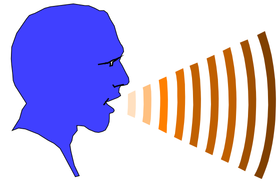
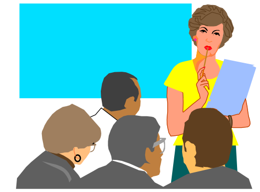
   

Image and Video Processing

Contrast Enhancement

Yao Wang
Tandon School of Engineering, New York University

Image Contrast Enhancement

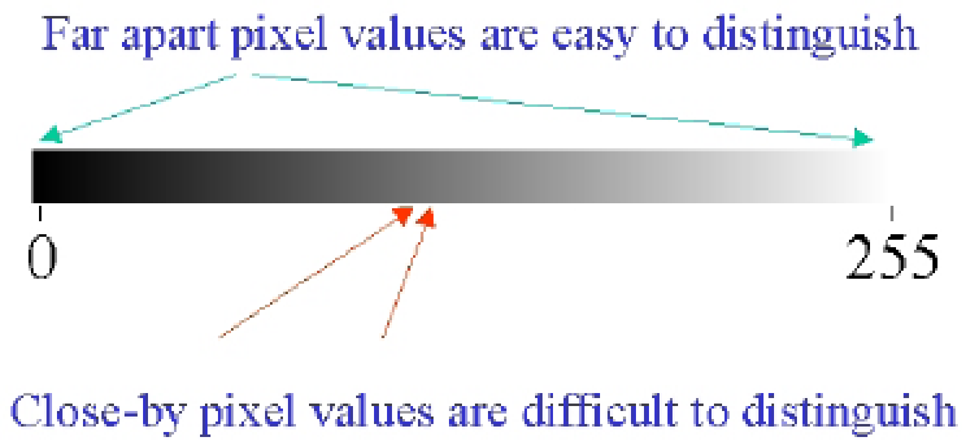
- Introduction
- Linear stretching
- Nonlinear stretching
- Histogram equalization
- Histogram specification
- Adaptive histogram modification

What is Contrast Enhancement

Original image with low contrast

Enhanced image

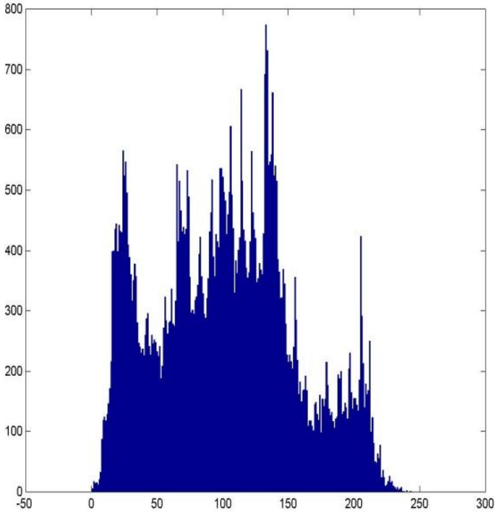
How to enhance the contrast?



- Low contrast → image values concentrated near a narrow range (mostly dark, or mostly bright, or mostly medium values)
- Contrast enhancement → change the image value distribution to cover a wide range
- Contrast of an image can be revealed by its histogram

Histogram

- Histogram of a monochrome image with L possible gray levels, l = 0, 1, …, L-1.
  - $P(l) = n_l / n$,
    - n_l is the number of pixels with gray level l.
    - n is the total number of pixels in the image.



Histogram vs. Image Contrast

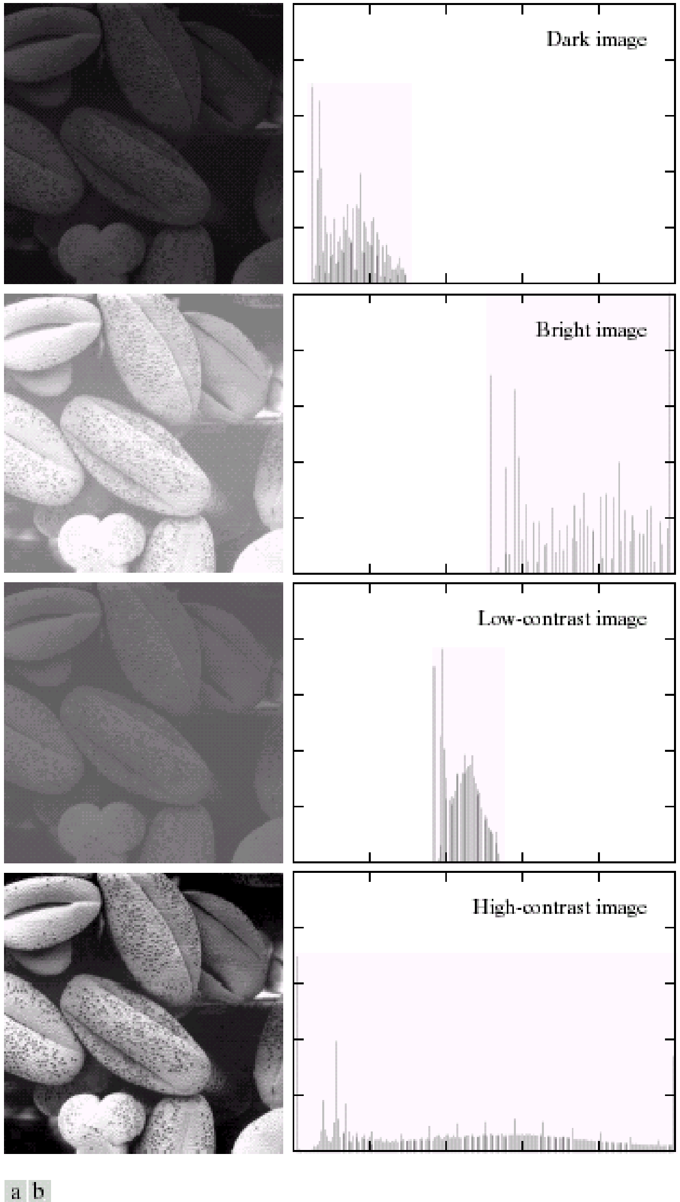


FIGURE 3.15 Four basic image types: dark, light, low contrast, high contrast, and their corresponding histograms. (Original image courtesy of Dr. Roger Heady, Research School of Biological Sciences, Australian National University, Canberra, Australia.)

Images with figure captions in this and other slides are from [Gonzalez02]

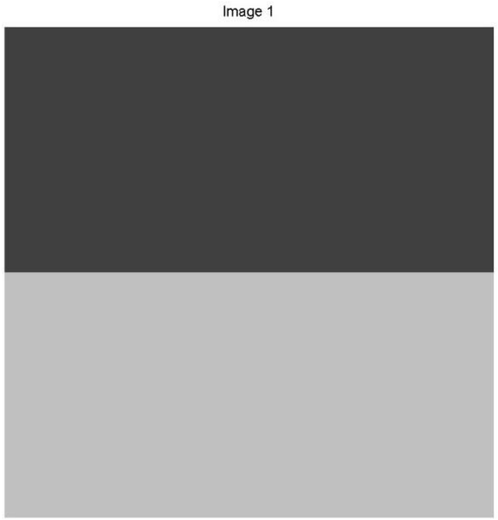
Histogram Calculation

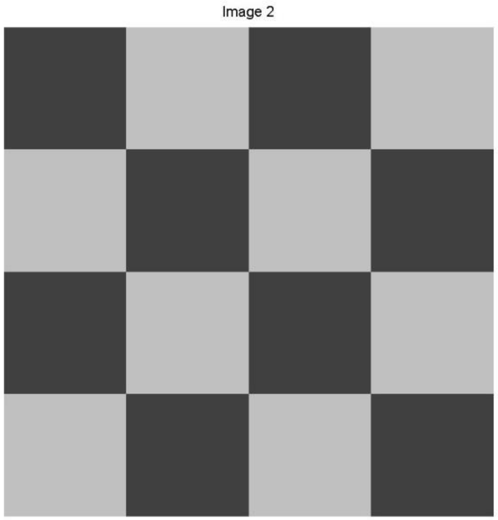
```
function h = histogram(imgname)
img = imread(imgname);
figure; imshow(img);

% method 1
h = zeros(256,1);
for l = 0:255
   for i = 1:N,
      for j = 1:M,
         if img(i, j) == l,
            h(l + 1) = h(l + 1) + 1;
         end
      end
   end
end
figure; bar(h);
```

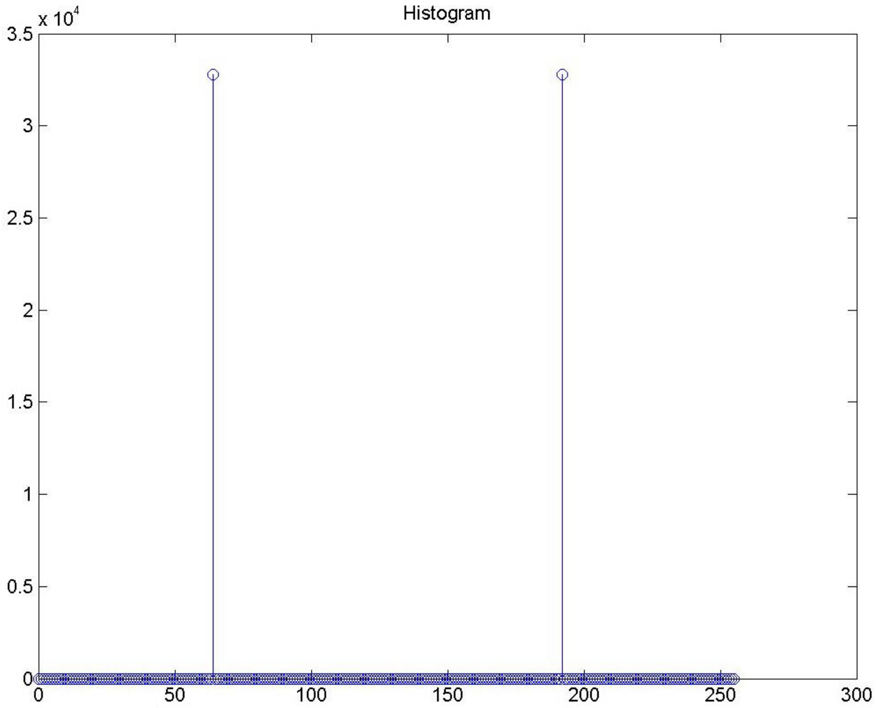
```
% method 2
img = double(img);
h = zeros(256,1);
for i=1:M,
   for j=1:N,
      f = img(i,j);
      h(f+1) = h(f+1) + 1;
   end
end

% method 3
h = zeros(256,1);
for l = 0 : 255,
  h(l + 1)=sum(sum(img == l));
end
```

Photoshop has extensive histogram display tools
Matlab: imhist( ): can compute and display histograms
Python: matplotlib.pyplot.hist( ), numpy.histogram( )

Very Different Images May Have Same Histogram!







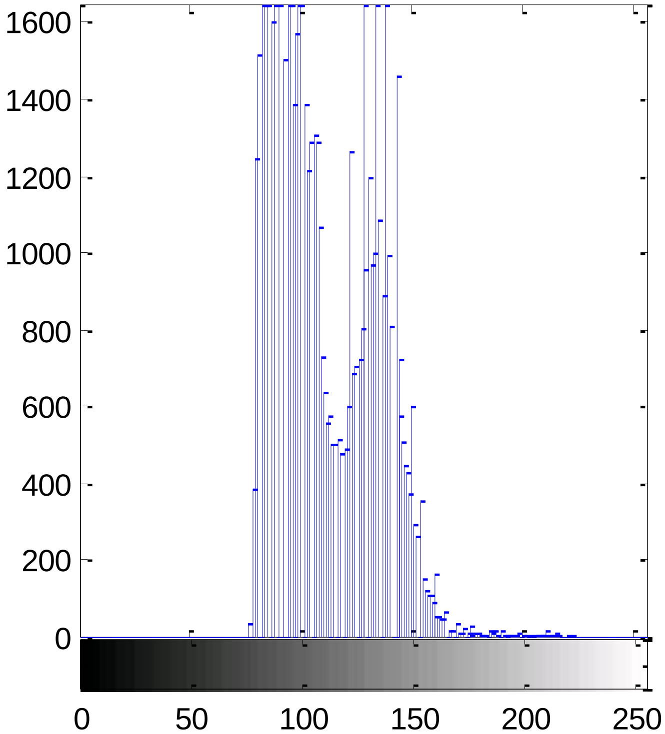
Histogram reflects the pixel intensity distribution, not the spatial distribution!

Previous Example

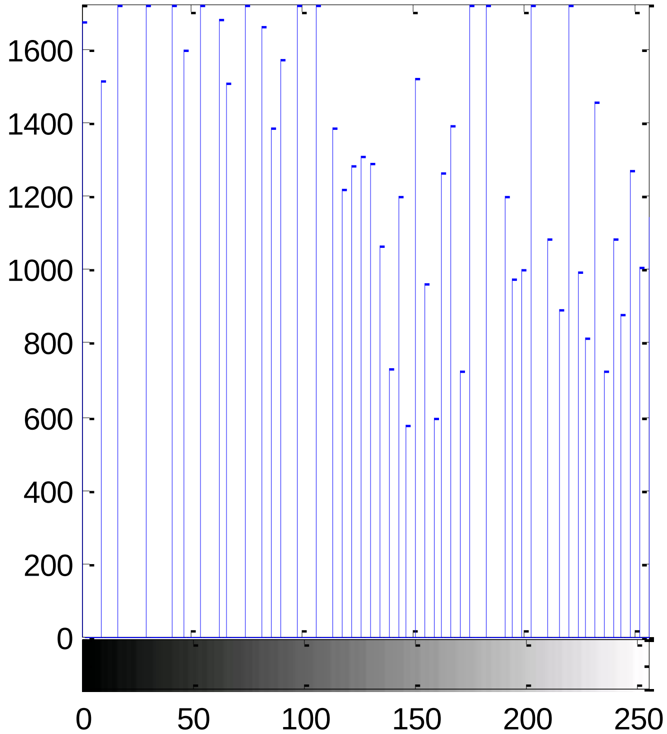
Original image with low contrast

Enhanced image

Histograms of Example Images



Original girl image with low contrast



Enhancement image with histogram equalization

How to change the histogram? Using Point-Wise Transformation

- Use a "function" g(f) to generate a new image B from a given image A via:

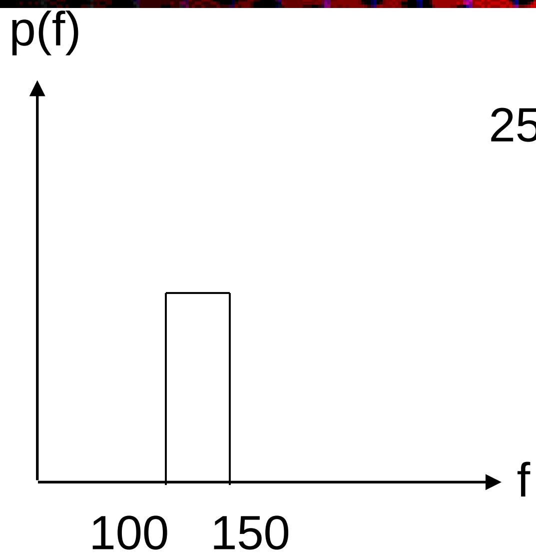
$$B(i,j) = g(A(i,j)), \quad i = 0,\ldots,N-1,\ j = 0,\ldots,M-1$$

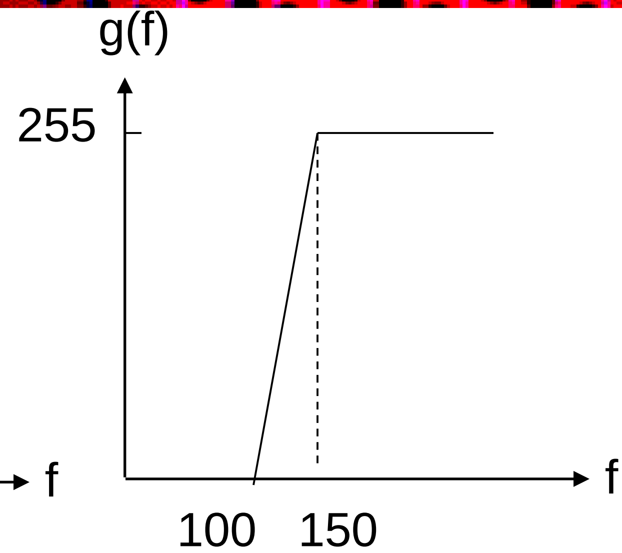
- The function *g*(*f*) operates on each image pixel independently. All pixels with original gray level f are changed to have gray level g(f)
- Properties that g(f) should satisfy
  - Monotonically non-decreasing, so that relative brightness of pixels do not change.
  - G(f) in the same range as original f, i.e. with same min (e.g. 0) and max values (e.g. 255), and be integers for digital images.
    - Rounding/truncation may be needed

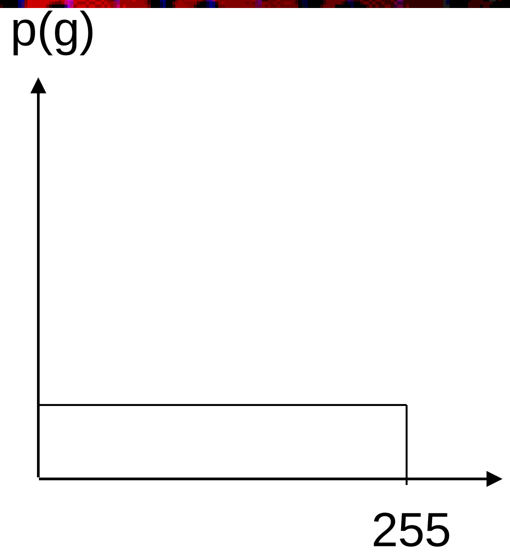
How to Determine the Transformation Function?

- How to design the transformation function g(f)?
  - depends on the histogram of the original image h_A(f) and the desired histogram of the transformed image h_B(f).
  - To enhance contrast, we like h_B(f) to be as flat as possible.
- Different approaches
  - Using fixed functional forms: linear, non-linear
  - Using adaptive transform, that is determined from h_A(f) and h_B(f):
    - Histogram equalitzation (h_B(f) is uniform): Fully automatic!
    - Histogram specification or matching

Illustration of Linear Stretching

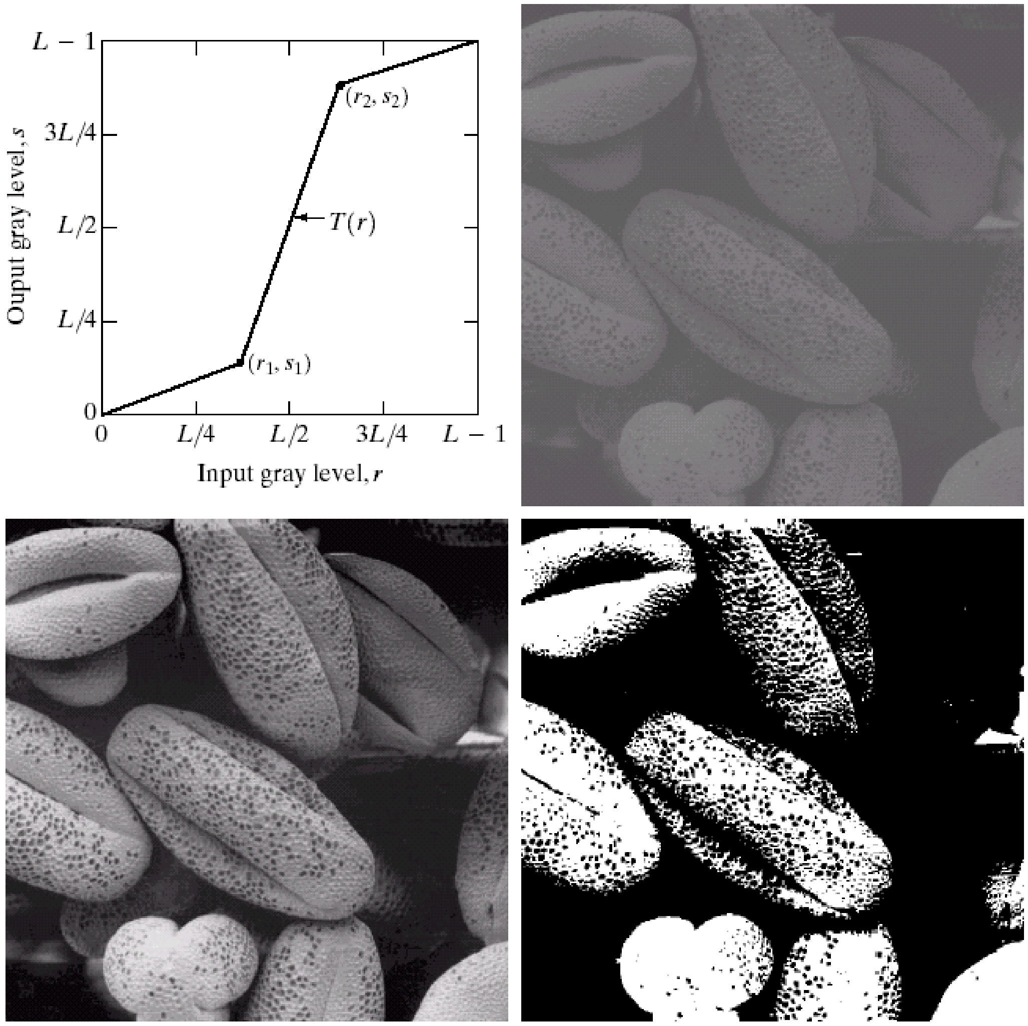






Linear stretching

Piece-Wise Linear for Middle Range Image



a b
c d

FIGURE 3.10 Contrast stretching. (a) Form of transformation function. (b) A low-contrast image. (c) Result of contrast stretching. (d) Result of thresholding. (Original image courtesy of Dr. Roger Heady, Research School of Biological Sciences, Australian National University, Canberra, Australia.)

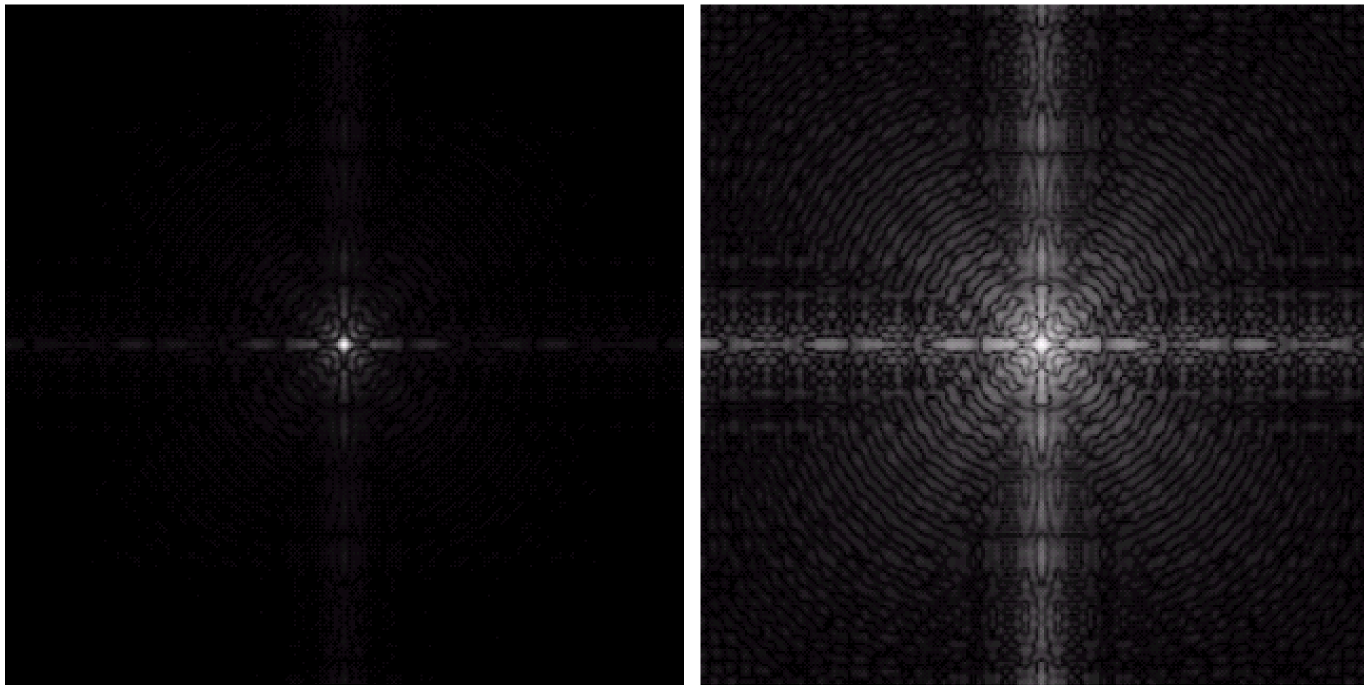
Nonlinear Stretching

- Nonlinear functions with a fixed form
- Fewer parameters to adjust
- Satisfying $0 = f_{\min} \le g \le f_{\max} = L-1$
- Examples
  - Logarithmic transformation $g = b\log(af+1)$
    - Stretch dark region, suppress bright region
  - Exponential transformation $g = b(e^{af}-1)$
    - Expand bright region
  - Power Law $g = af^{k}$
    - K = 2: square law, similar to exponential
    - K = 1/3: cubic root, similar to logarithmic

Example of Log Transformation

a b

FIGURE 3.5
(a) Fourier spectrum.
(b) Result of applying the log transformation given in Eq. (3.2-2) with $c = 1$.



Eq. (3.2-2)
"log" function:
g=c log (1+f)

Power Law (Gamma) Transformations

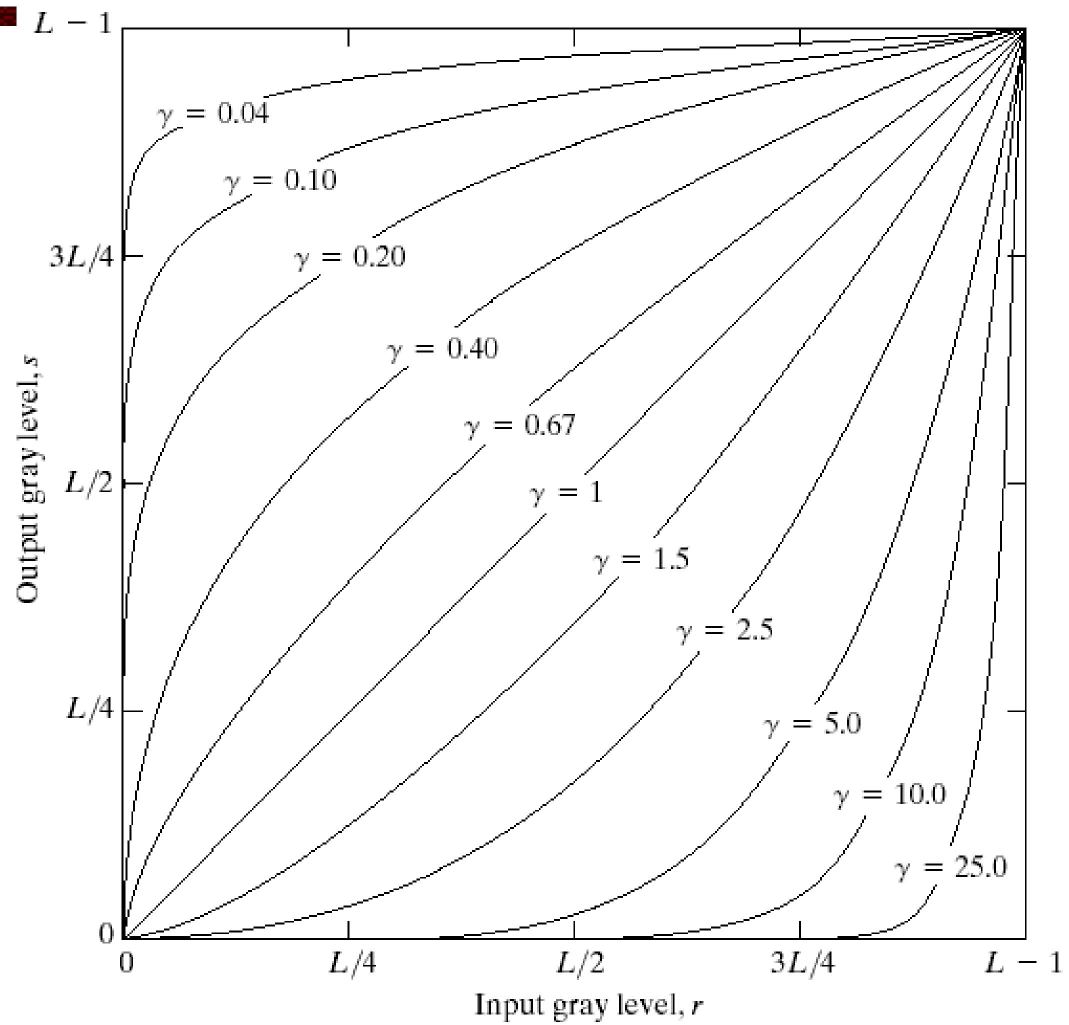


FIGURE 3.6 Plots of the equation $s = cr^{\gamma}$ for various values of γ ($c = 1$ in all cases).

Histogram Equalization

- Transforms an image with an arbitrary histogram to one with a flat histogram
  - Suppose f has PDF $p_F(f)$, $0 \le f \le 1$
  - Transform function (continuous version)

$$g(f) = \int_0^f p_F(t)\,dt$$

  - g is uniformly distributed in (0, 1)

Histogram Equalization

Proof

$$g(f) = \int_{f_{\min}}^{f} p_F(t)dt,$$

$$p_G(g) = \frac{p_F(f)}{\left|\frac{dg}{df}\right|}, \quad g \in (0,1)$$

$$\frac{dg}{df} = p_F(f)$$

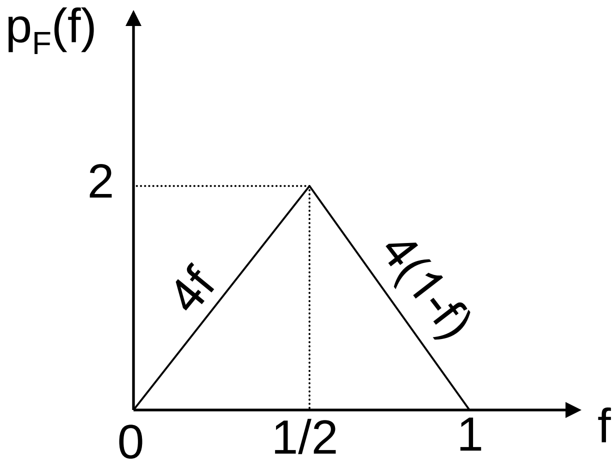
$$p_G(g) = 1, \quad g \in (0,1)$$

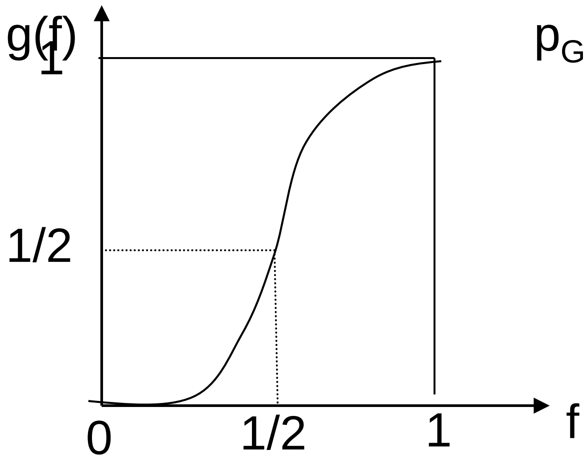
Example

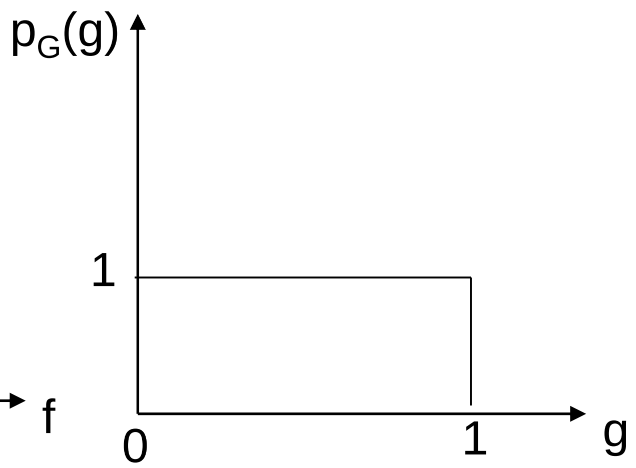
$$p_F(f) = \begin{cases} 4f & f \in (0,1/2) \\ 4(1-f) & f \in (1/2,1) \end{cases}$$

$$g(f) = \begin{cases} \int_0^f 4f\,df = 2f^2 & f \in (0,1/2) \\ \int_0^{1/2} 4f\,df + \int_{1/2}^f 4(1-f)\,df = 1 - 2(f-1)^2 & f \in (1/2,1) \end{cases}$$

$$p_G(g) = 1, \quad g \in (0,1)$$







Discrete Implementation

- For a discrete image f which takes values k=0,…,K-1, use

$$\tilde{g}(l) = \sum_{k=0}^{l} p_F(k), l = 0,1,\ldots,K-1.$$

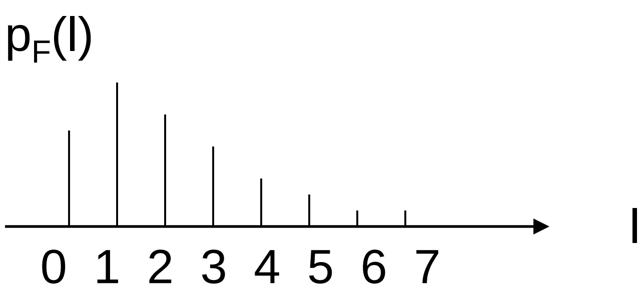
- To convert the transformed values to the range of (0, L-1):

$$g(l) = round\left\{\left(\sum_{k=0}^{l} p_F(k)\right) * (L-1)\right\}$$

  – Note: {x} is the rounding of x

Example

f_k	$p_F(l)$	$\tilde{g}_l = \sum_{k=0}^{l} p_F(k)$	$g_l = [\tilde{g}_l * 7]$	$p_G(l)$	g_k
0	0.19	0.19	[1.33]=1	0	0
1	0.25	0.44	[3.08]=3	0.19	1
2	0.21	0.65	[4.55]=5	0	2
3	0.16	0.81	[5.67]=6	0.25	3
4	0.08	0.89	[6.03]=6	0	4
5	0.06	0.95	[6.65]=7	0.21	5
6	0.03	0.98	[6.86]=7	0.16+0.08=0.24	6
7	0.02	1.00	[7]=7	0.06+0.03+0.02=0.11	7



We don't get perfectly flat histogram with discrete

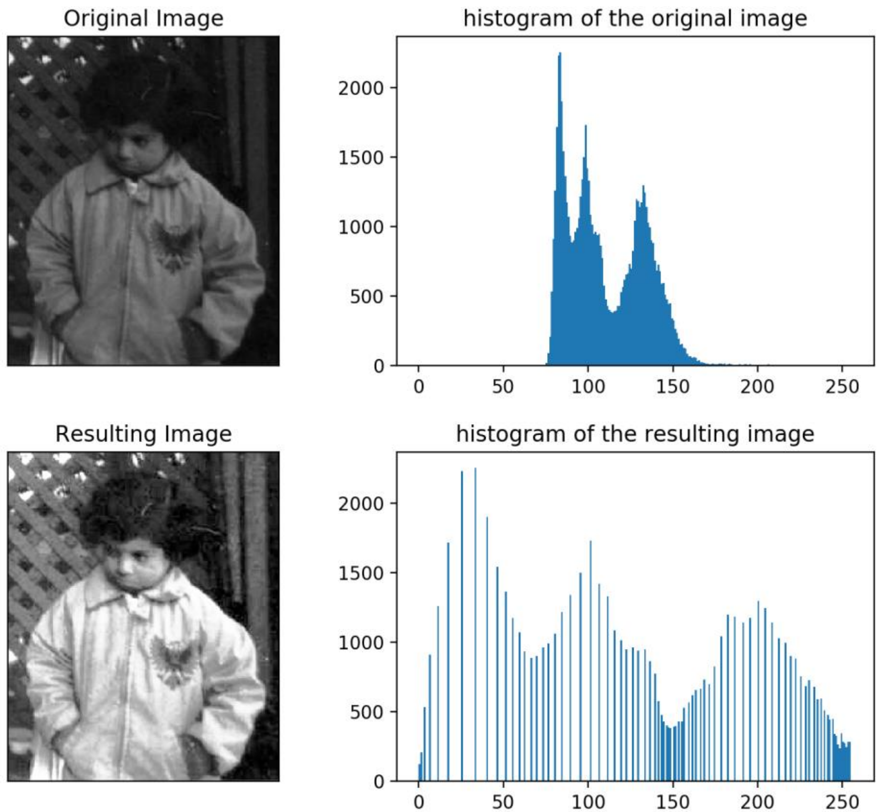
Sample Matlab Code

```
function histogram_eq(inimgname)

img=imread(imgname);
figure; imshow(img);
[M,N]=size(img);

H=imhist(img);
H=H/(M*N);
figure; bar(H);

%Computing the mapping function
for (k=1:256)
   C(k)=uint8(sum(H(1:k))*255);
end;
% C = uint8(cumsum(H)*255);

figure;plot(C);
```

```
%perform mapping
for (i=1:M)
   for (j=1:N)
      f=double(img(i,j))+1;
      histeqimg(i,j)=C(f);
   end;
end;

%note the above loop can be replaced by:
%histeqimg=C(double(img)+1);
%this will be much faster!

figure;
imshow(histeqimg);
```

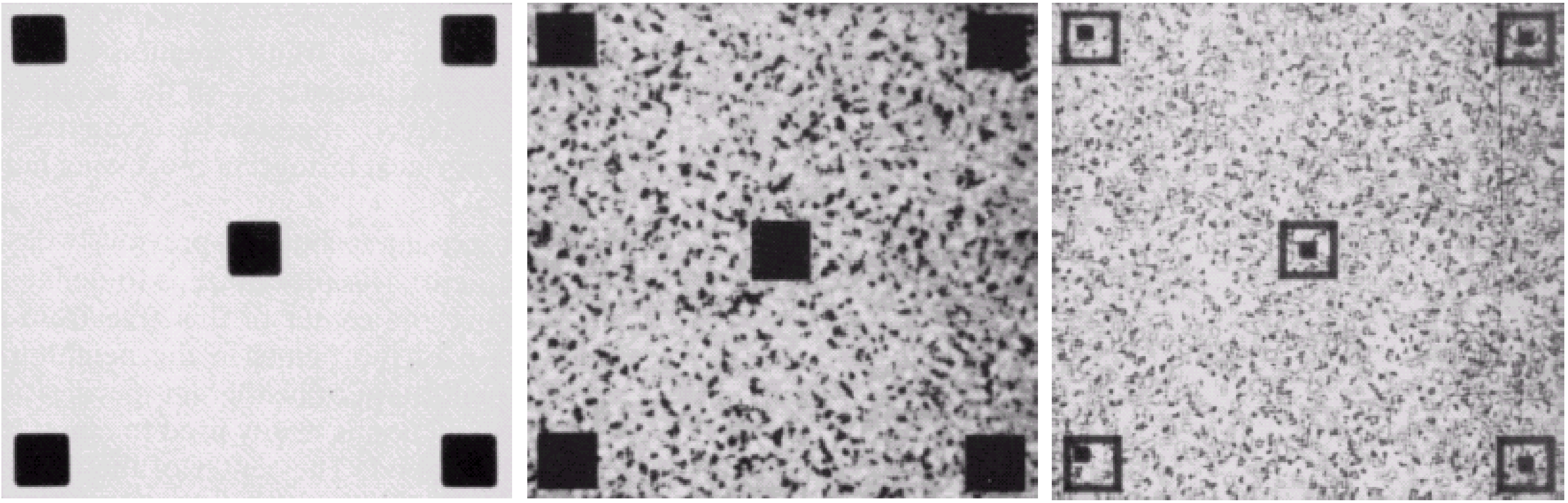
Example Python Code for Histogram Equalization

```
import cv2
import numpy as np
from matplotlib import pyplot as plt

# read the image using openCV
img = cv2.imread('kid.jpg',0)
# Calculate the histogram and corresponding bins
hist,bins = np.histogram(img.flatten(),256,[0,256])
# Calculate the cdf and normalize the values to 0-255
cdf = hist.cumsum()
cdf_normalized = cdf * 255/ cdf[-1]
# Replace the vales with normalized cdf values
img_histeq = cdf_normalized[img]

#display results
fig = plt.figure()
ax1 = plt.subplot(2,2,1)
ax1.get_xaxis().set_visible(False)
ax1.get_yaxis().set_visible(False)
plt.imshow(img,cmap=plt.cm.gray)
ax2 = plt.subplot(2,2,2)
plt.hist(img.ravel(),256,[0,256])
ax3 = plt.subplot(2,2,3)
ax3.get_xaxis().set_visible(False)
ax3.get_yaxis().set_visible(False)
plt.imshow(img_histeq,cmap=plt.cm.gray)
ax4 = plt.subplot(2,2,4)
plt.hist(img_histeq.ravel(),256,[0,256])
plt.show()
```



Local Histogram Modification

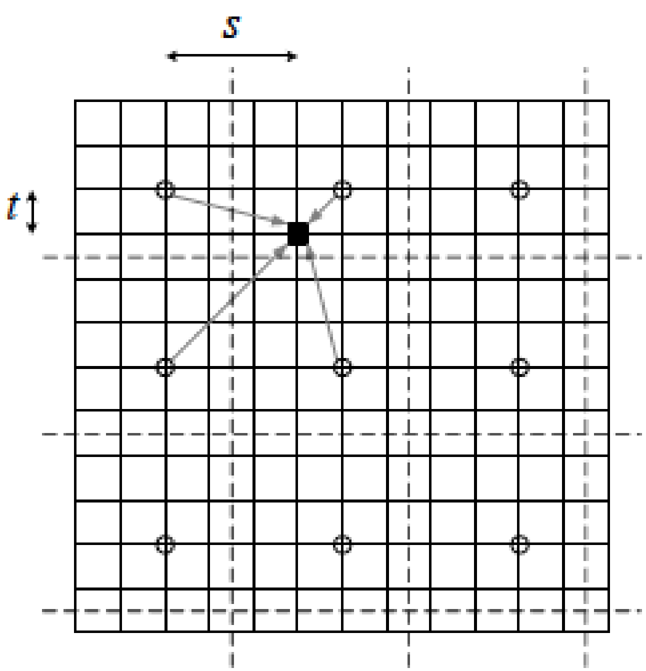
- Compute histogram in each local block and use it to modify the center pixel



a b c

FIGURE 3.23 (a) Original image. (b) Result of global histogram equalization. (c) Result of local histogram equalization using a 7×7 neighborhood about each pixel.

Adaptive Histogram Equalization



Using non-overlapping blocks to compute the histograms and the mapping function for each block center.

The black square pixel's mapping function $f_{s,t}(I)$ is determined by interpolating the 4 mapping functions of the four block centers

Using blinear weights determined based on its distance to the block centers (s,t), (1-s,t), (s,1-t), (1-s,1-t)

$$f_{s,t}(I) = (1-s)(1-t)f_{00}(I) + s(1-t)f_{10}(I) + (1-s)tf_{01}(I) + stf_{11}(I)$$

From [Szeliski2010]

Essential idea behind MATLAB adapthist( ) function

(a) (b) (c)

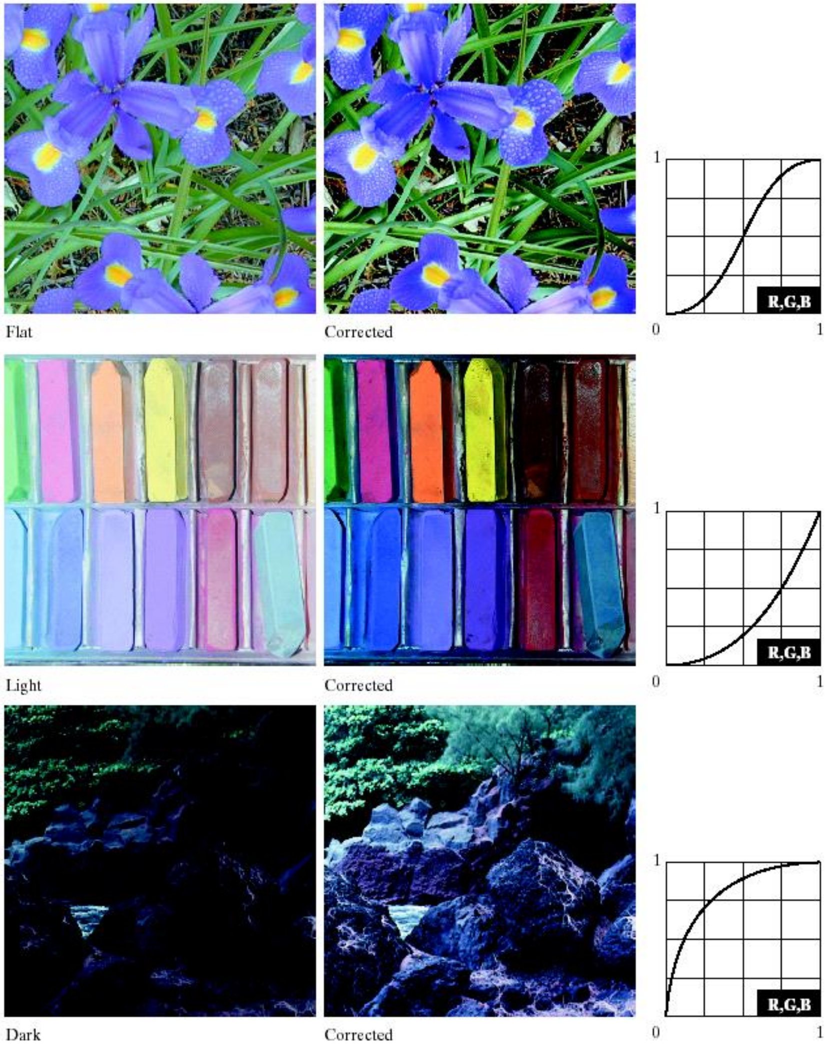
Figure 3.8 Locally adaptive histogram equalization: (a) original image; (b) block histogram equalization; (c) full locally adaptive equalization.

From [Szeliski2010]

Contrast Enhancement for Color Images

- How should we apply the previous techniques to color images
  - To all three color components (RGB or CMY) separately
  - To the Intensity component only while keeping the Hue and Saturation in the HSI coordinate
  - Can also enhance saturation for more vivid colors
  - Can change individual color components to add certain tone to an image

Examples for Color Image (1)



Examples for Color Image (2)

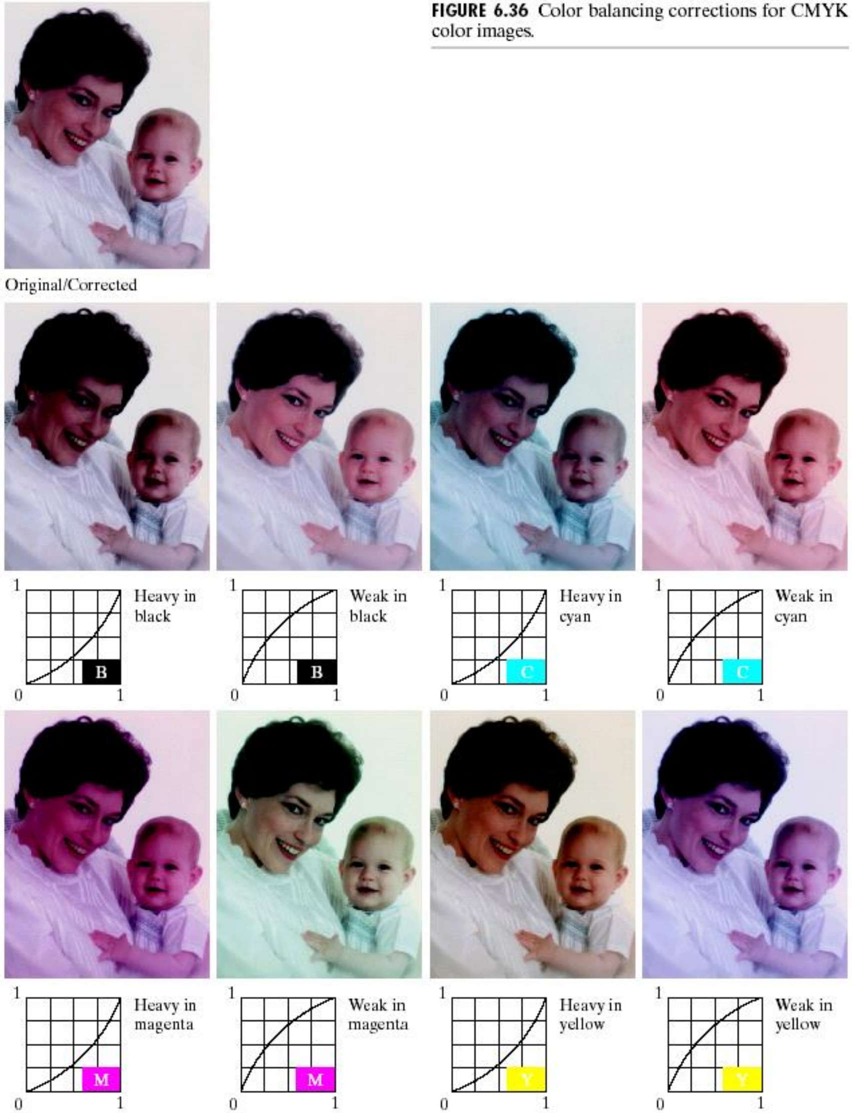


FIGURE 6.36 Color balancing corrections for CMYK color images.

Summary (Contrast Enhancement)

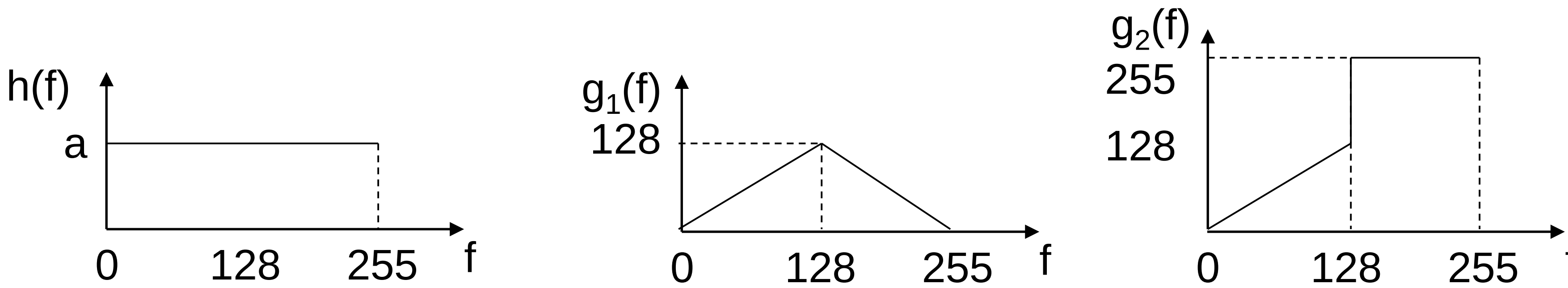
- What is image histogram?
- How to tell whether an an image have a good contrast from its histogram?
- Given the histogram of an image, can you sketch a transformation that will likely improve the image contrast.
- The principle of histogram equalization and adaptive histogram equalization
- Color image enhancement

Reading Assignments

- Richard Szeliski, Computer Vision: Algorithms and Applications. Sec. 3.1.
- Gonzalez & Woods, "Digital Image Processing", Prentice Hall, 2008, 3rd ed. Chapter 3 (Section 3.1 – 3.3) (optional. Containing more detailed explanation)

Written Homework

1. Following figure shows the histogram of an image and two transformation functions. Sketch the histograms of the images obtained by applying the two functions to the original image.



2. Following figure (next slide) shows the histograms of three different images, and three possible point functions. For each original image, which point operation can best equalize its histogram? Briefly explain your reasoning.

Written Homework (cnt'd)

3. For the histogram $h_3(f)$ given in the following figure, determine analytically the histogram equalizing function, assuming the dynamic range of the signal f is from 0 to 1 (i.e. replacing 128 by ½, 255 by 1).
4. An 8-level images have histograms h=[.1, .2, .3, 0, .1, 0, 0, .3]. Find a point function g(f) that will equalize this histogram. Show the resulting mapping function and the histogram. Is the resulting histogram flat? If not, what possibe mechanism you can use to make it better?