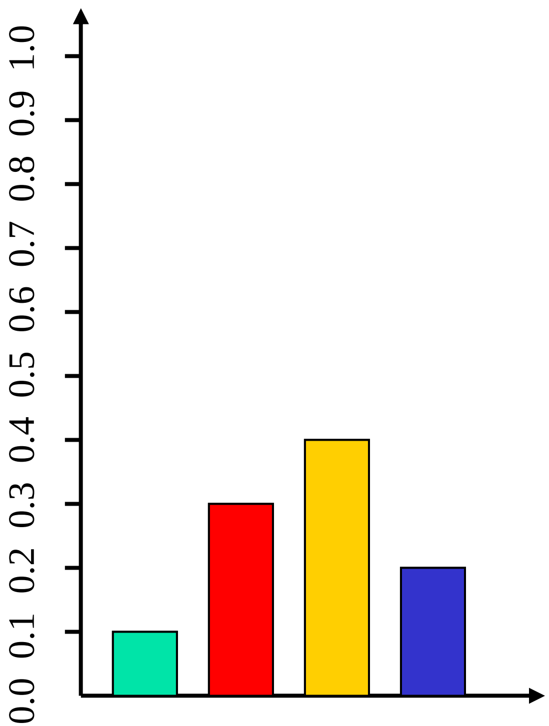
- $P_{\text{band}}(g+1)$ is the fraction of pixels in (a specific band of) an image that have intensity values $\boxed{\text{less than or equal to } g.}$
- $P_{\text{band}}(g+1)$ is the probability that a pixel randomly selected from the given band has an intensity value $\boxed{\text{less than or equal to } g.}$
- $P_{\text{band}}(g+1)$ 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 non-decreasing(非减函数).

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.

a.k.a. : also known as

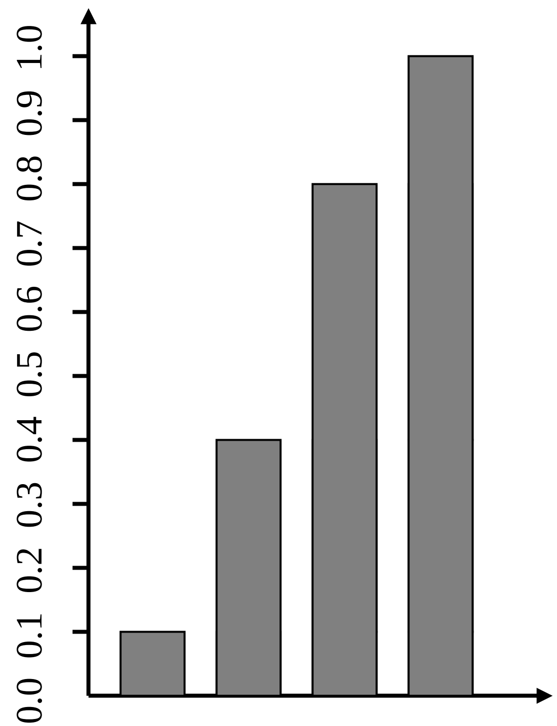
Point Processes:
Histogram Equalization(直方图均衡化)

- The normalized histogram is a probability density function (pdf)
- From the pdf, build the cumulative distribution function (cdf)



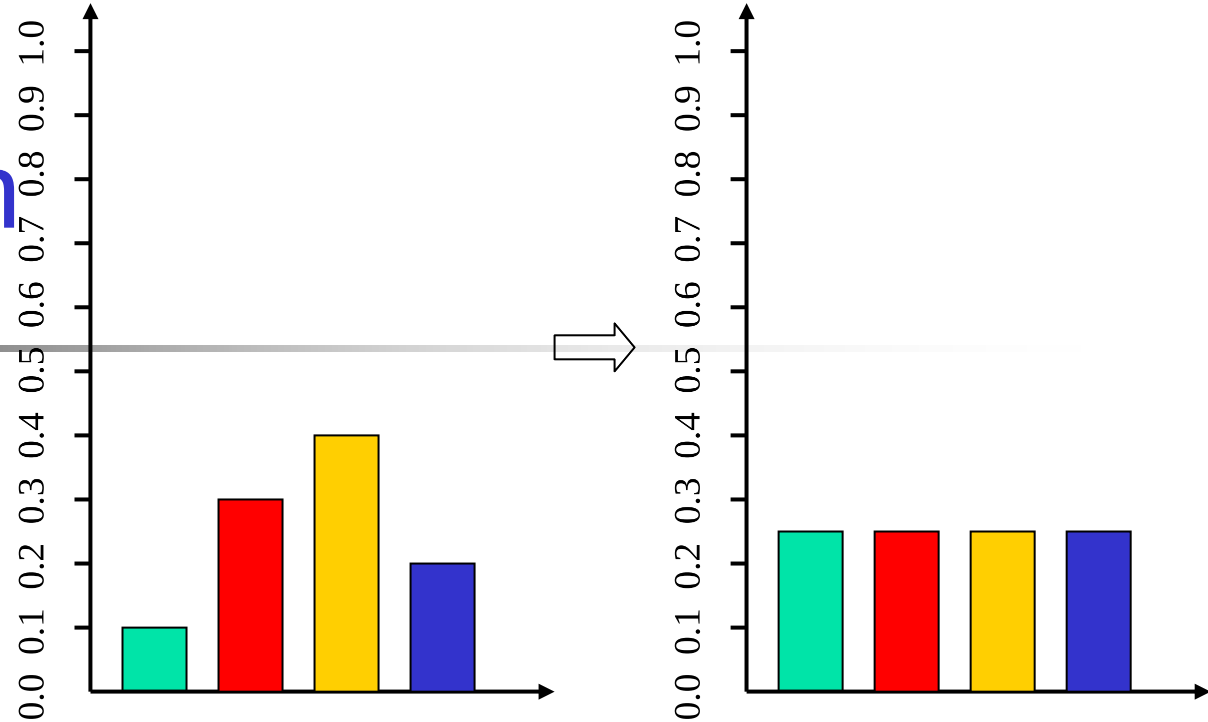
pdf

→

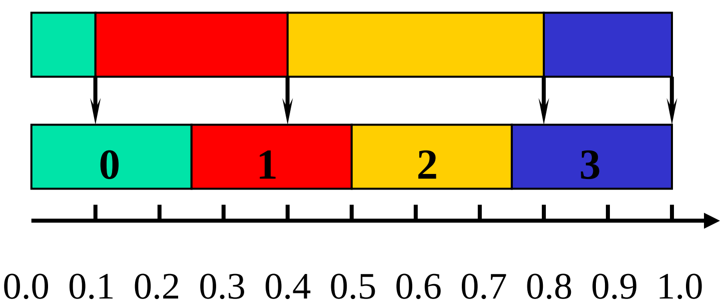


cdf

Histogram Equalization



- Histogram equalization tries to match the pdf of the result image to the uniform pdf.
- It is easier to implement by working on the cdf.



index	value
0	0
1	1
2	3
3	3
input	output

Any other Solutions?

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.

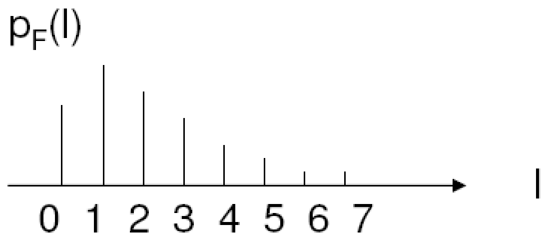
Then J has, as closely as possible, the correct histogram if

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

The CDF itself is used as the LUT.

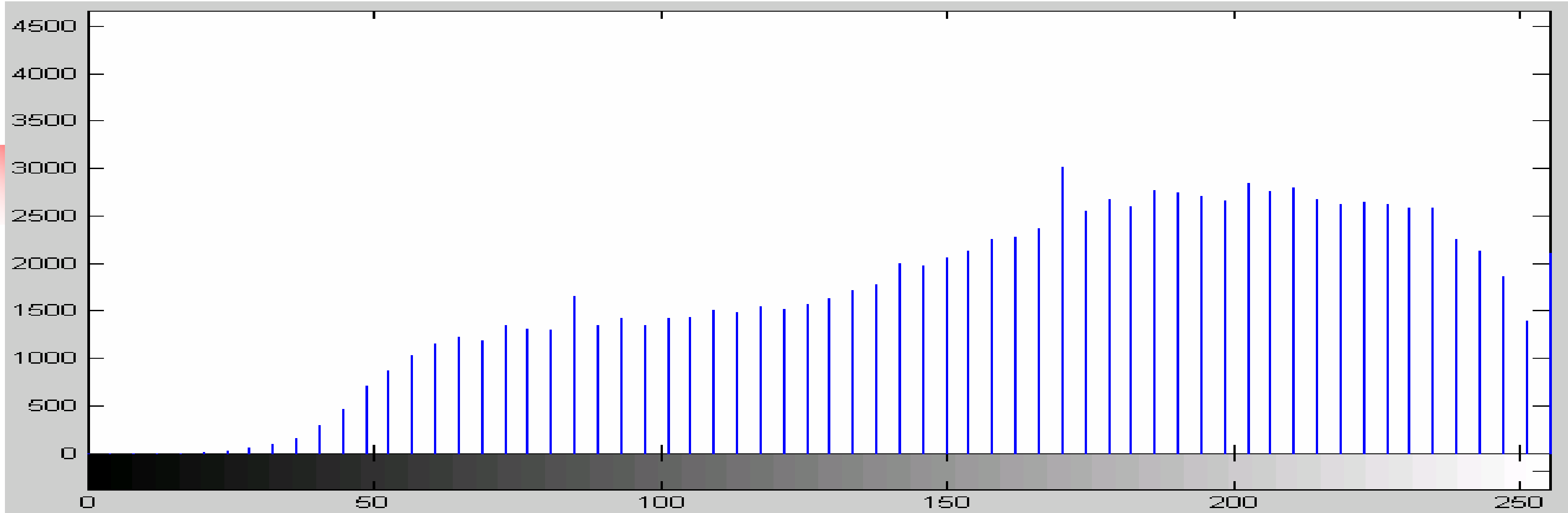
all bands processed similarly

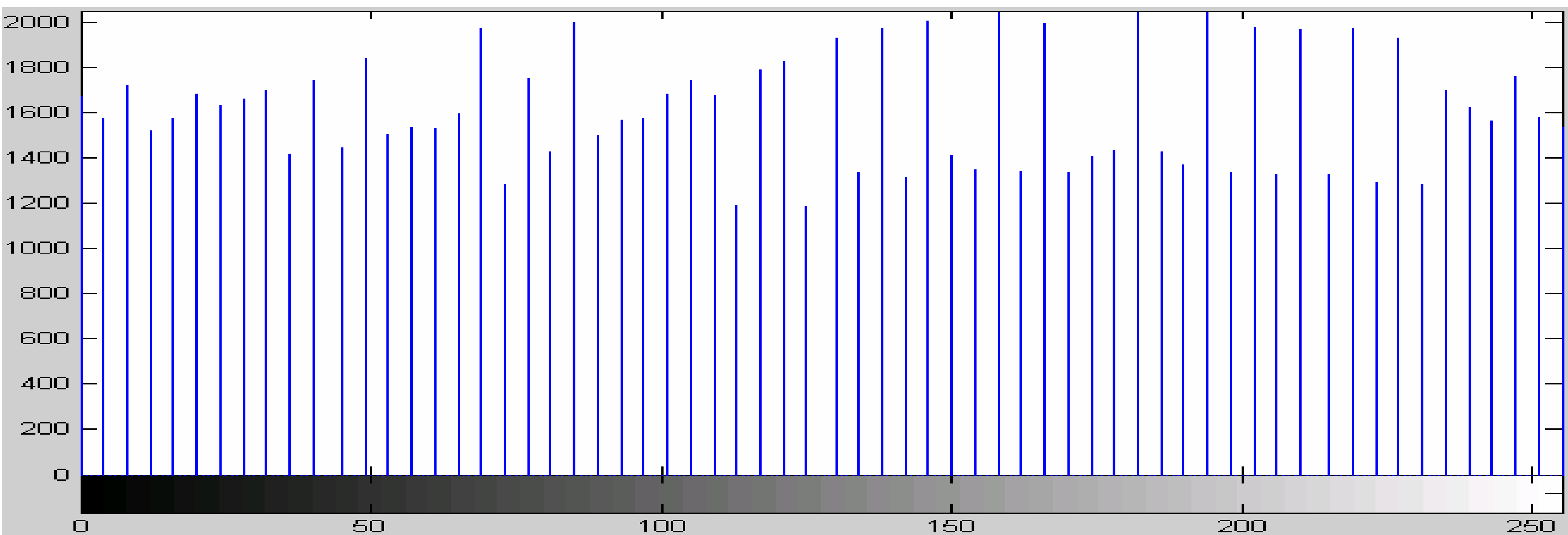
Example



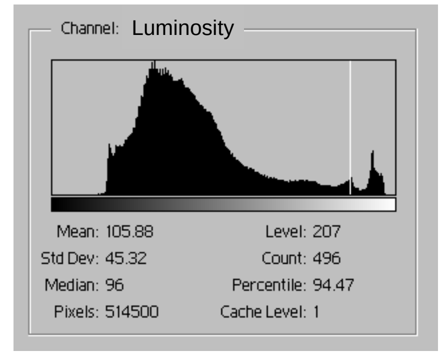
LUT

0	1
1	3
2	5
3	6
4	6
5	7
6	7
7	7

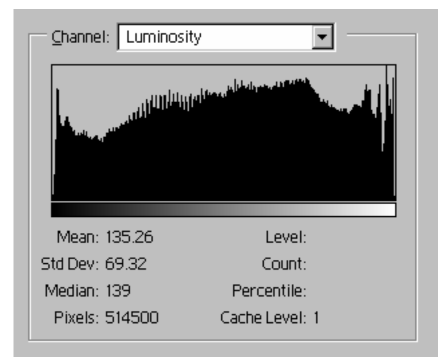

4500
4000
3500
3000
2500
2000
1500
1000
500
0
0
50
100
150
200
250


2000
1800
1600
1400
1200
1000
800
600
400
200
0
0
50
100
150
200
250

Point Processes: Histogram Equalization



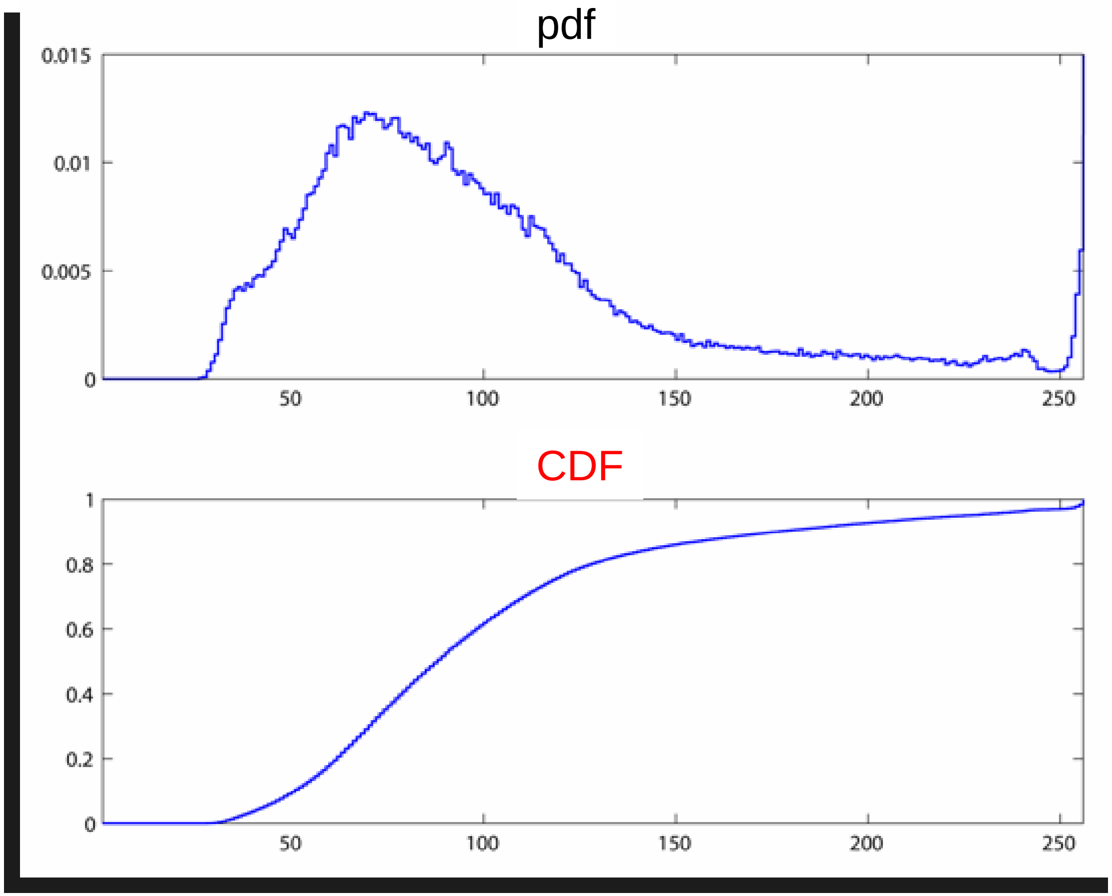
before



after

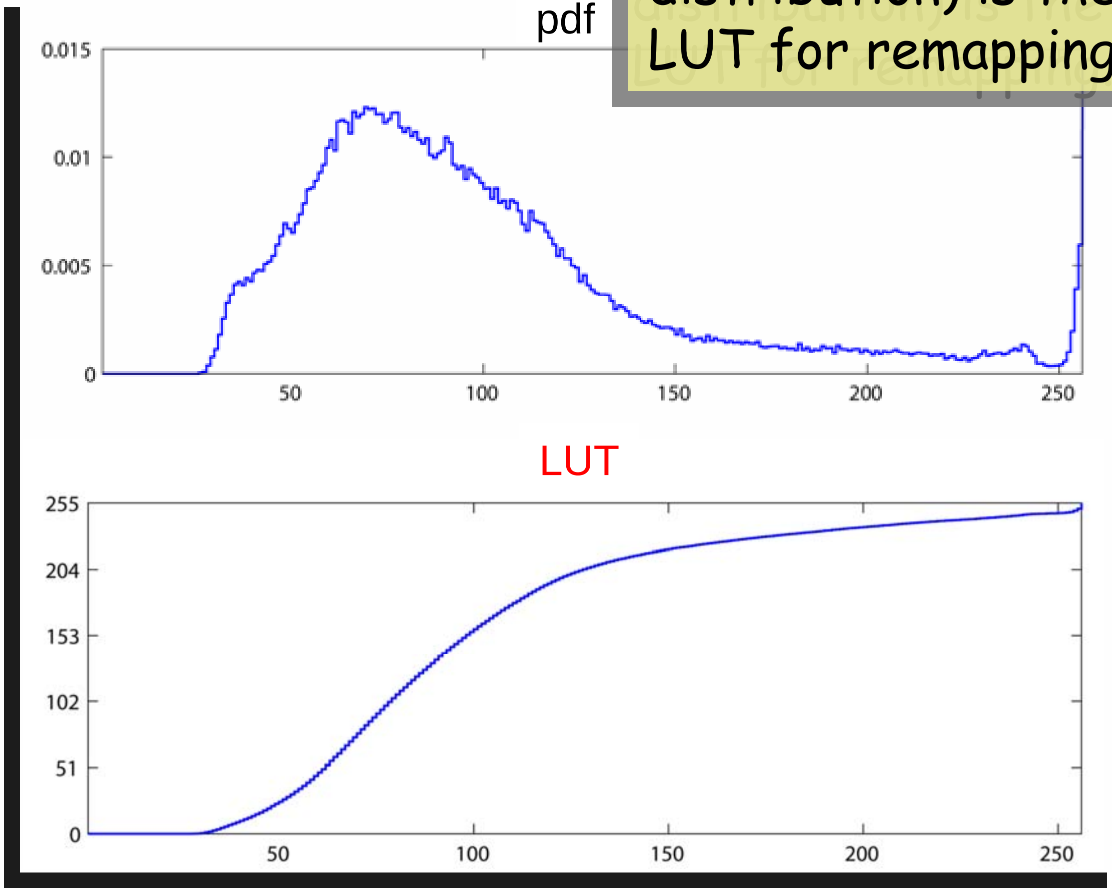
$$J(r,c) = 255 \cdot P_I(g+1)$$

Histogram Equalization



Histogram Equalization

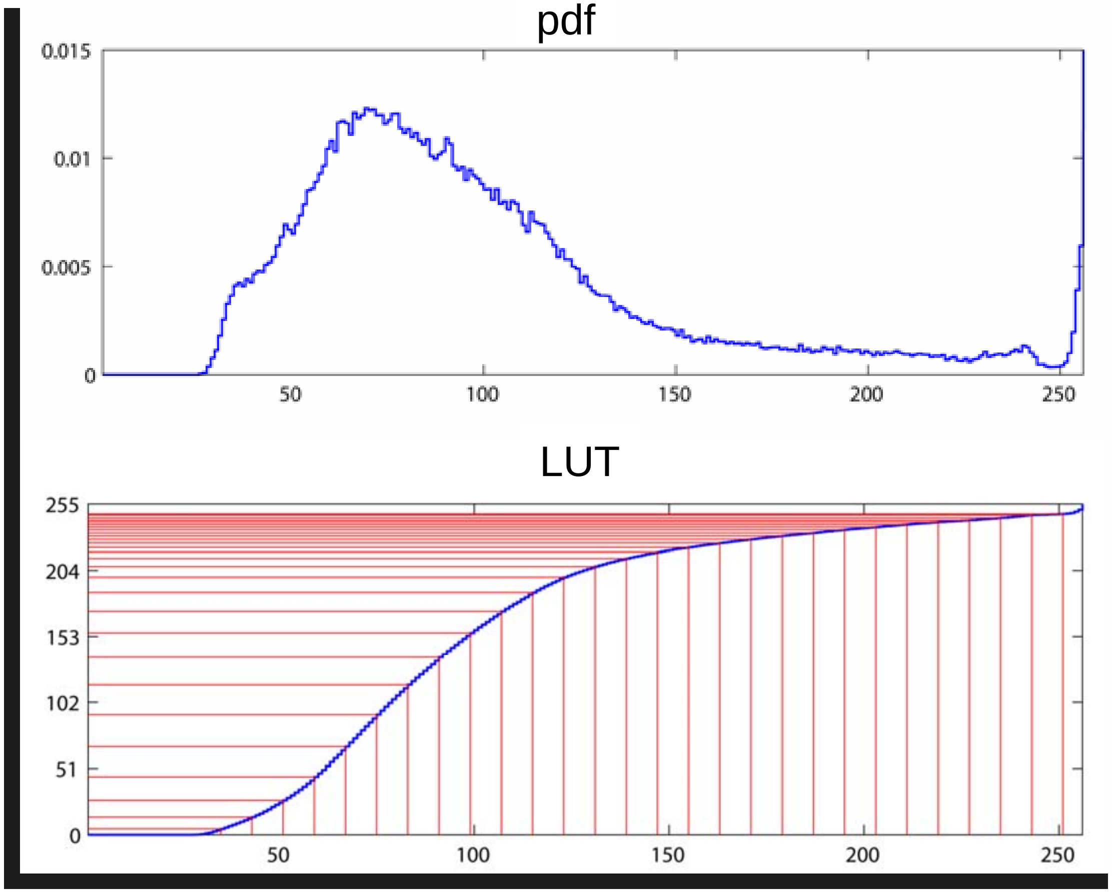
The CDF (cummulative distribution) is the LUT for remapping.



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

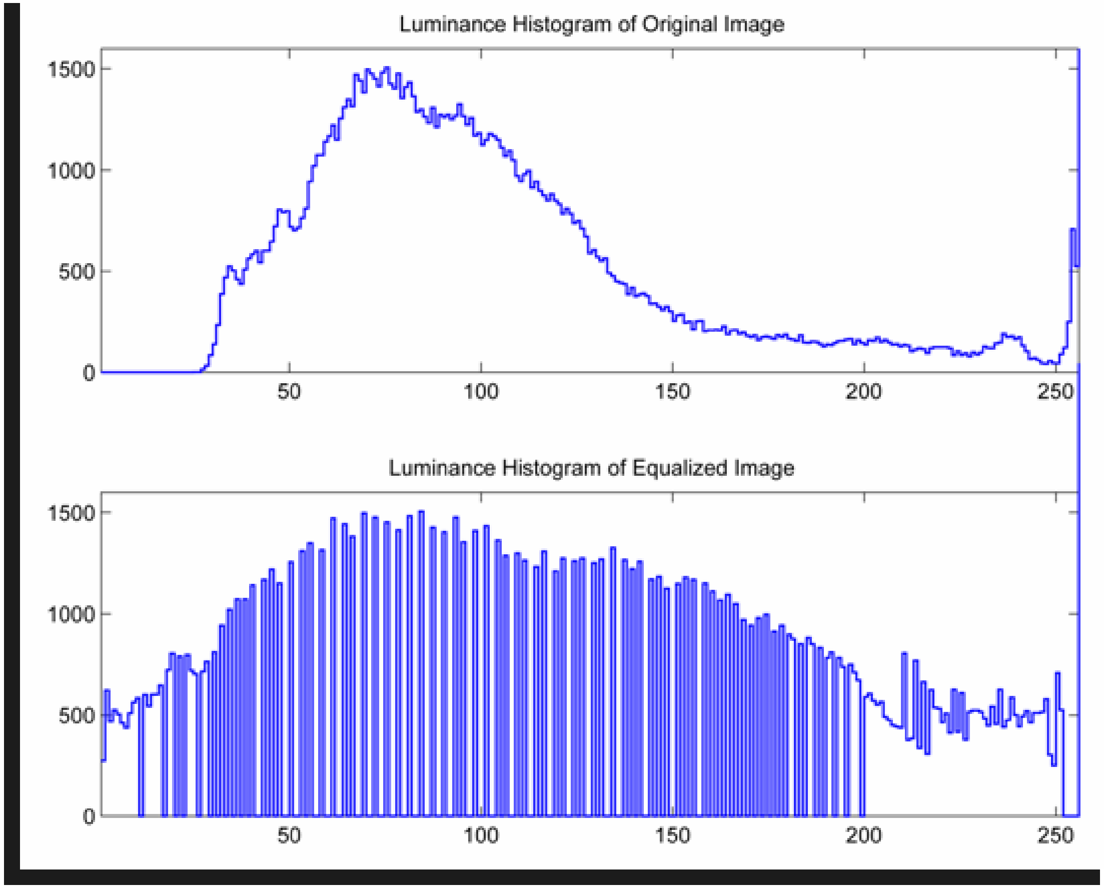
Histogram Equalization

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



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

Histogram Equalization



Challenge: Histogram Equalization on specified interval

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(直方图匹配)

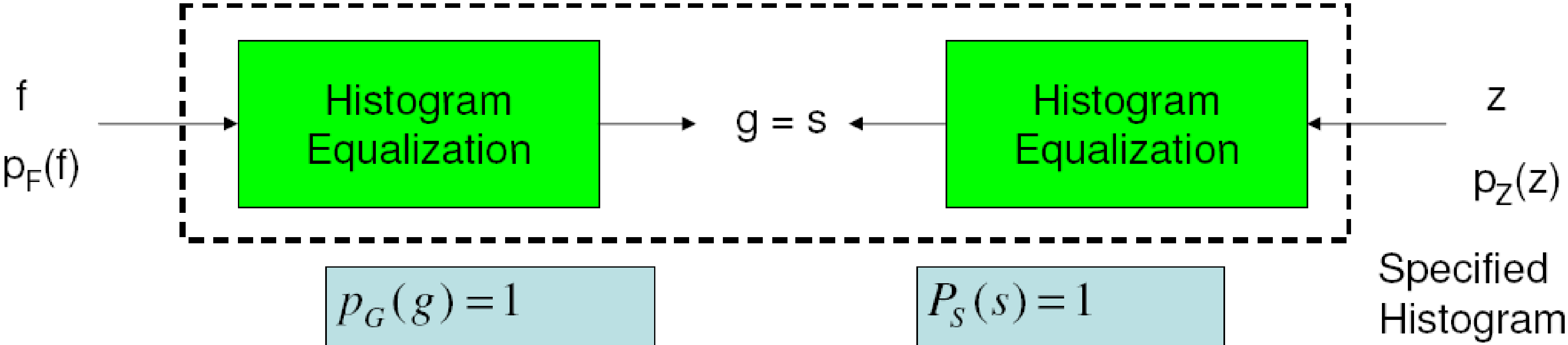
Task: remap image *I* so that it has, as closely as possible, the same histogram as image *J*.

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(百分位数).

Histogram Specification

- What if the desired histogram is not flat?



$$p_G(g) = 1$$
$$g = g(f) = \int_0^f p_F(f)\,df$$

$$P_S(s) = 1$$
$$s = s(z) = \int_0^z p_Z(z)\,dz$$

$$z = s^{-1}(g) = s^{-1}(g(f))$$

Example

f_k	$p_F(f)$	$g_k = \sum_{i=0}^{k} p_F(i)$		$s_k = \sum_{i=0}^{k} p_Z(i)$	$p_Z(k)$	z_k
0	0.19	0.19		0.0	0.0	0
1	0.25	0.44		0.0	0.0	1
2	0.21	0.65		0.0	0.0	2
3	0.16	0.81		0.15	0.15	3
4	0.08	0.89		0.35	0.20	4
5	0.06	0.95		0.65	0.30	5
6	0.03	0.98		0.85	0.20	6
7	0.02	1.00		1.00	0.15	7

f	0	1	2	3	4	5	6	7
z	3	4	5	6	6	7	7	7

z	0	1	2	3	4	5	6	7
$p_Z(z)$	0	0	0	.19	.25	.21	.24	.11

Matching Percentiles(匹配百分位数)

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

Recall:

- The CDF of image I is such that $0 \le P_I(g_I) \le 1$.
- $P_I(g_I + 1) = c$ 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

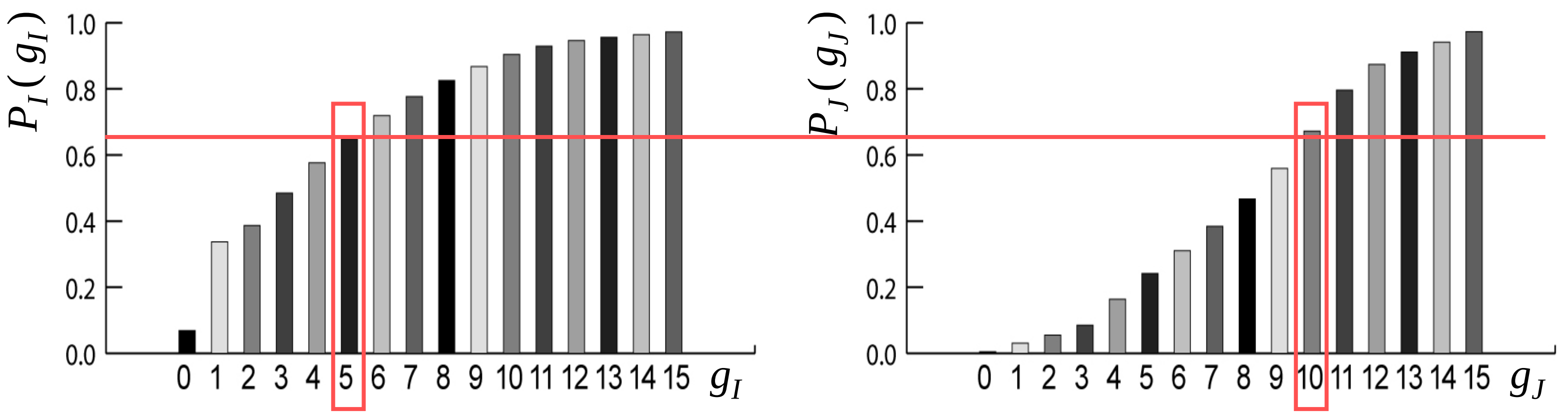
… assuming a 1-band image or a single band of a color image.

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

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

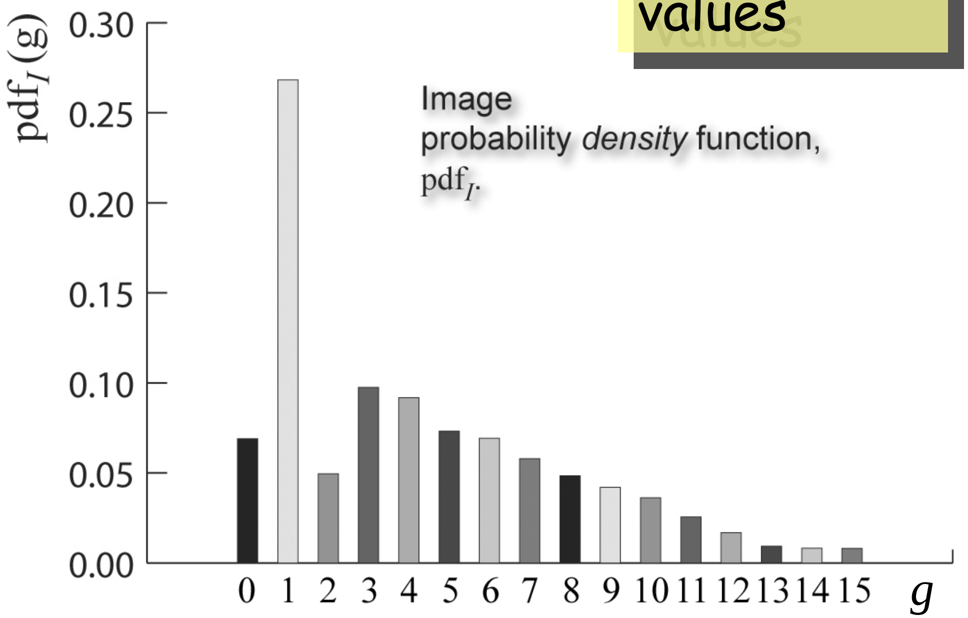
Example:
$I(r,c) = 5$
$P_I(5) = 0.65$
$P_J(9) = 0.56$
$P_J(10) = 0.67$
$K(r,c) = 10$

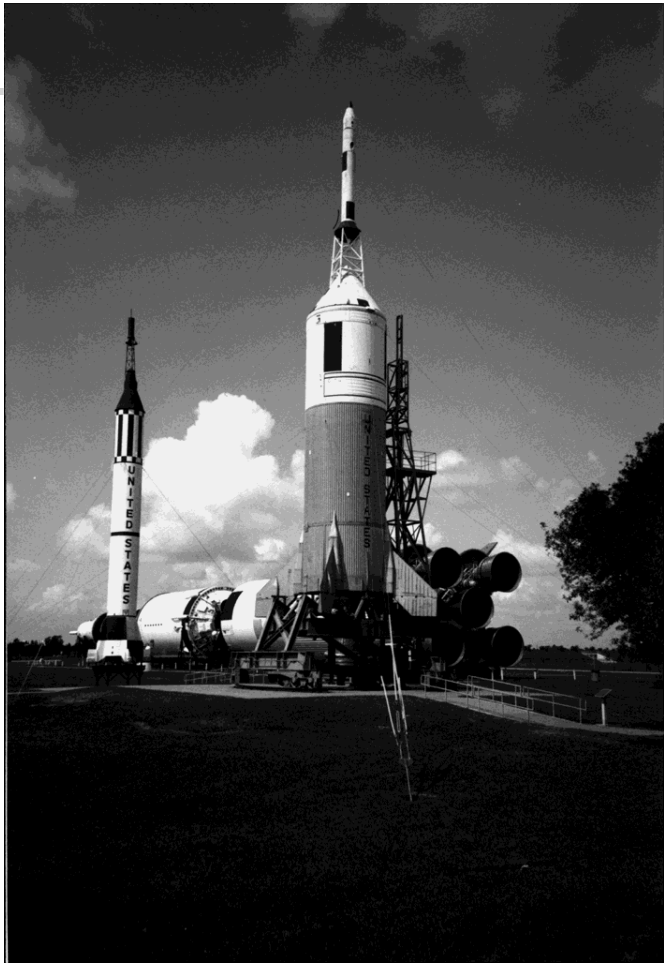


Example: Histogram Matching

Image pdf

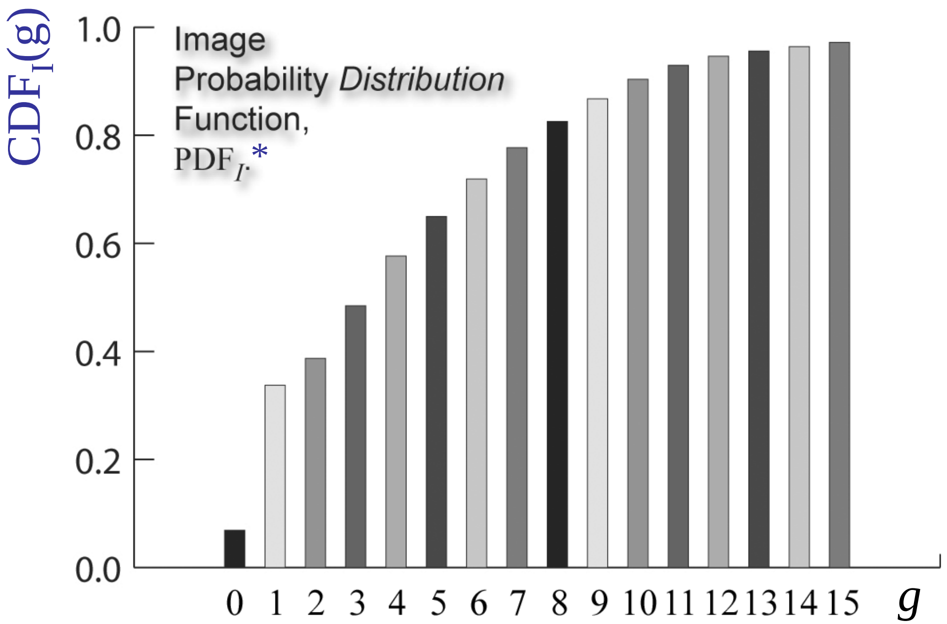
Image with 16 intensity values



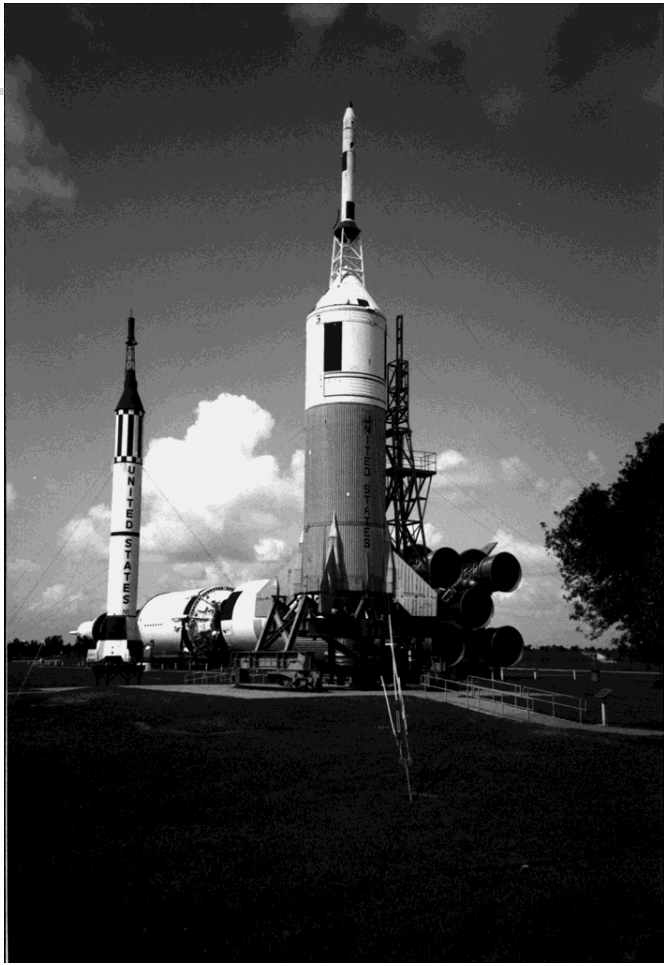


Example: Histogram Matching

Image CDF



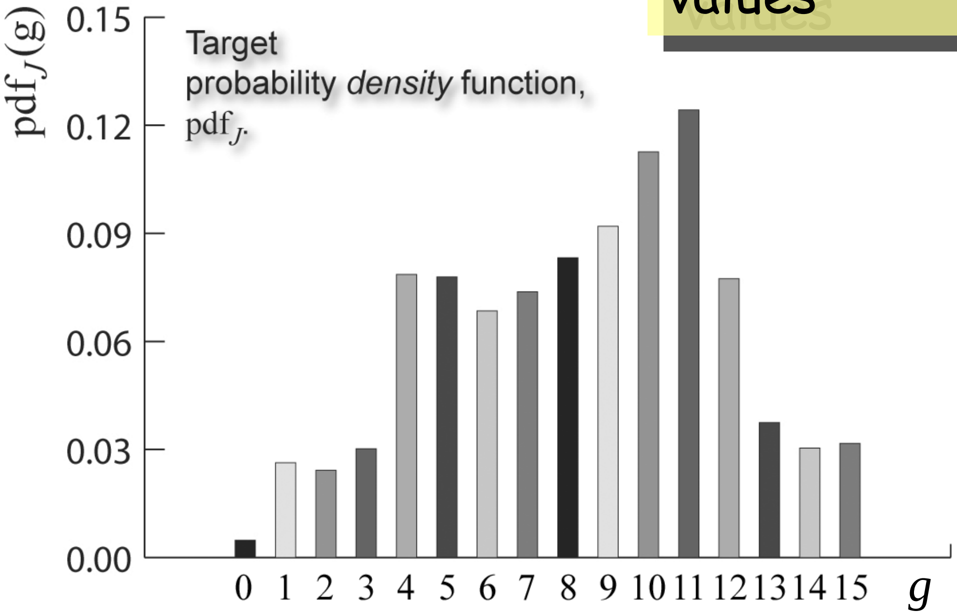
*a.k.a Cumulative Distribution Function, CDF$_I$.

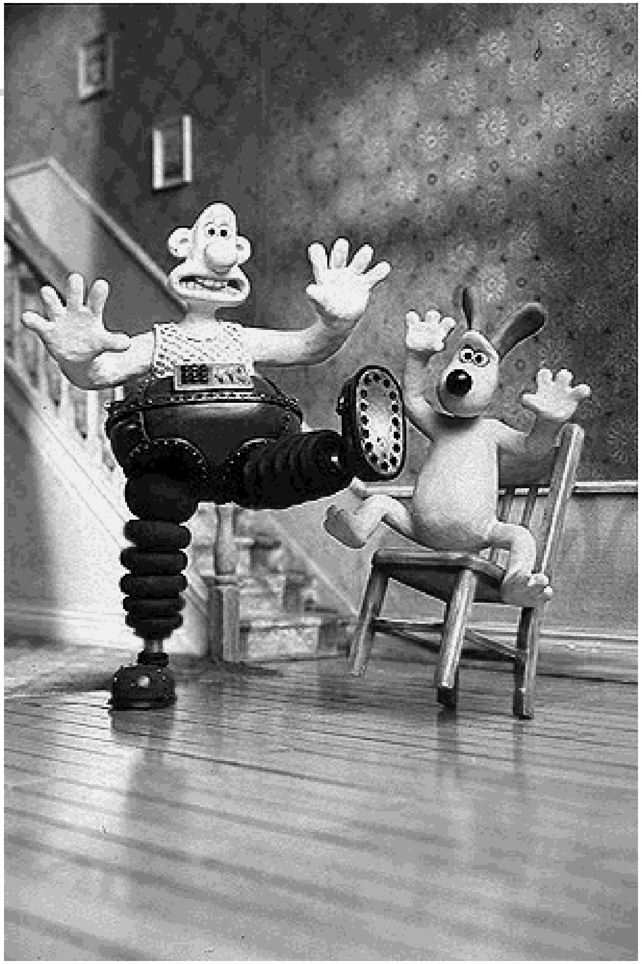


Example: Histogram Matching

Target pdf

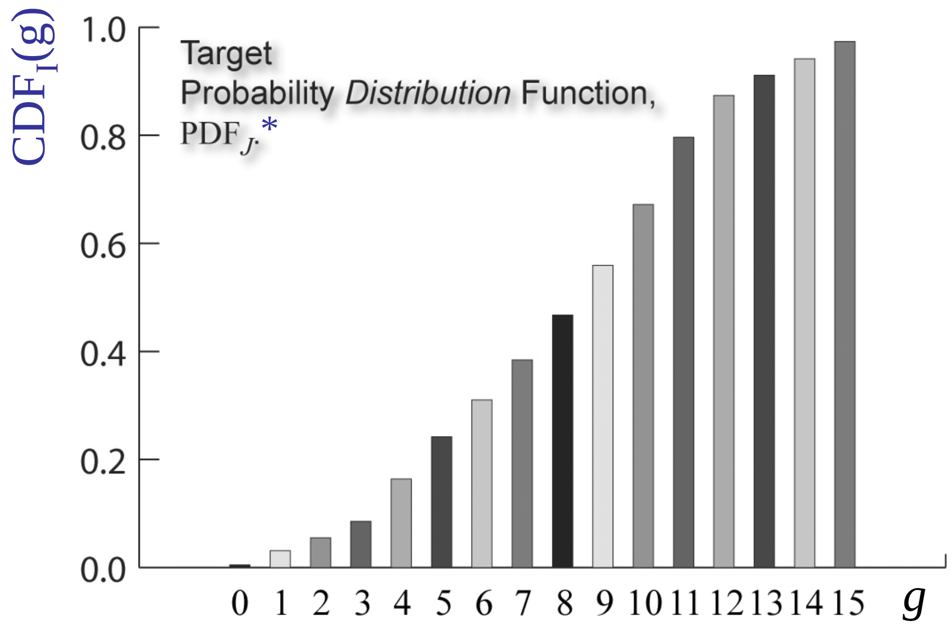
Target with 16 intensity values



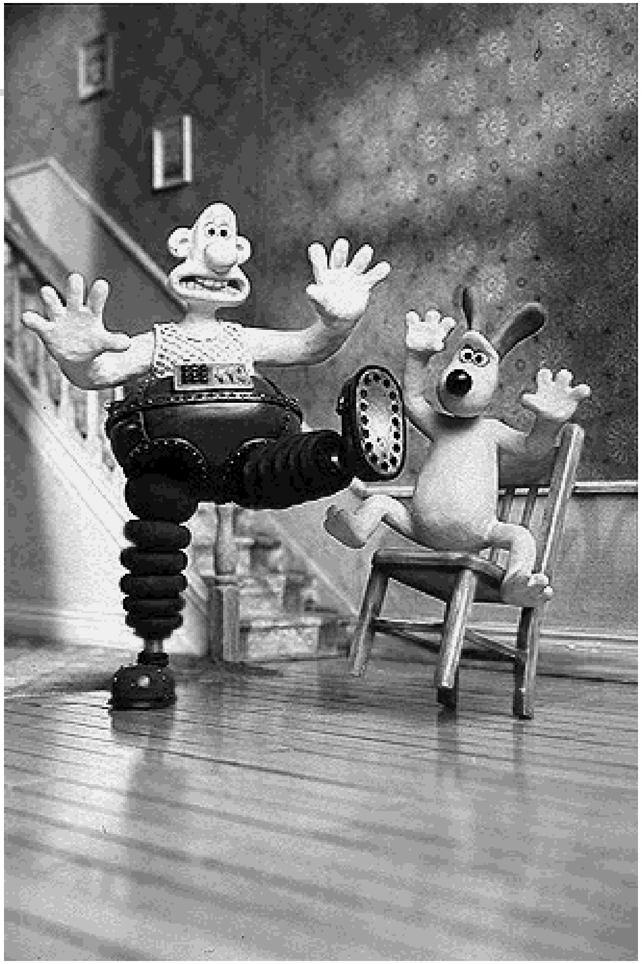


Example: Histogram Matching

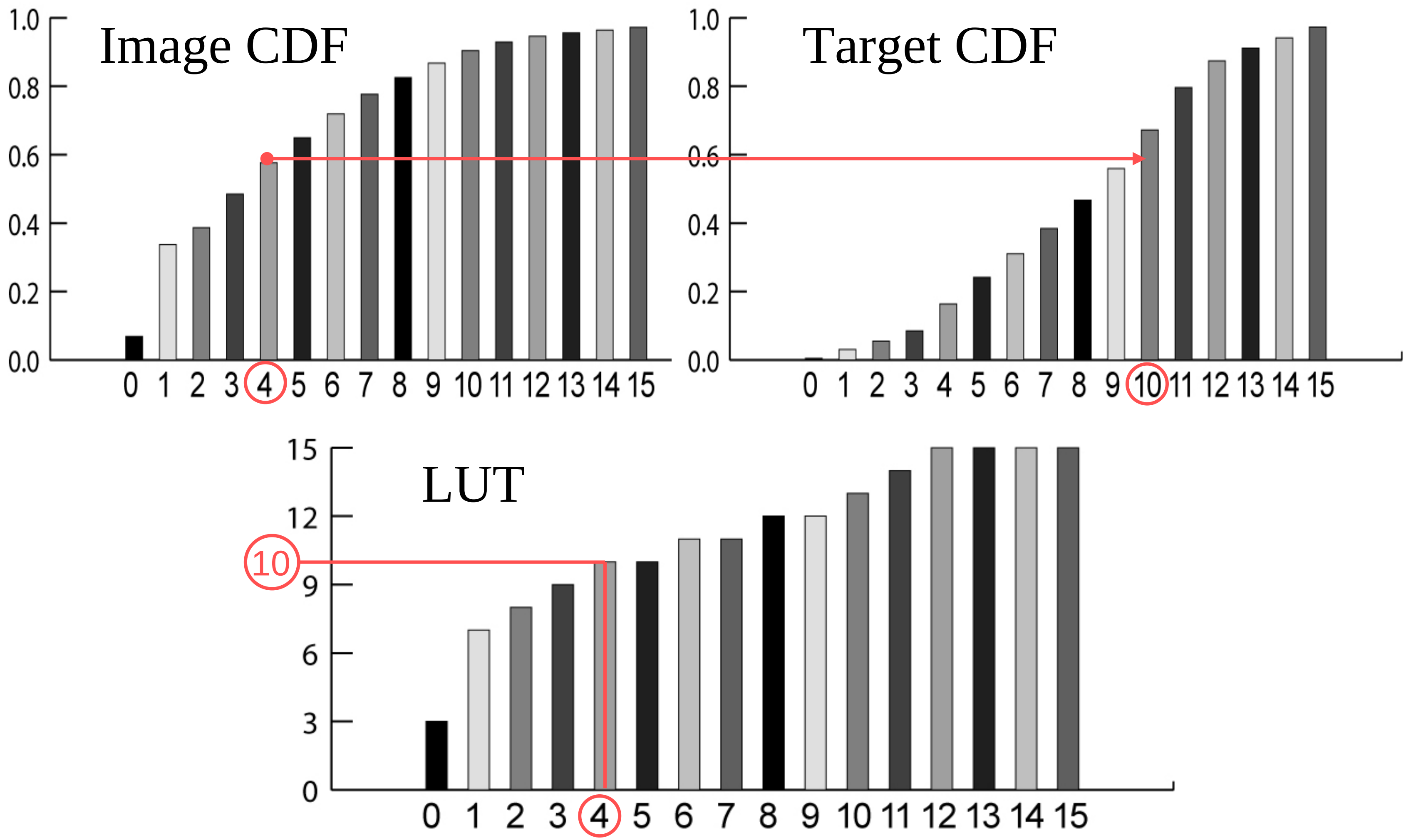
Target CDF



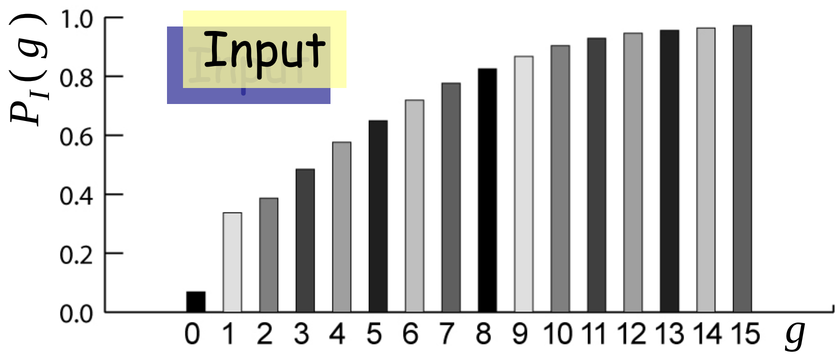
*a.k.a Cumulative Distribution Function, CDF_J.

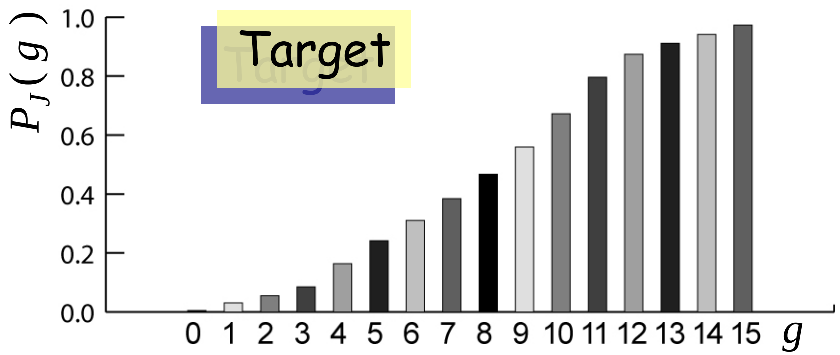


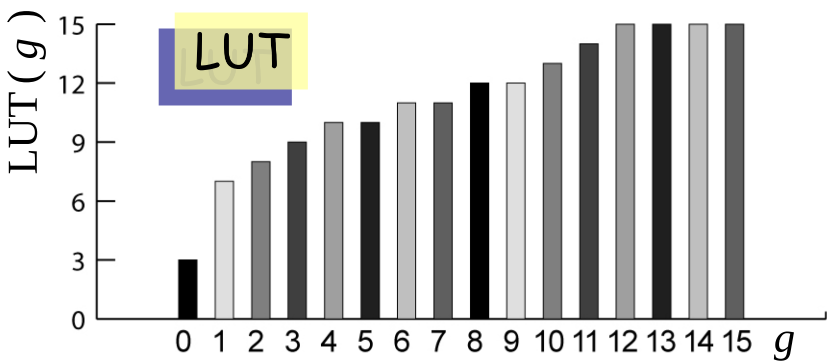
LUT Creation

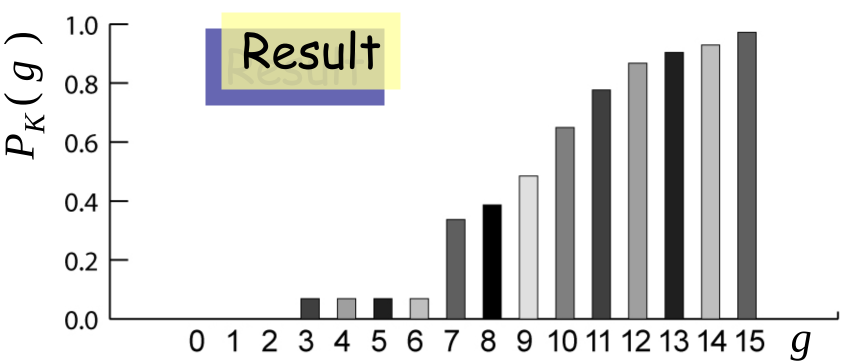


Input & Target CDFs, LUT and Resultant CDF









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.

$$
\begin{aligned}
P_I(g_I+1) &: \quad \text{CDF of } I \\
P_J(g_J+1) &: \quad \text{CDF of } J. \\
m_J &= \min J, \\
M_J &= \max J, \\
m_I &= \min I, \\
M_I &= \max I.
\end{aligned}
$$

Better to use a LUT. See next slide

Histogram Matching with a Lookup Table

The algorithm on the previous slide 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));
```

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
```

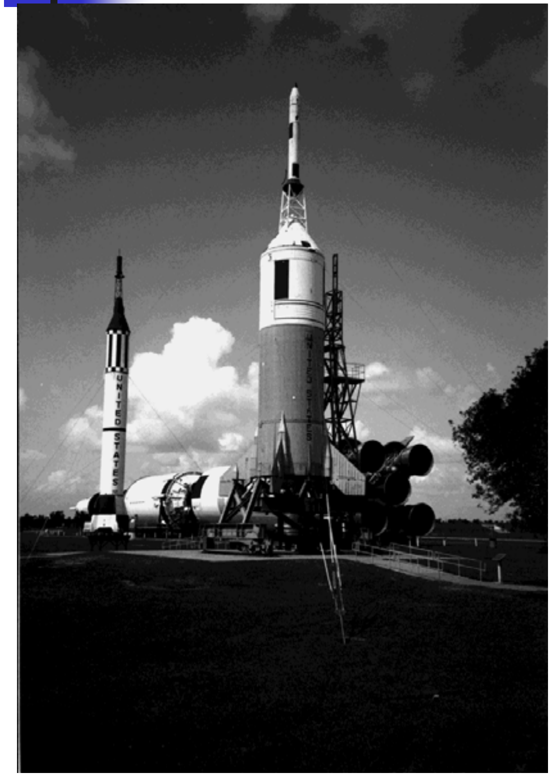
This creates a look-up table which can then be used to remap the image.

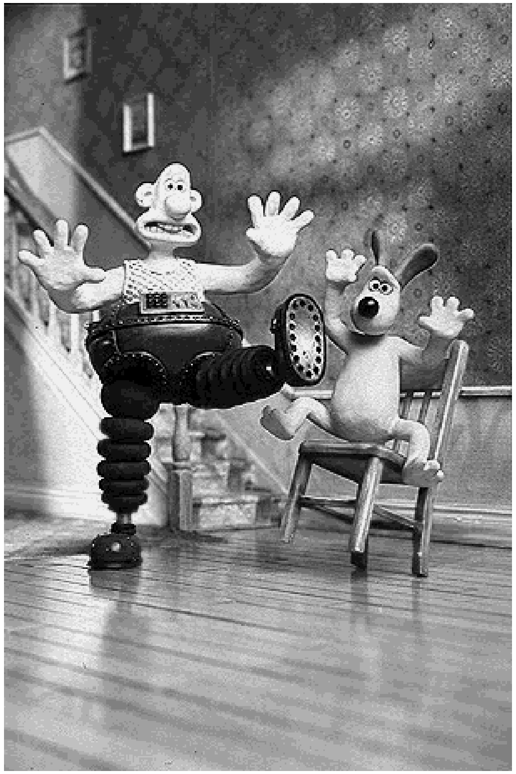
$P_I(g_I + 1)$: CDF of I ,

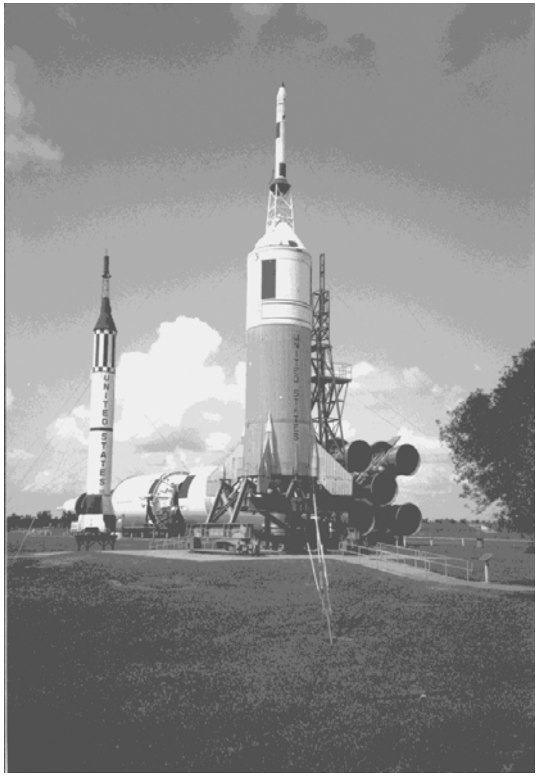
$P_J(g_J + 1)$: CDF of J,

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

Example: Histogram Matching


original


target


remapped

Is there any relation between the LUT and the source/target image CDF?

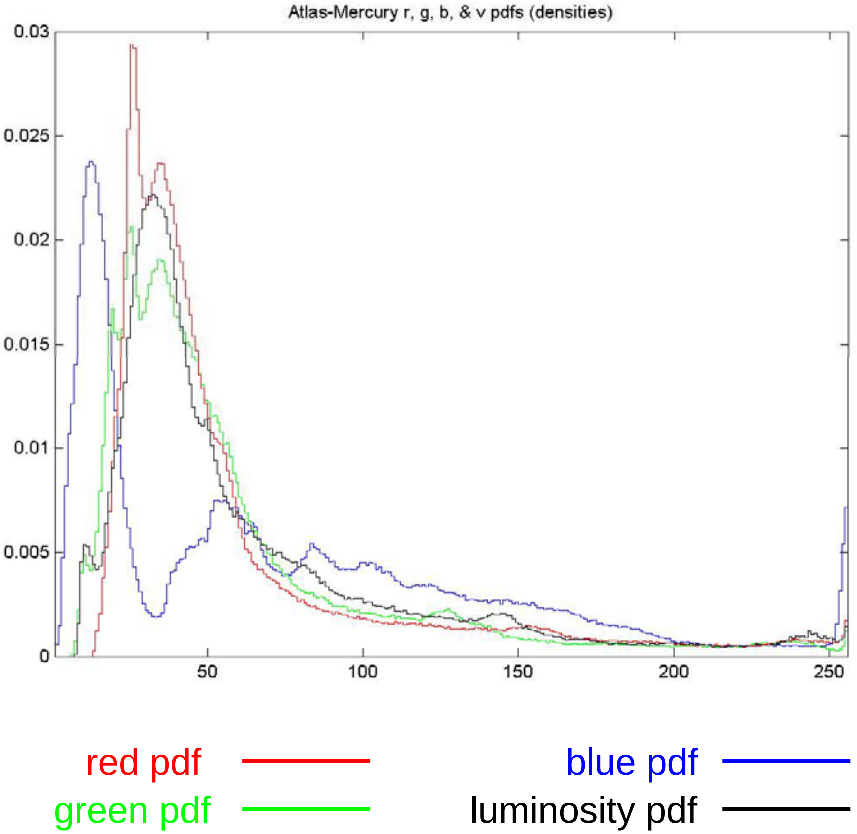
Break

- Get confused by more and more terms, abbreviations and acronyms ?
  - histogram, channel, band, luminance, binear, bicubic, percentiles;
  - Interpolation, decimation, matching, equalization, normalization;
  - LUT, cdf, pdf, vs., a.k.a.;
  - ……

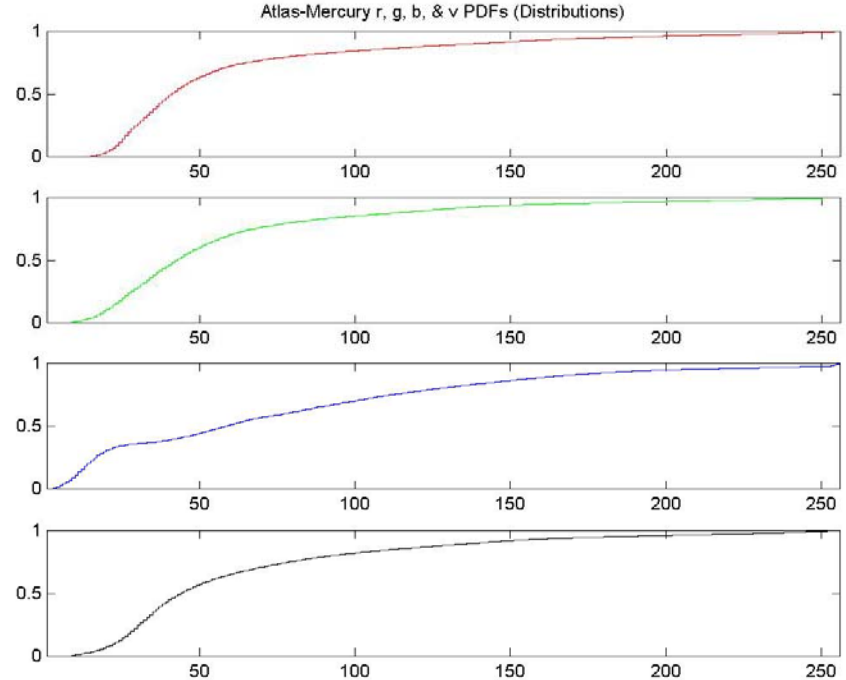
Homework III

- 1. Convert the following image to a visually meaningful one by point processing. Submit your code and the result image.
- 2. Realize your own Histogram equalization or matching algorithm. Submit your code and demo images. Compare your result with matlab function histeq().

Probability Density Functions of a Color Image

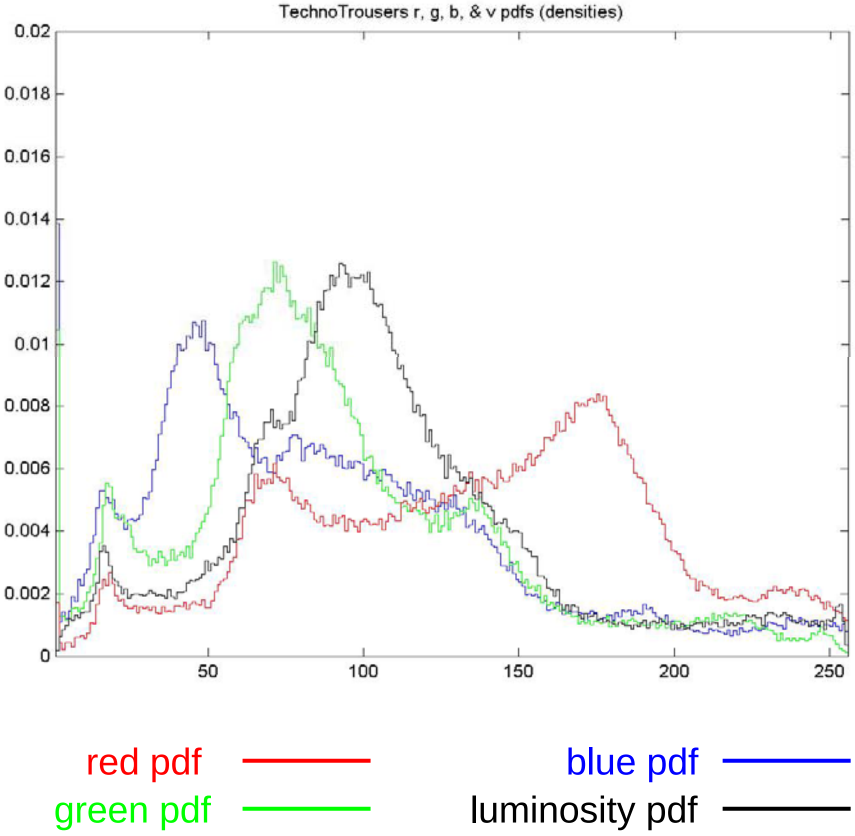


Cumulative Distribution Functions (CDF)

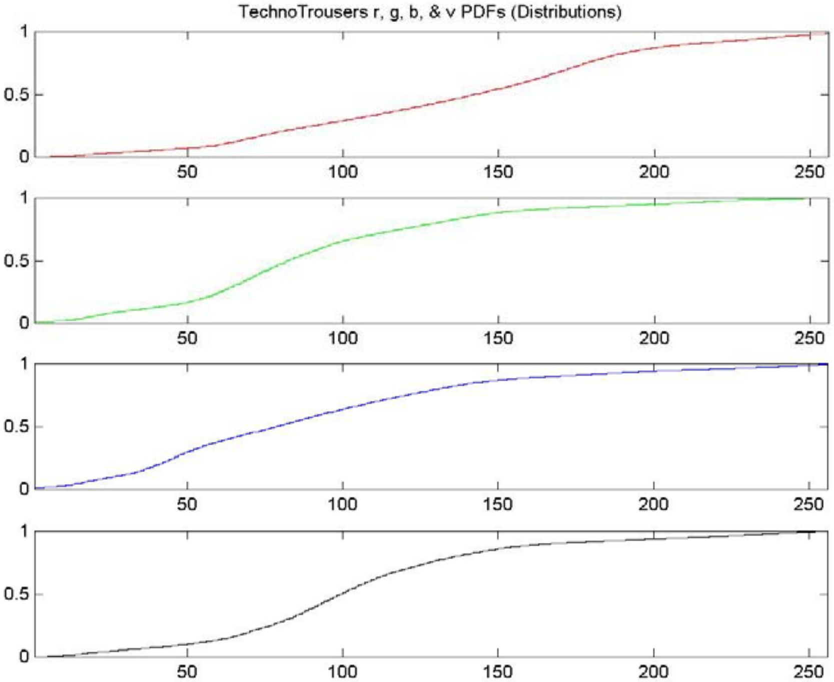


red CDF
green CDF
blue CDF
luminosity CDF

Probability Density Functions of a Color Image



Cumulative Distribution Functions (CDF)



red CDF
green CDF
blue CDF
luminosity CDF

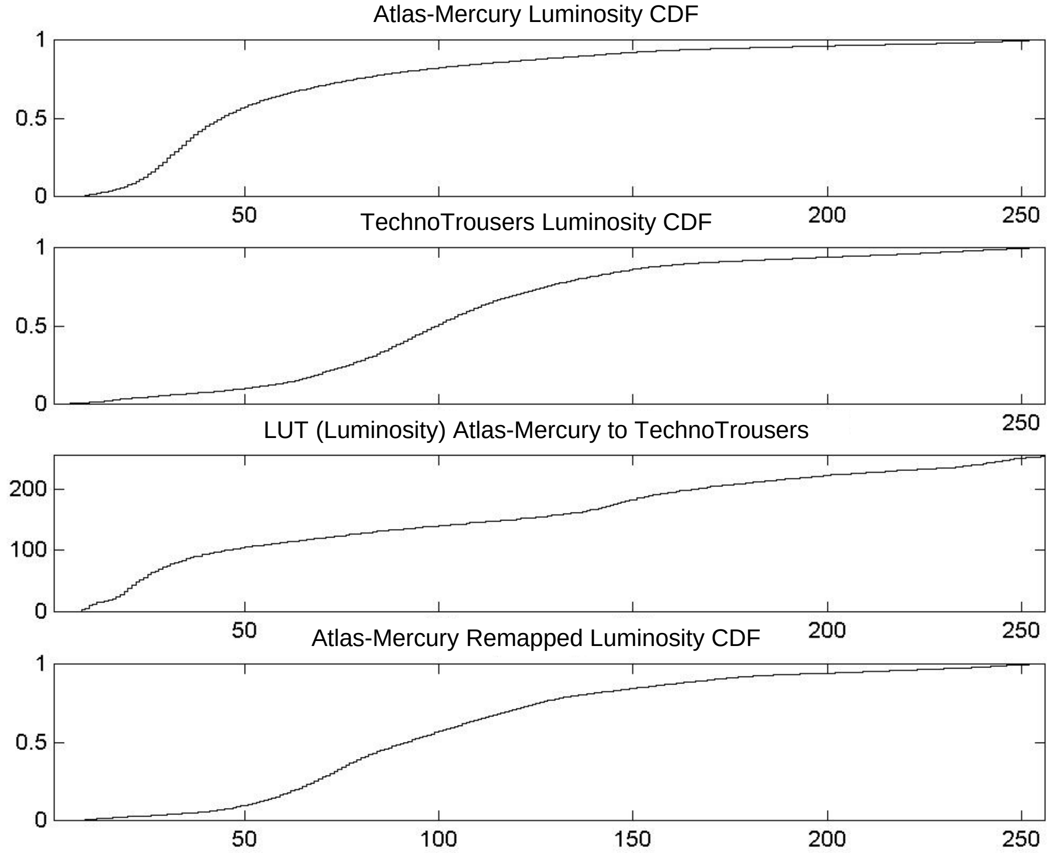
Remap an Image to have the Luminance CDF of Another

original

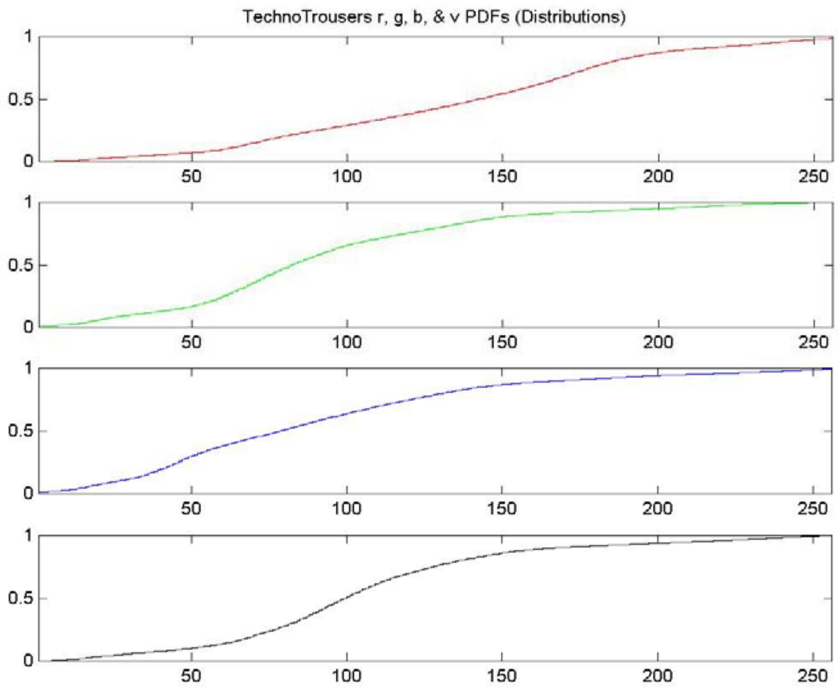
target

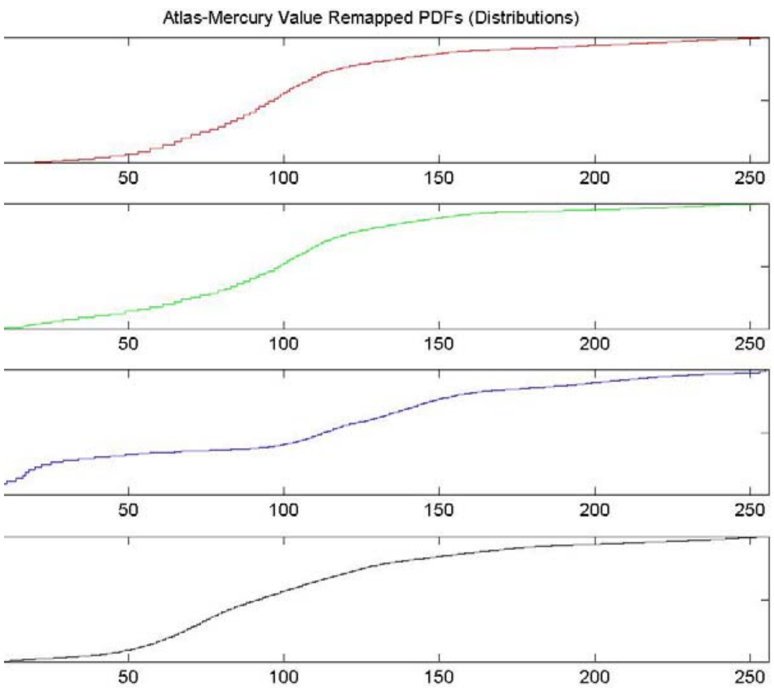
luminosity remapped

CDFs and the LUT



Effects of Luminance Remapping on CDFs





Before

After

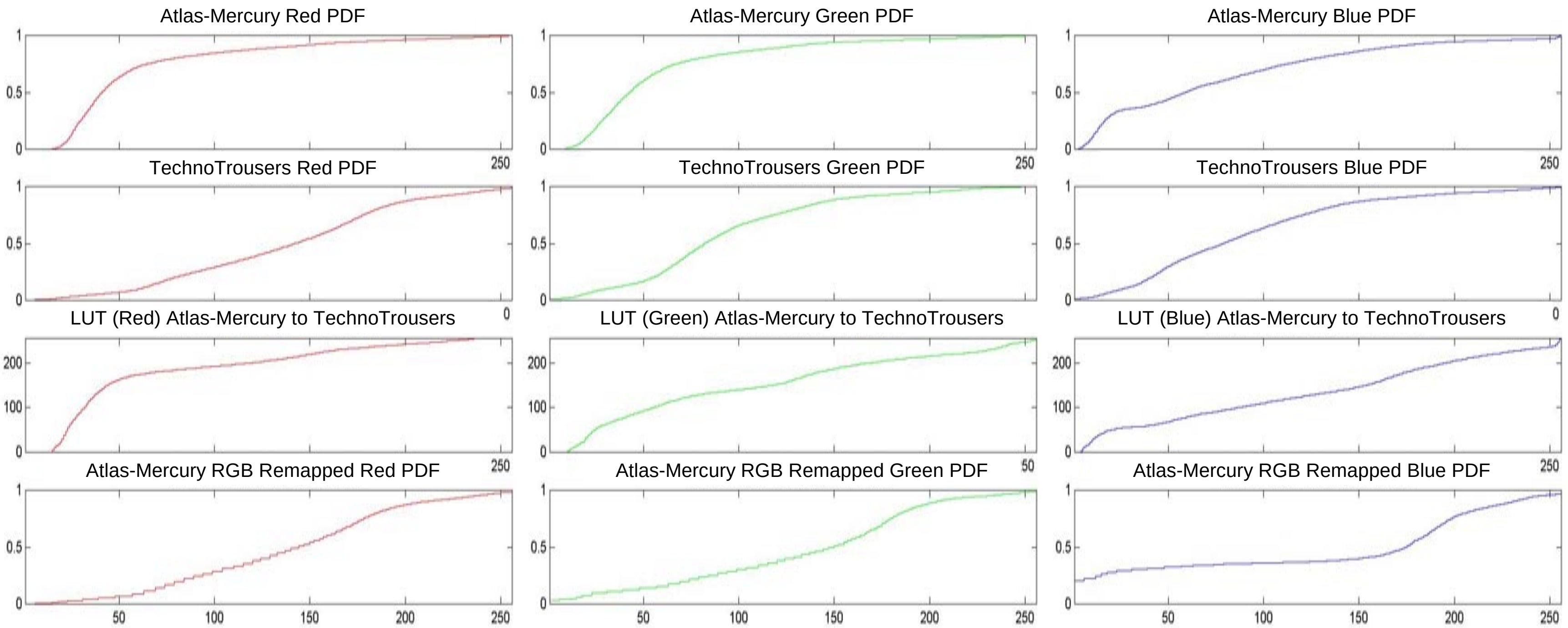
Remap an Image to have the R/G/B CDFs of Another

original

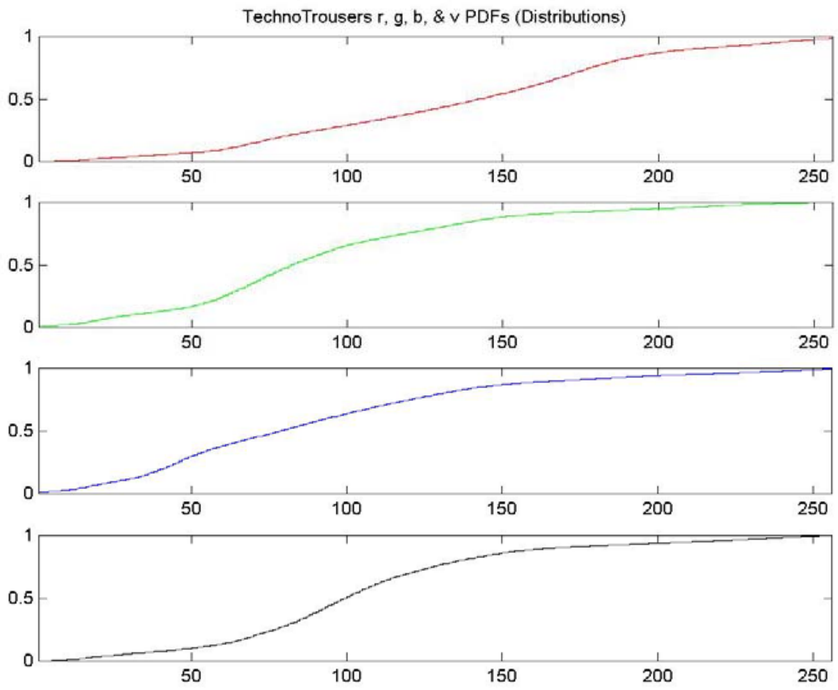
target

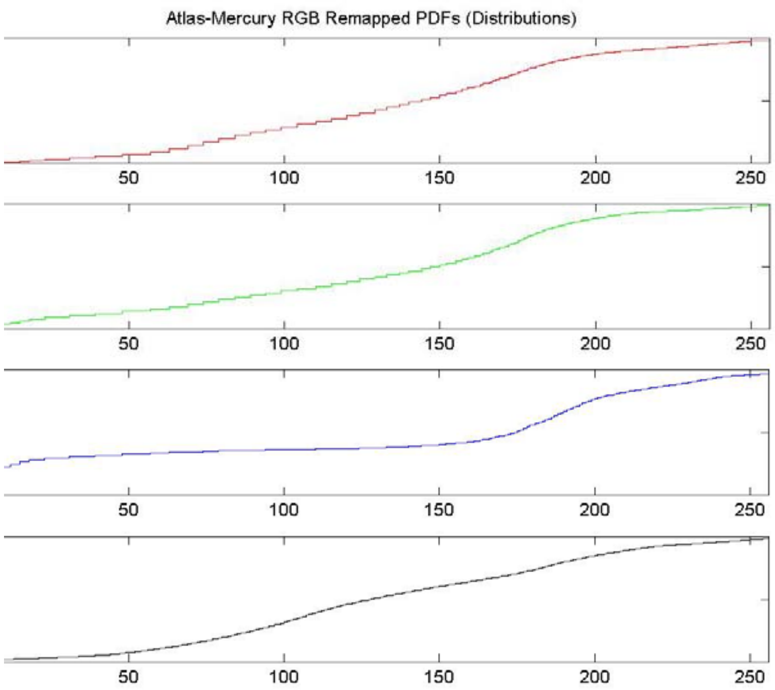
R, G, & B remapped

CDFs and the LUTs



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

Challenge:

Display of HDRI

报告人：何小伟

什么是HDR

- **Dynamic Range**（动态范围）：是指图像中所包含的从“最亮”至“最暗”的比值，也就是图像从“最亮”到“最暗”之间灰度划分的等级数；动态范围越大， 所能表示的层次越丰富，所包含的色彩空间也越广。

- 高动态范围**(High Dynamic Range)**顾名思义就是从“最亮”到“最暗”可以达到非常高的比 值。

什么是HDR

8bit的动态范围

100：1

人眼所能感知的动态范围

1000 000 000:1

用32bit float来表示

理论上动态范围可到10^76.8：1

什么是HDR

HDR

- 亮的地方可以非常亮
- 暗的地方可以非常暗
- 亮暗部的细节都很明显

HDRI

OpenEXR

• Industrial Light & Magic

•扩展名为（.exr）
• FP16

• 6.14 × 10 ^ -5到 6.41 × 10 ^ 4

Radiance RGBE

• 扩展名为（.hdr）

•4个通道，每个通道8bit

Float TIFF

•扩展名为（.tif）
•
•3个通道每个通道为FP32

Radiance RGBE

- **rgb为颜色通道，e为亮度级别，每个通道8bit**

R = r * 2^(e – 128 - 8)
G = g * 2^(e – 128 - 8)
B = b * 2^(e – 128 - 8)


webpark.ru

Tone mapping

- 动态范围的压缩，压缩后具有较好的全局对比度；
- 压缩后的图像具有真实感，避免光晕(halo)等痕迹，较好地保持局部细节。

Tone mapping

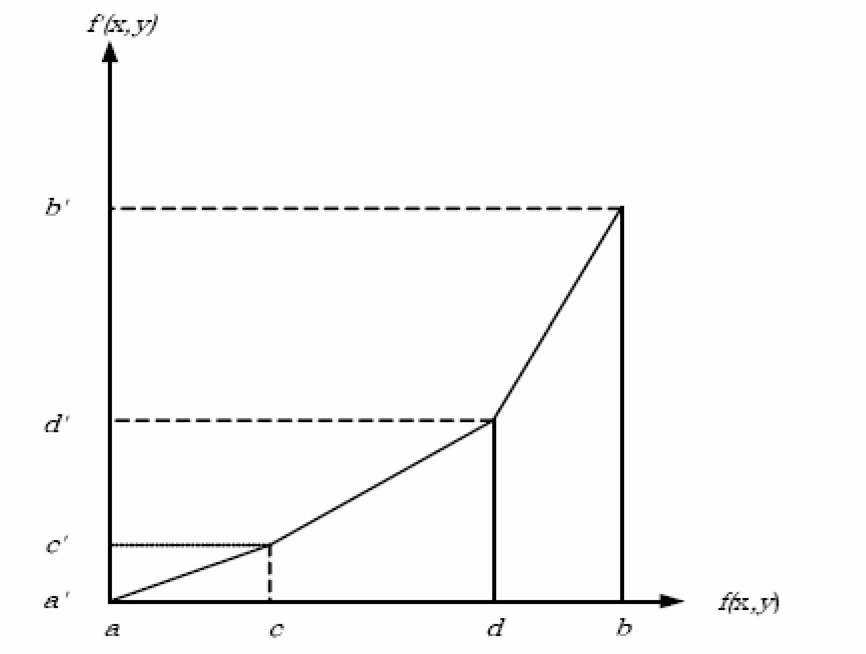
❖ 全局映射

$$L_{final} = \frac{L-\alpha}{\beta-\alpha}$$

$$L_{final} = \frac{L}{1+L}$$

❖ 局部映射

- 分段线性映射
- 双边滤波



Tone mapping

- 存在信息丢失
- 关键：保留主要特征区域、如边缘信息，而去除噪声。