EL5123 --- Image Processing

Contrast Enhancement

Yao Wang
Polytechnic University, Brooklyn, NY 11201

With contribution from Zhu Liu, Onur Guleryuz, and Gonzalez/Woods, Digital Image Processing, 2ed

Lecture Outline

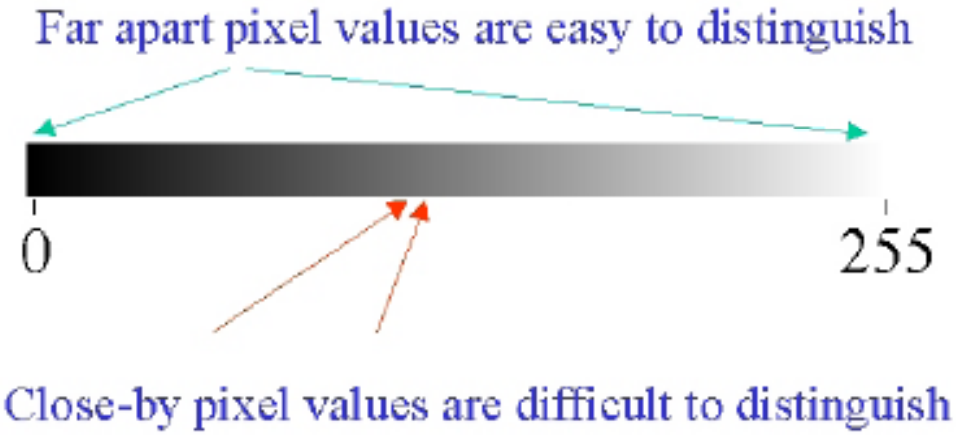
- Introduction
- Linear stretching
- Nonlinear stretching
- Histogram equalization
- Histogram specification
- Adaptive histogram modification

What is Contrast Enhancement

Original image with low contrast

Enhanced image

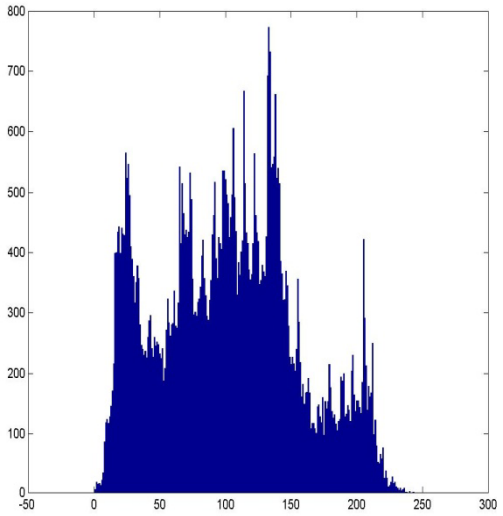
How to enhance the contrast



- 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, f = 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 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);
```

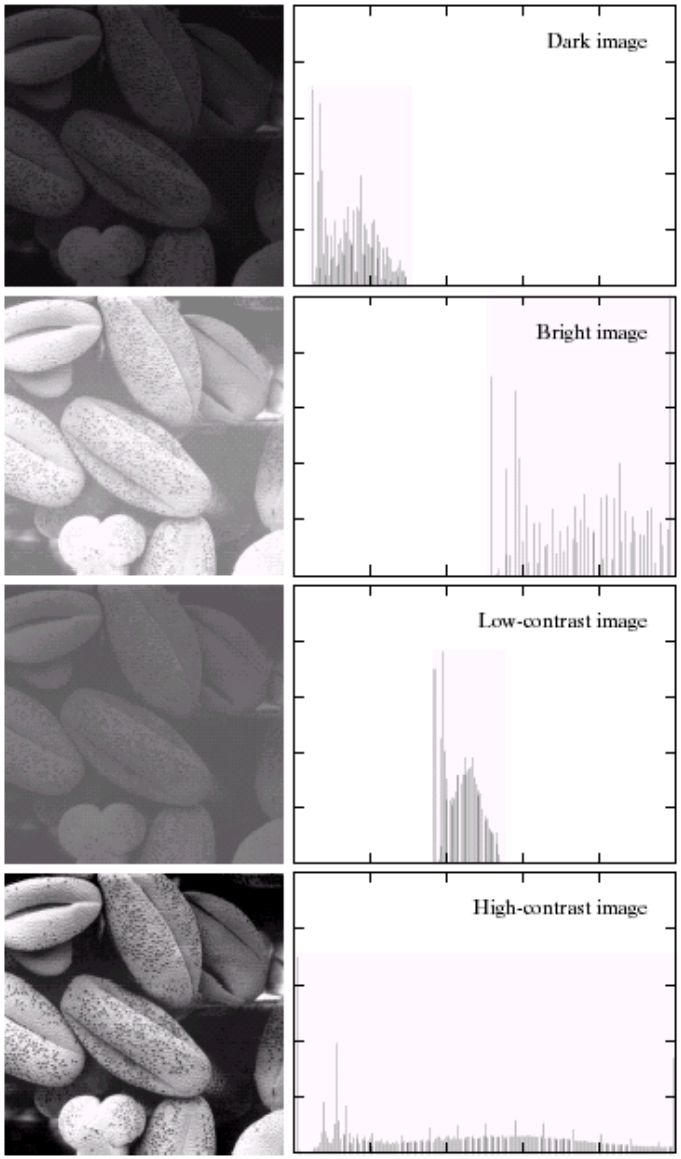
```
% method 2
img = double(img);
h = zeros(256,1);
for i=1:M,
   for j=1:N,
      f = img(i,j);
      h(f+1) = h(f+1) + 1;
   end
end

% 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

Histogram vs. Image Contrast

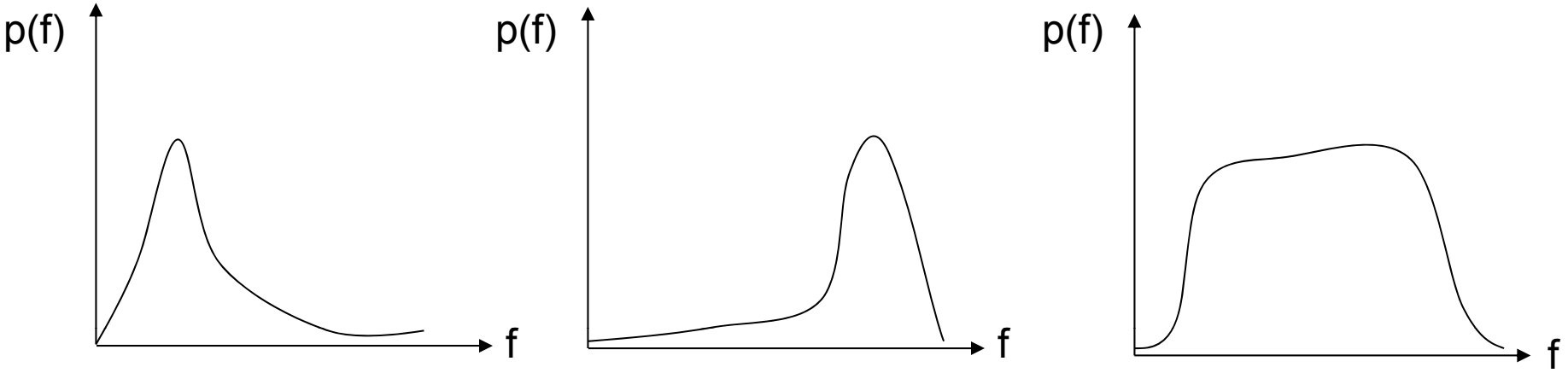


a b

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]

Examples of Histograms

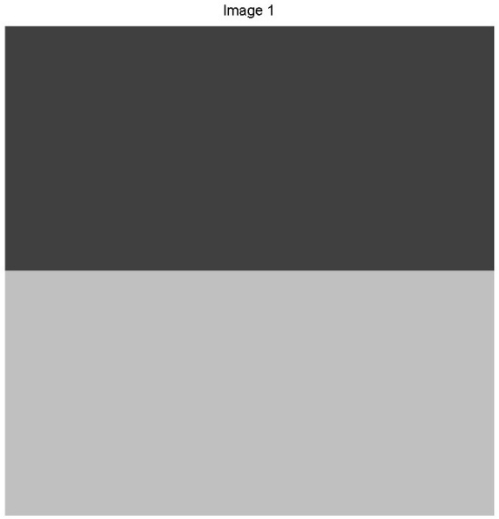


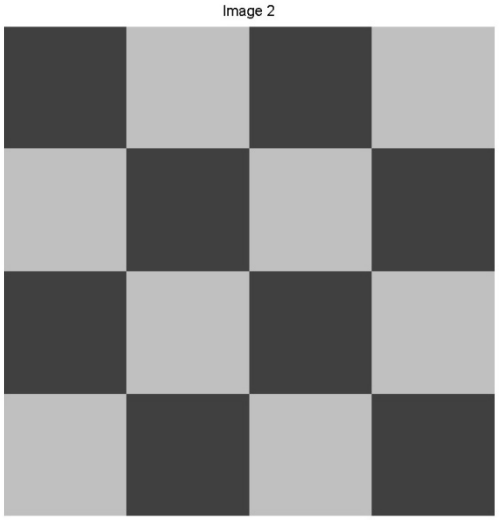
(a) Too dark

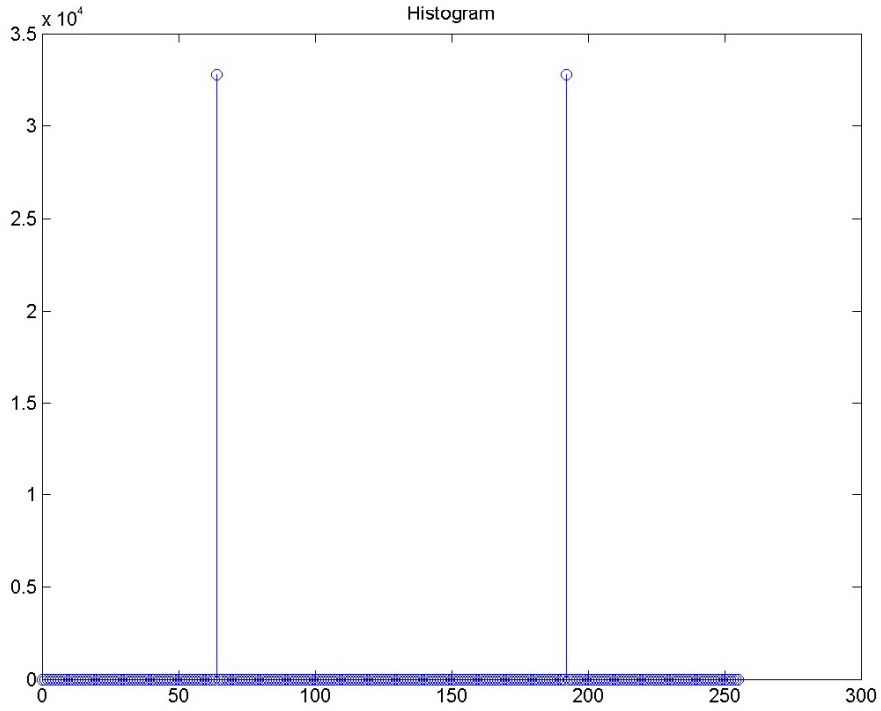
(b) Too bright

(c) Well balanced

Very Different Images May Have Same Histogram!







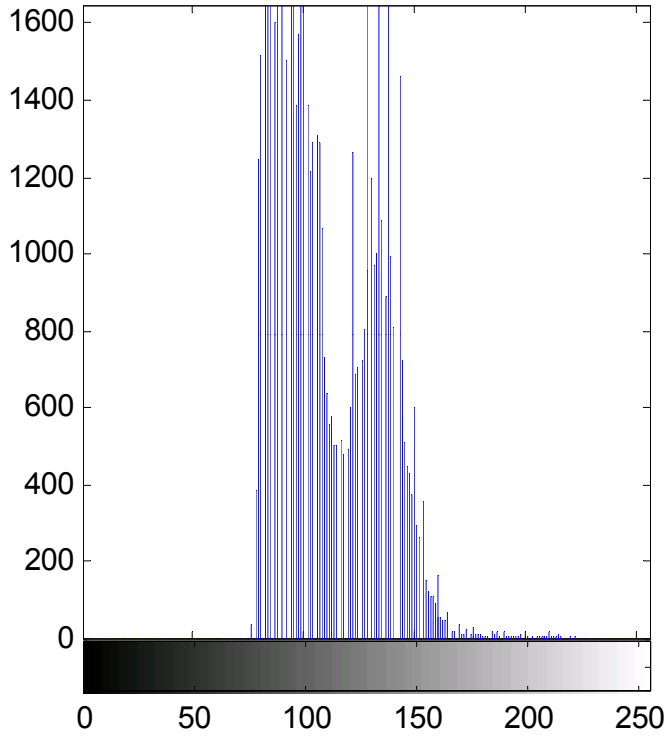
Histogram reflects the pixel intensity distribution, not the spatial distribution!

Previous Example

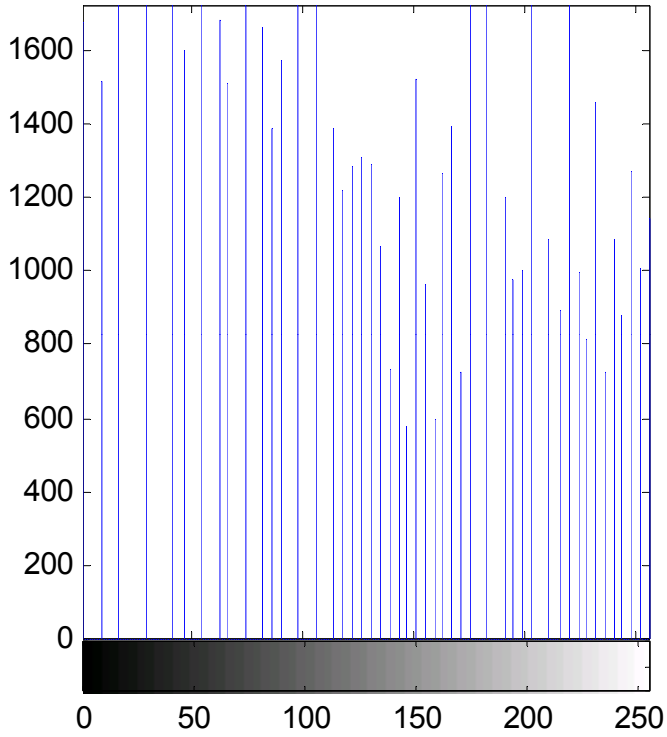
Original image with low contrast

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:

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

- 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

Linear Stretching

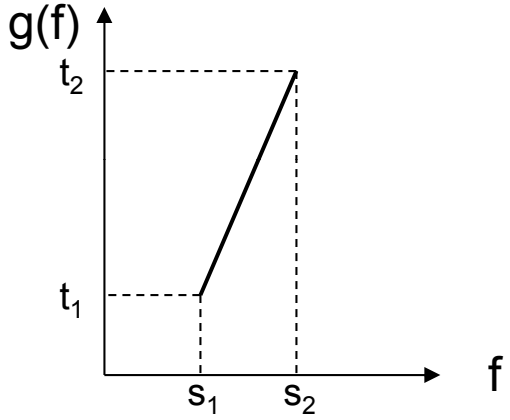
- Enhance the dynamic range by linearly stretching the original gray levels to the range of

$$g(f) = af + b$$

$$[s_1 \quad s_2] \Rightarrow [t_1 \quad t_2]$$

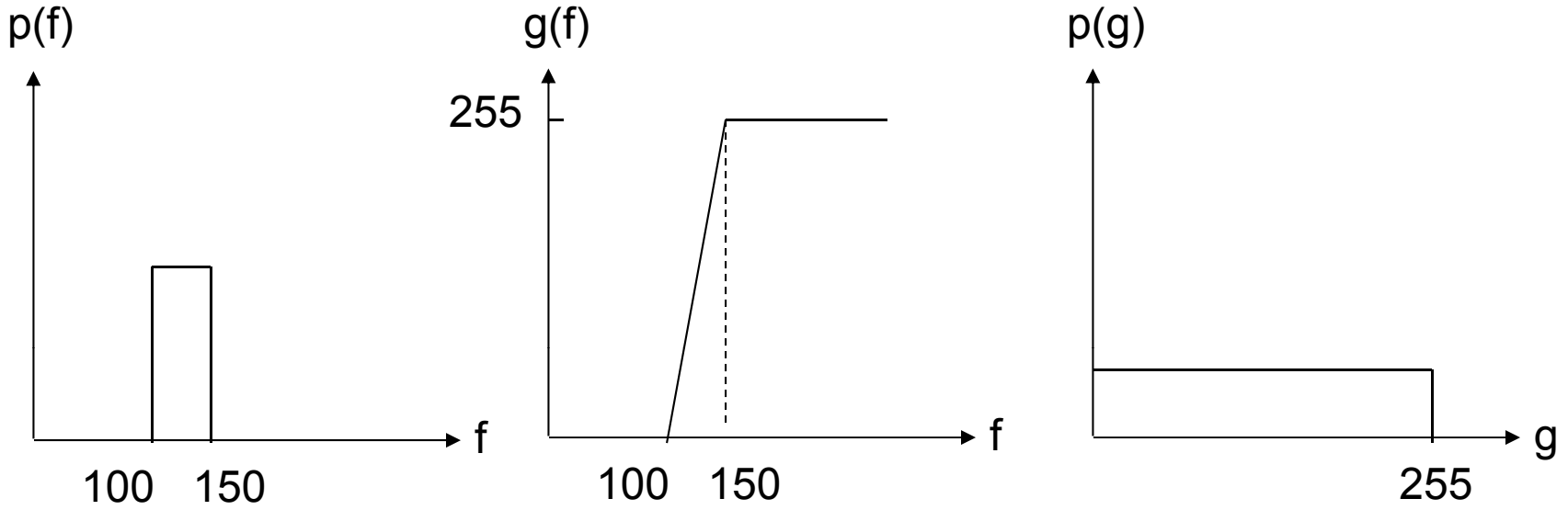
$$g(f) = (f - s_1) * (t_2 - t_1)/(s_2 - s_1) + t_1$$

$$a = (t_2 - t_1)/(s_2 - s_1), \quad b = t_1 - s_1 * (t_2 - t_1)/(s_2 - s_1)$$



- Example
  - The original gray levels are [100, 150].
  - The target gray levels are [0, 255].
  - $g(f) = ((f - 100)/50) * 255, \quad for\ 100 \le f \le 150.$

Illustration of Linear Stretching

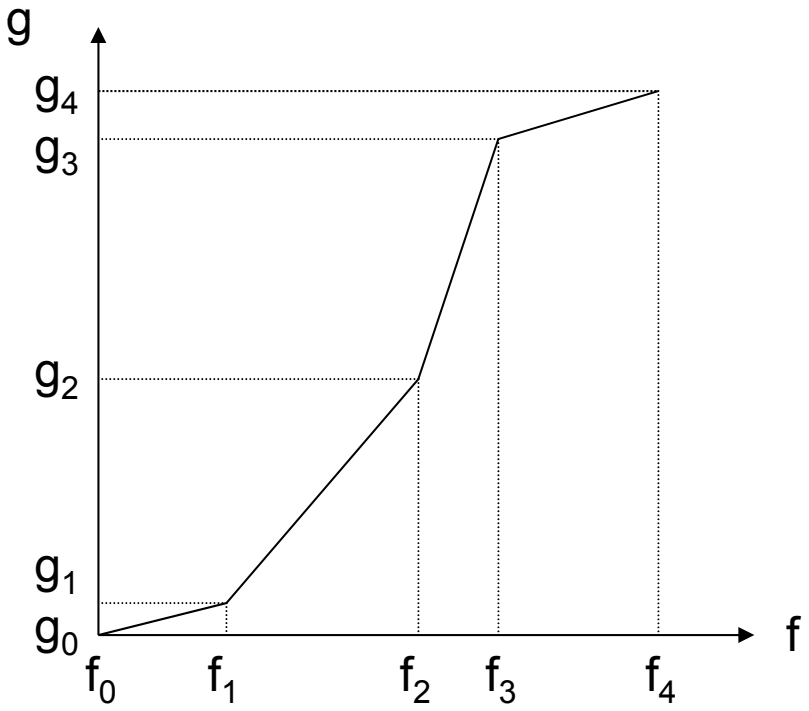


Linear stretching

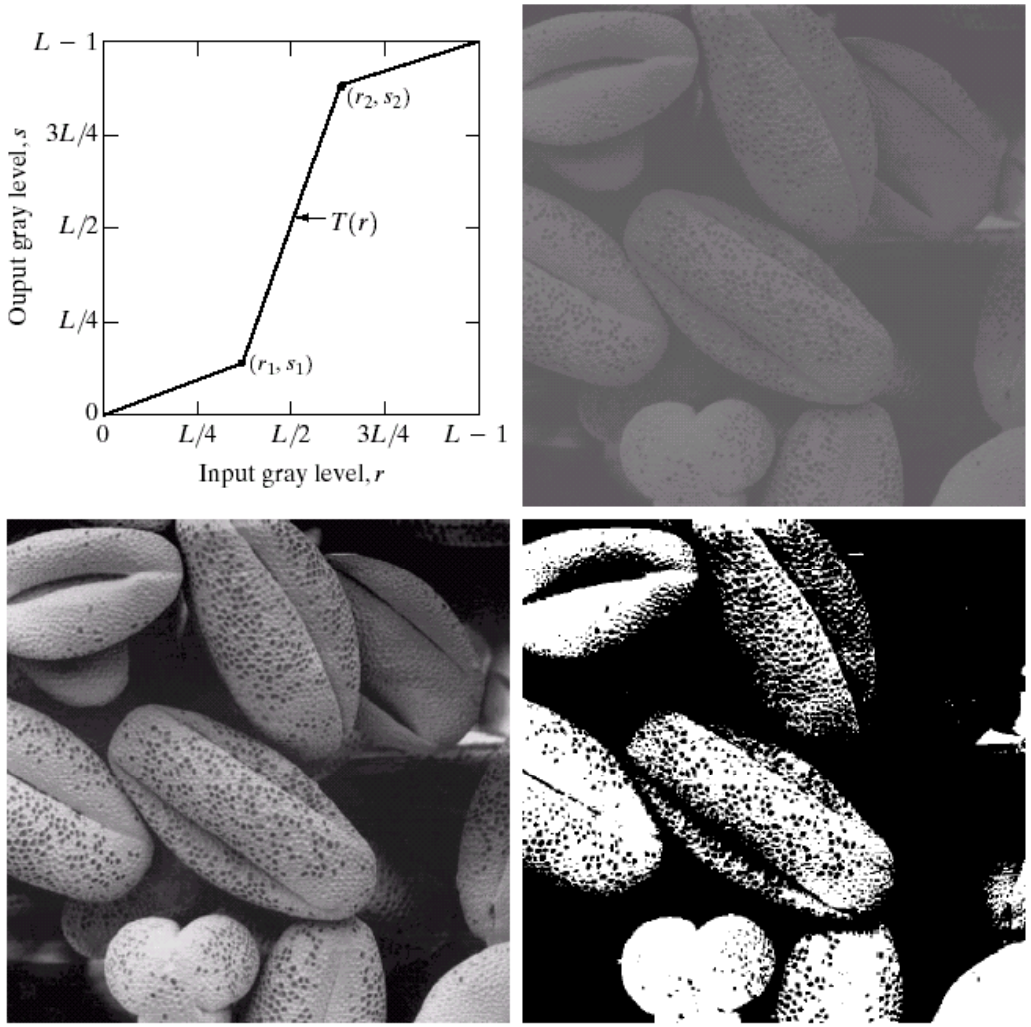
Piecewise Linear Stretching

- K segments
  - Starting position of input $\{f_k, k = 0, \ldots K-1\}$
  - Starting position of output $\{g_k, k = 0, \ldots, K-1\}$
  - Transform function

$$g(f) = \frac{f - f_k}{f_{k+1} - f_k} * (g_{k+1} - g_k) + g_k;$$
$$for \ f_k < f \le f_{k+1}, \quad k = 0,1,\ldots, K-1.$$



Piece-Wise Linear for Middle Range Image



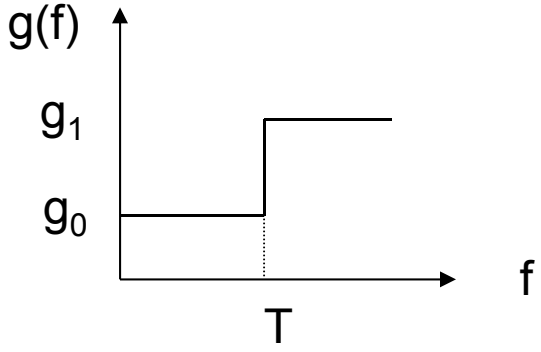
a b
c d

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.)

Special Cases

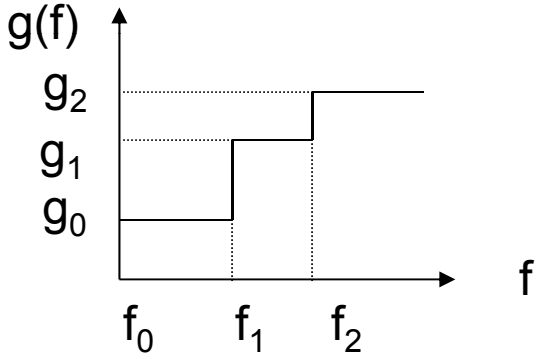
- Thresholding (slicing)

$$g(f)=\begin{cases} g_0 & f \le T \\ g_1 & f > T \end{cases}$$



- Multilevel Slicing

$$g(f)=g_k \quad f_k < f \le f_{k+1}$$

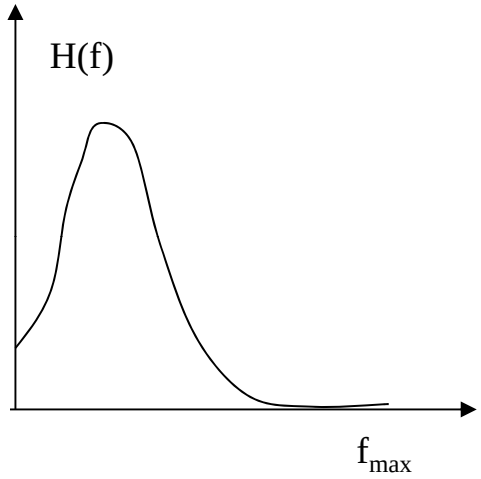


- The mapping function should be monotonically non-decreasing, so that the relative brightness orders of pixels do not change

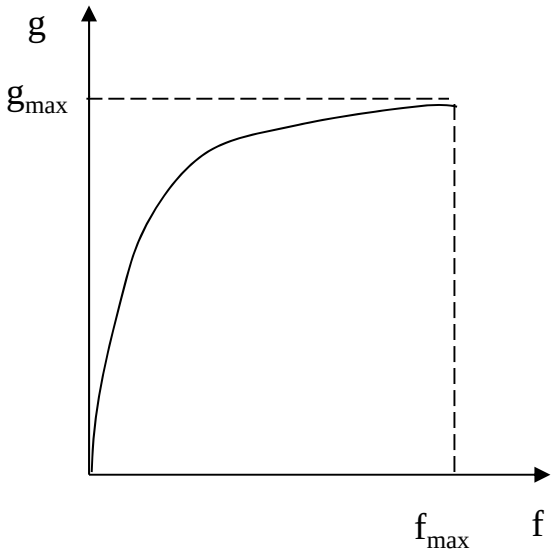
Nonlinear Stretching

- 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

Enhancement of Too-Dark Images

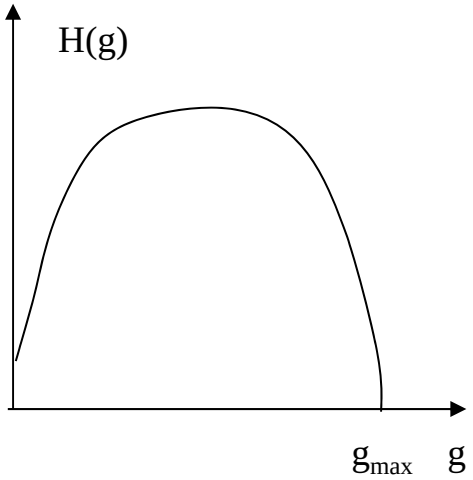


Original histogram



Transformation function:
"log" function: g=c log (1+f)
Or
Power law: g= c f^r, 0<r<1

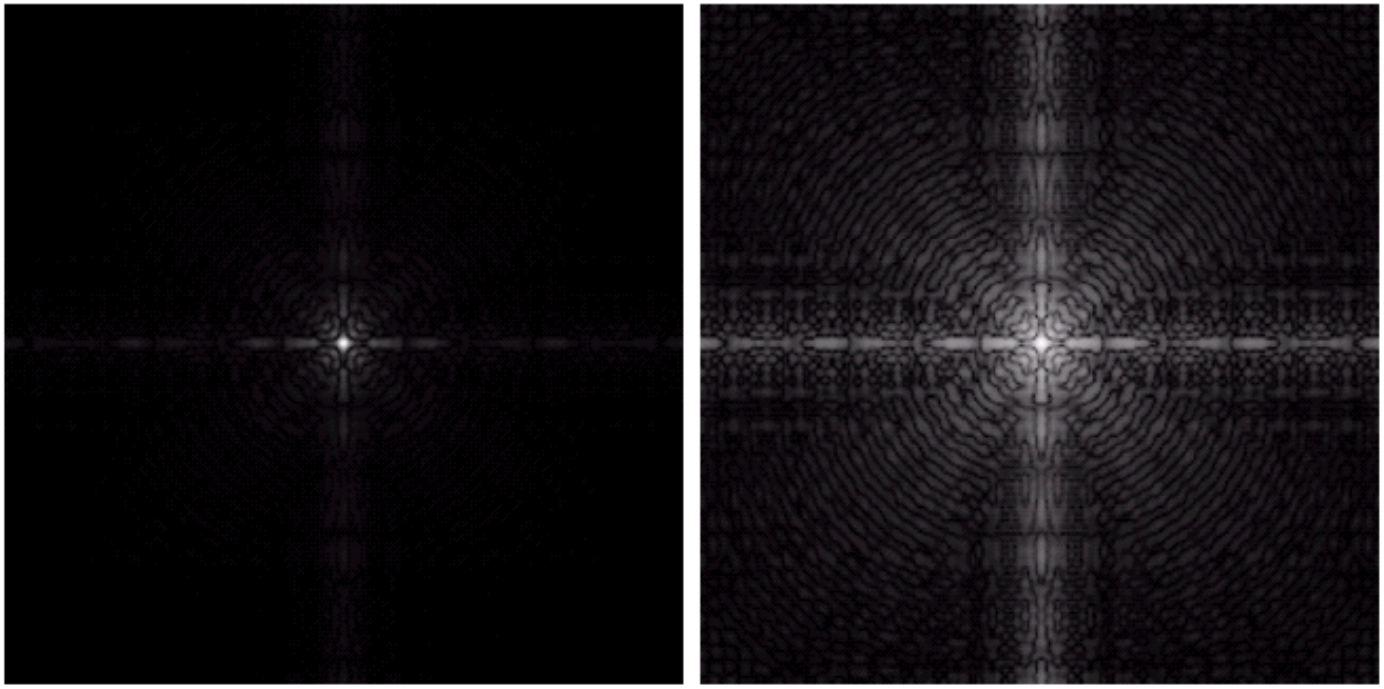
New histogram



Example of Log Transformation

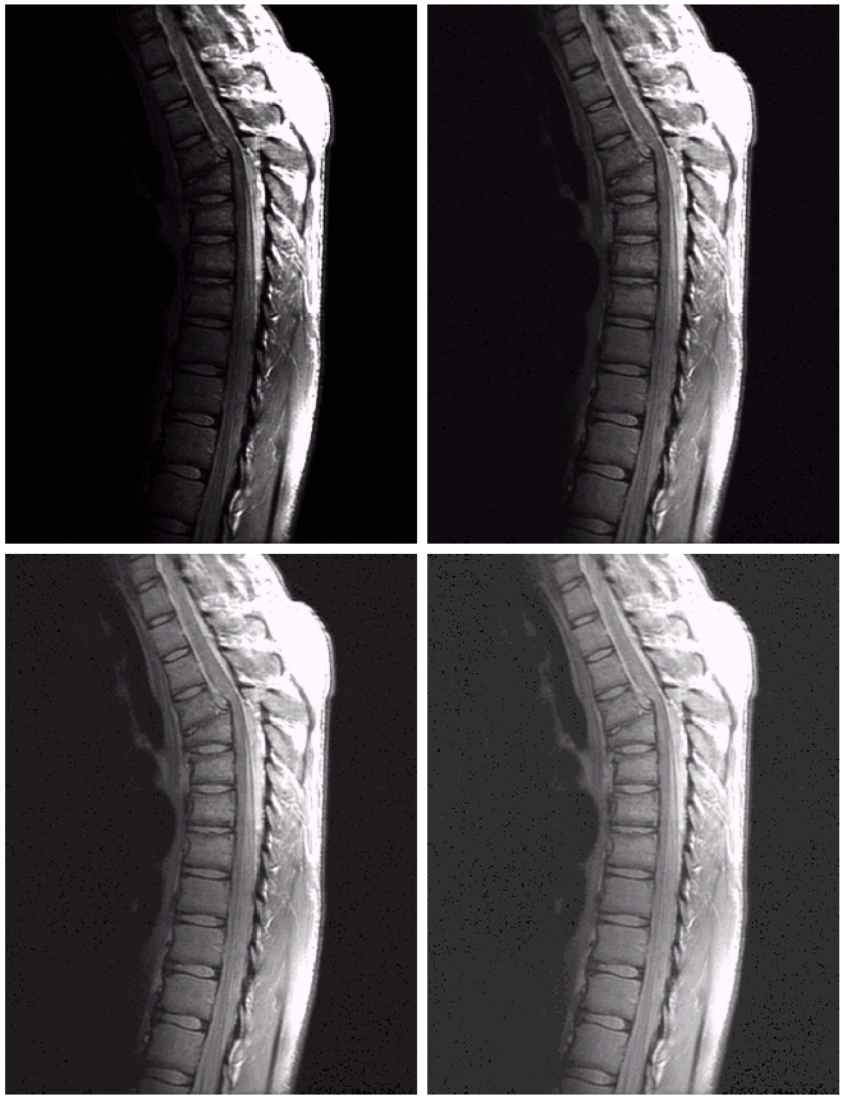
a b

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 for Too-Dark Image



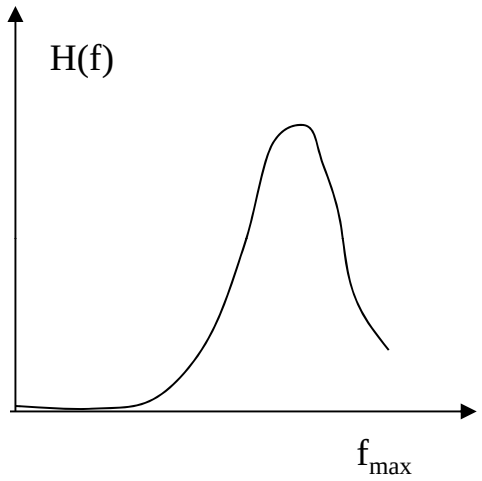
a b
c d

FIGURE 3.8
(a) Magnetic resonance (MR) image of a fractured human spine.
(b)–(d) Results of applying the transformation in Eq. (3.2-3) with $c = 1$ and $\gamma = 0.6, 0.4$, and 0.3, respectively. (Original image for this example courtesy of Dr. David R. Pickens, Department of Radiology and Radiological Sciences, Vanderbilt University Medical Center.)

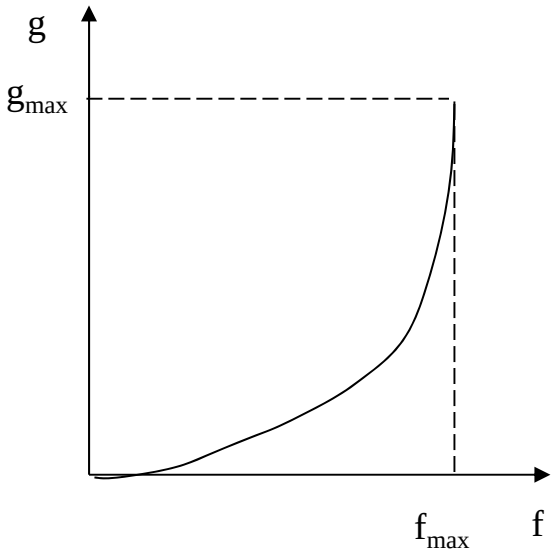
Eq. (3.2-3)
Power law:
g= c f^r

Need to boost dark values, use power law γ<1 (expansion of gray levels)

Enhancement of Too-Bright Images

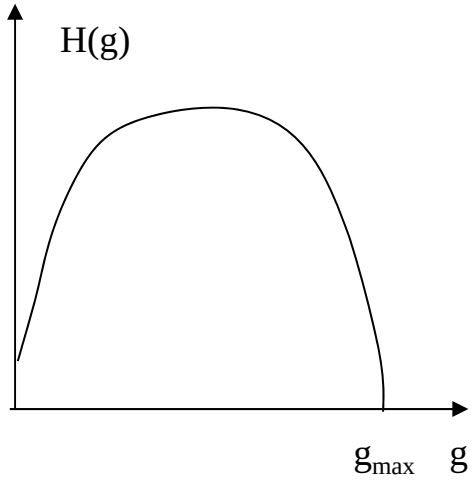


Original histogram



Transformation function:
Power law: g=c f^r, r>1

New histogram



Power Law for Too-Light Image

a b
c d

FIGURE 3.9
(a) Aerial image. (b)–(d) Results of applying the transformation in Eq. (3.2-3) with $c = 1$ and $\gamma = 3.0, 4.0$, and 5.0, respectively. (Original image for this example courtesy of NASA.)

Eq. (3.2-3)
Power law:
g= c f^r

Need to suppress high values, use power law γ >1 (Compression of gray levels)

Power Law Transformations

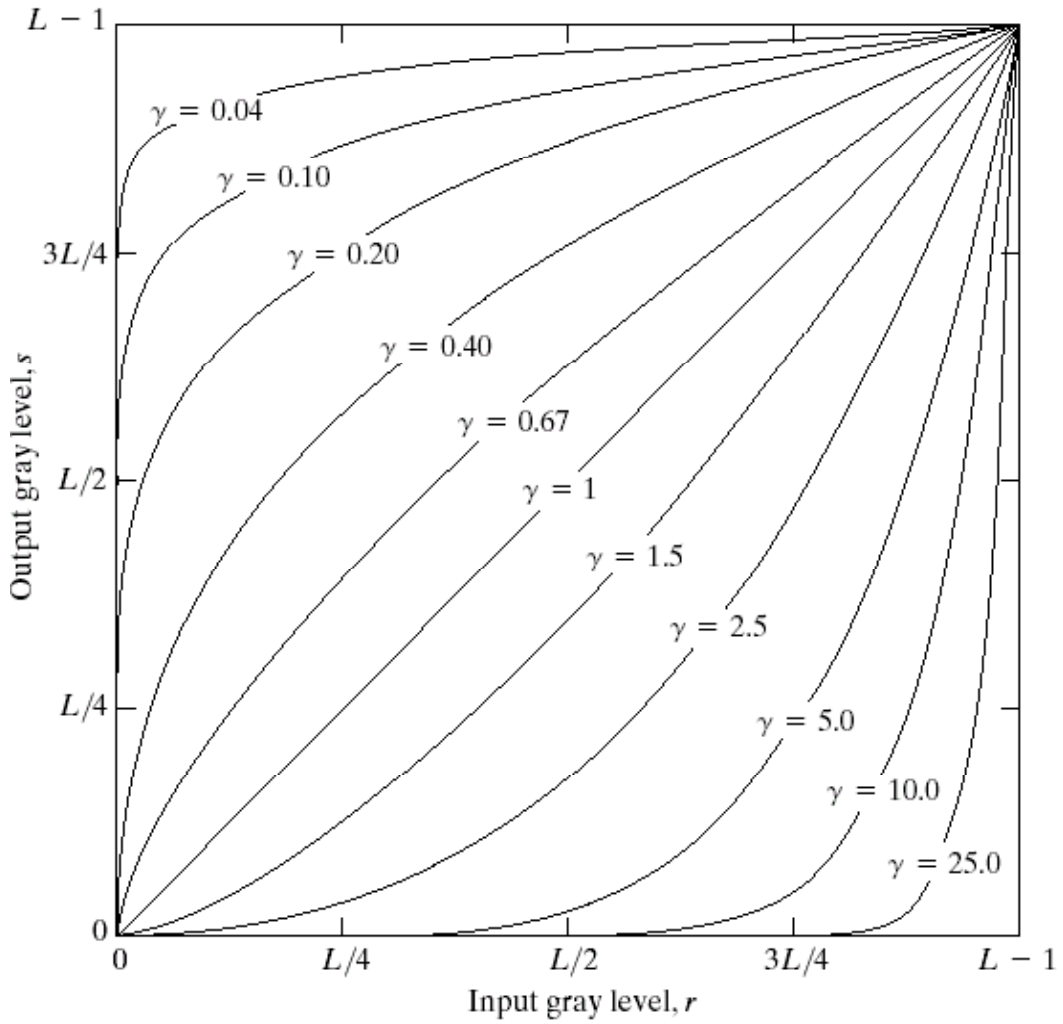
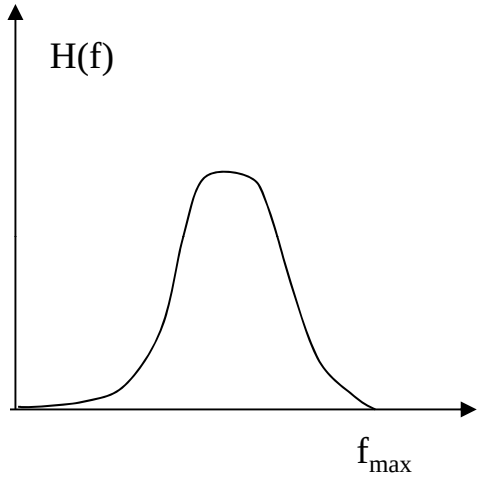
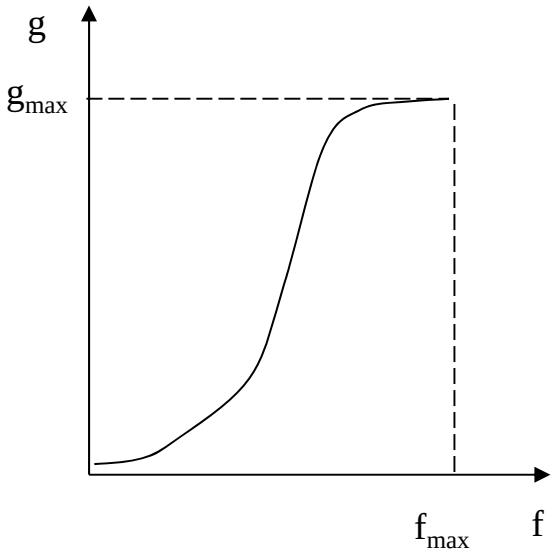


FIGURE 3.6 Plots of the equation $s = cr^{\gamma}$ for various values of γ ($c = 1$ in all cases).

Enhancement of Images Centered near the Middle Range

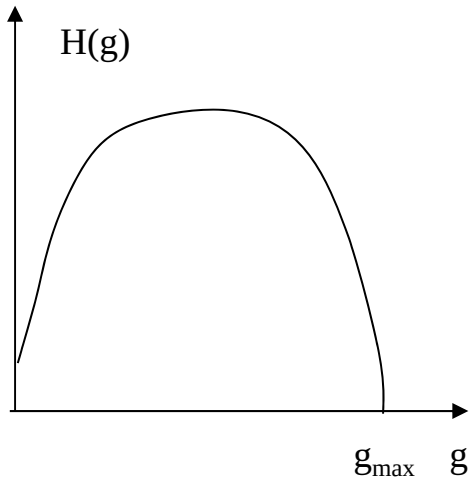


Original histogram



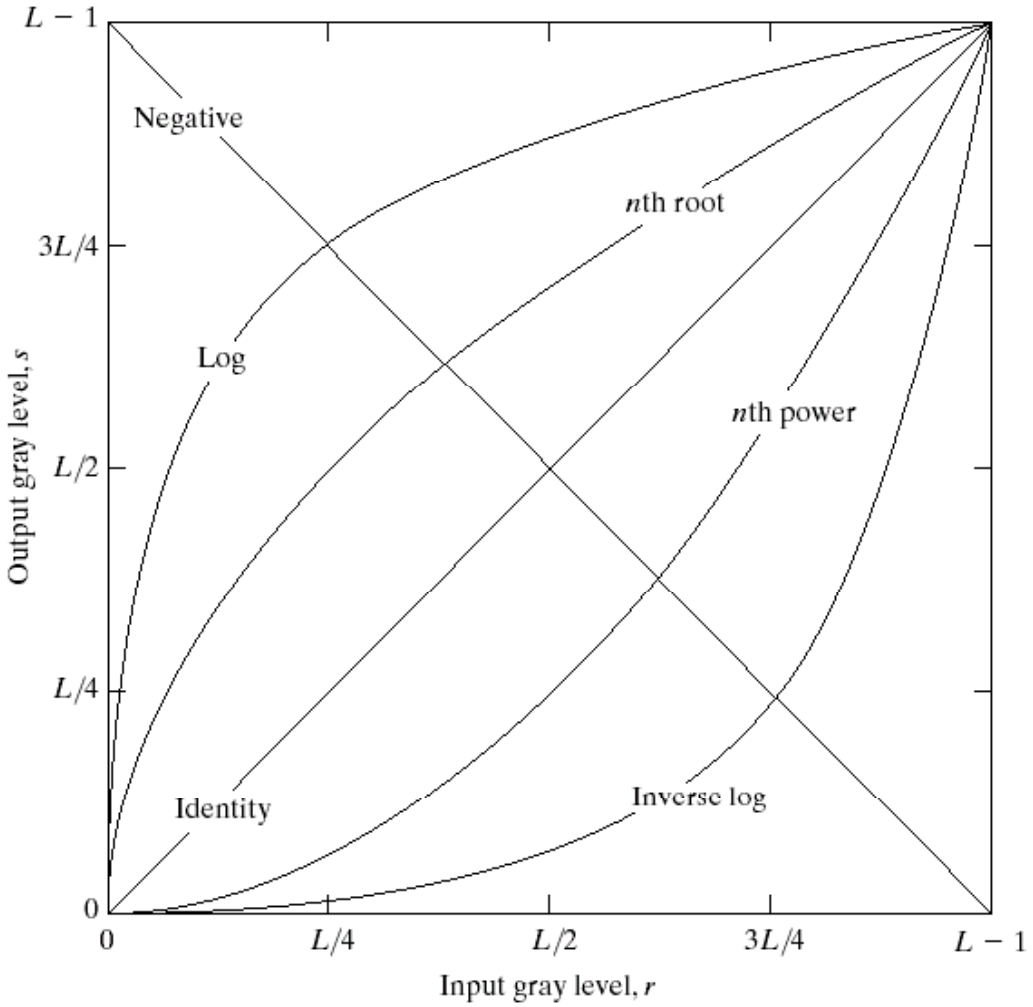
Transformation function

New histogram



Contrast Enhancement by Gray Level Transformation

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



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

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

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

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

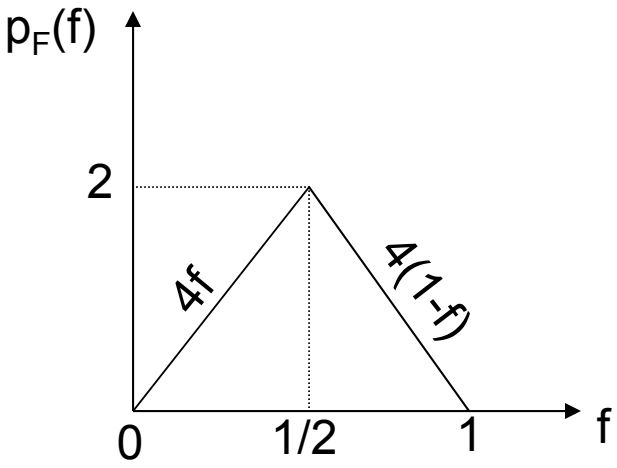
$$p_G(g) = 1, \quad g \in (0,1)$$

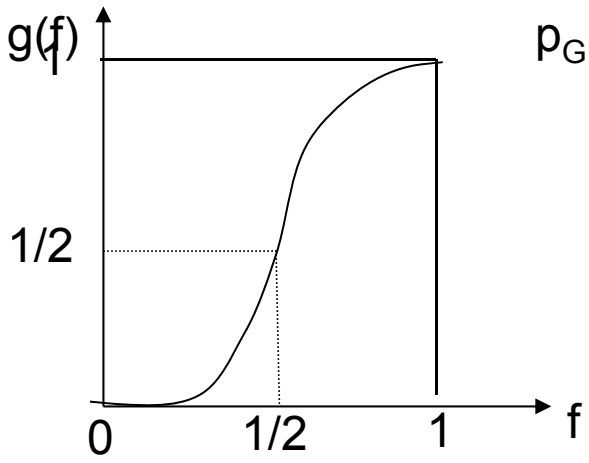
Example

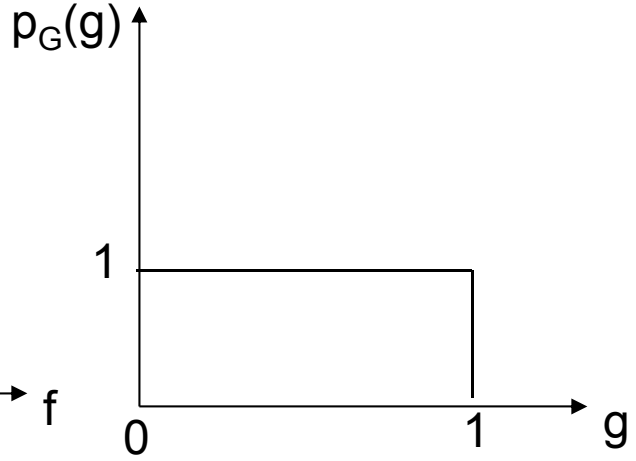
$$p_F(f)=\begin{cases} 4f & f\in(0,1/2) \\ 4(1-f) & f\in(1/2,1) \end{cases}$$

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

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







Discrete Implementation

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

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

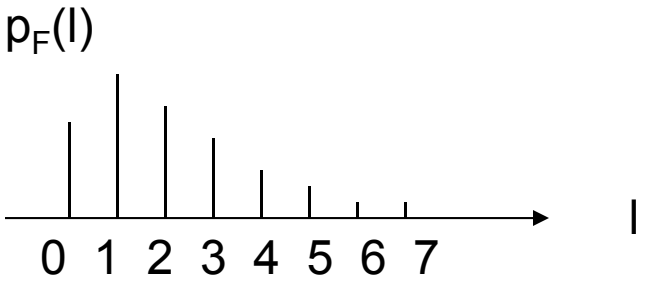
- To convert the transformed values to the range of (0, L-1):

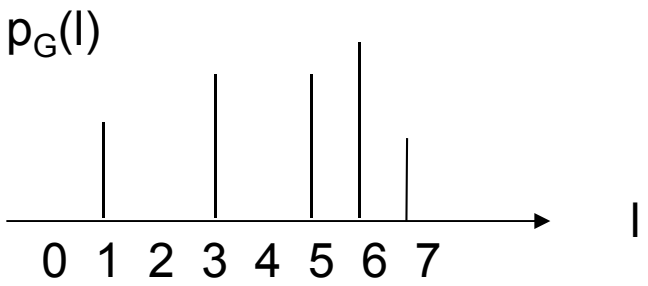
$$g(l) = round\left\{\left(\sum_{k=0}^{l} p_F(k)\right) * (L-1)\right\}$$

  - 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 implementation!

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);
```

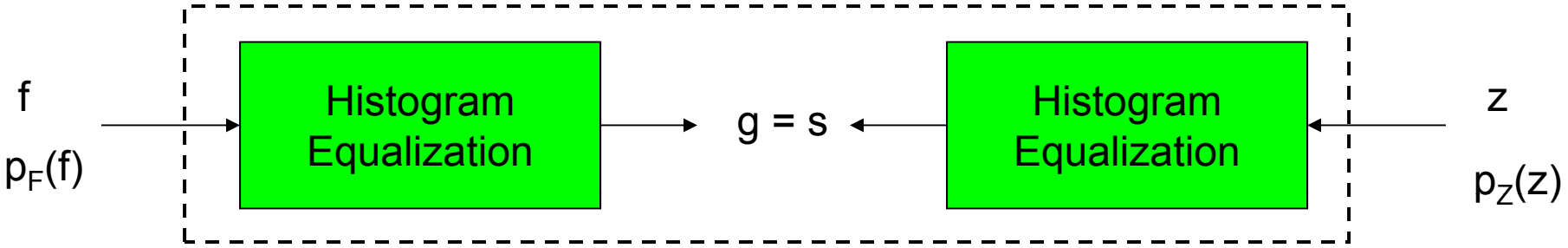
```
%perform mapping
for (i=1:M)
   for (j=1:N)
      f=double(img(i,j))+1;
      histeqimg(i,j)=C(f);
   end;
end;

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

figure;
imshow(histeqimg);
```

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

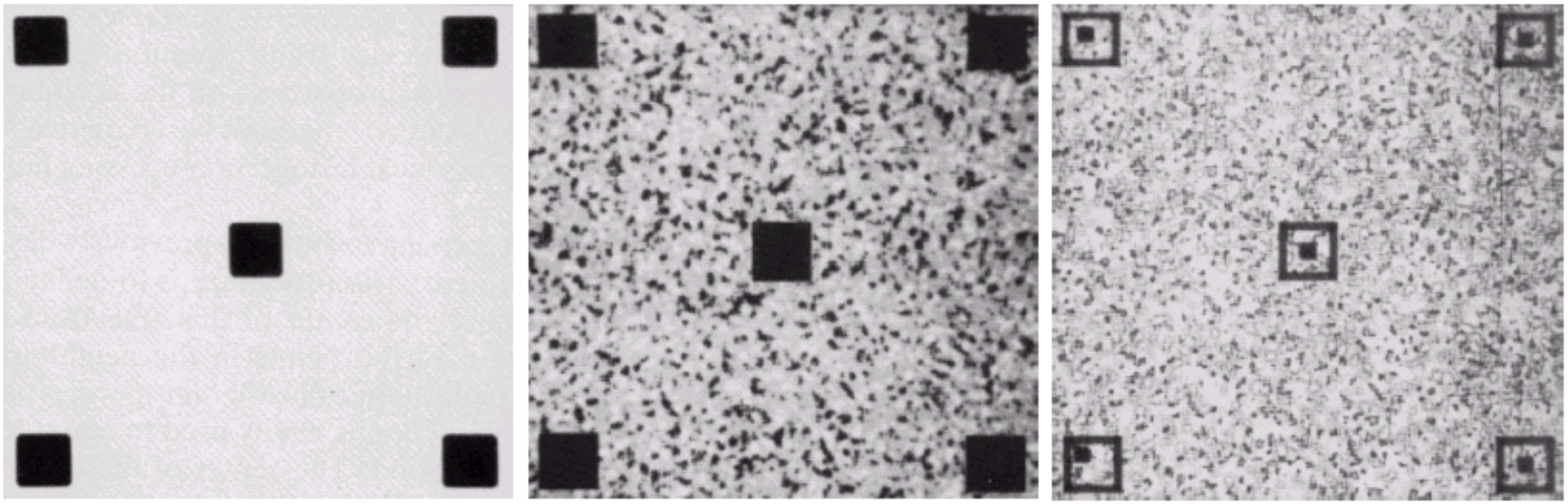
f_k	$p_F(f)$	$g_k = \sum_{i=0}^{k} p_F(i)$		$s_k = \sum_{i=0}^{k} p_Z(i)$	$p_Z(k)$	z_k
0	0.19	0.19		0.0	0.0	0
1	0.25	0.44		0.0	0.0	1
2	0.21	0.65		0.0	0.0	2
3	0.16	0.81		0.15	0.15	3
4	0.08	0.89		0.35	0.20	4
5	0.06	0.95		0.65	0.30	5
6	0.03	0.98		0.85	0.20	6
7	0.02	1.00		1.00	0.15	7

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

Adaptive Histogram Modification

- Local histogram equalization



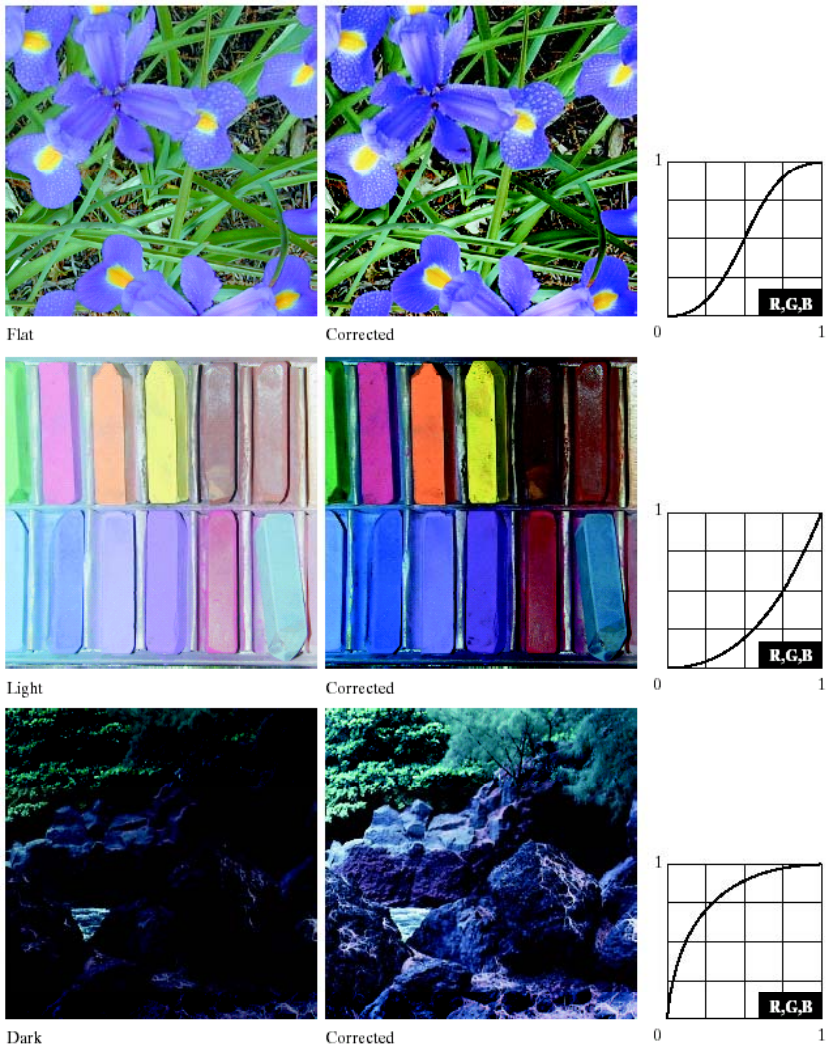
a b c

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.

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)

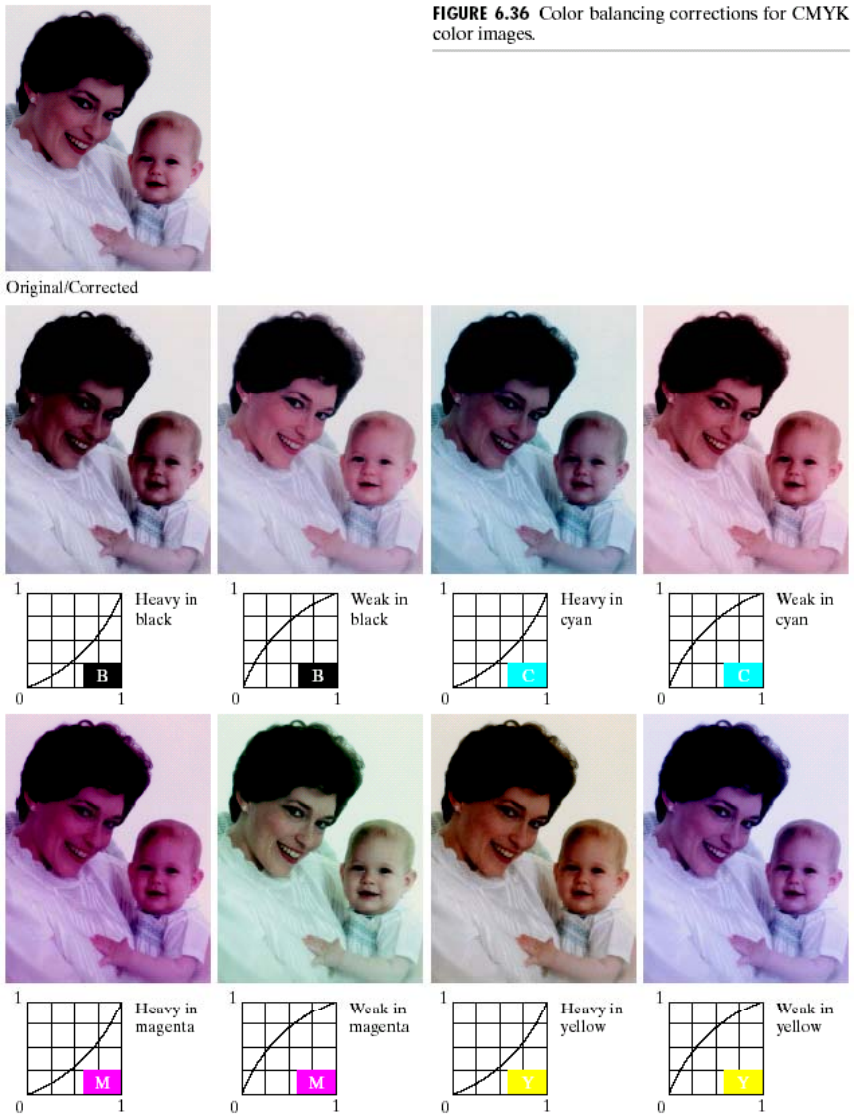
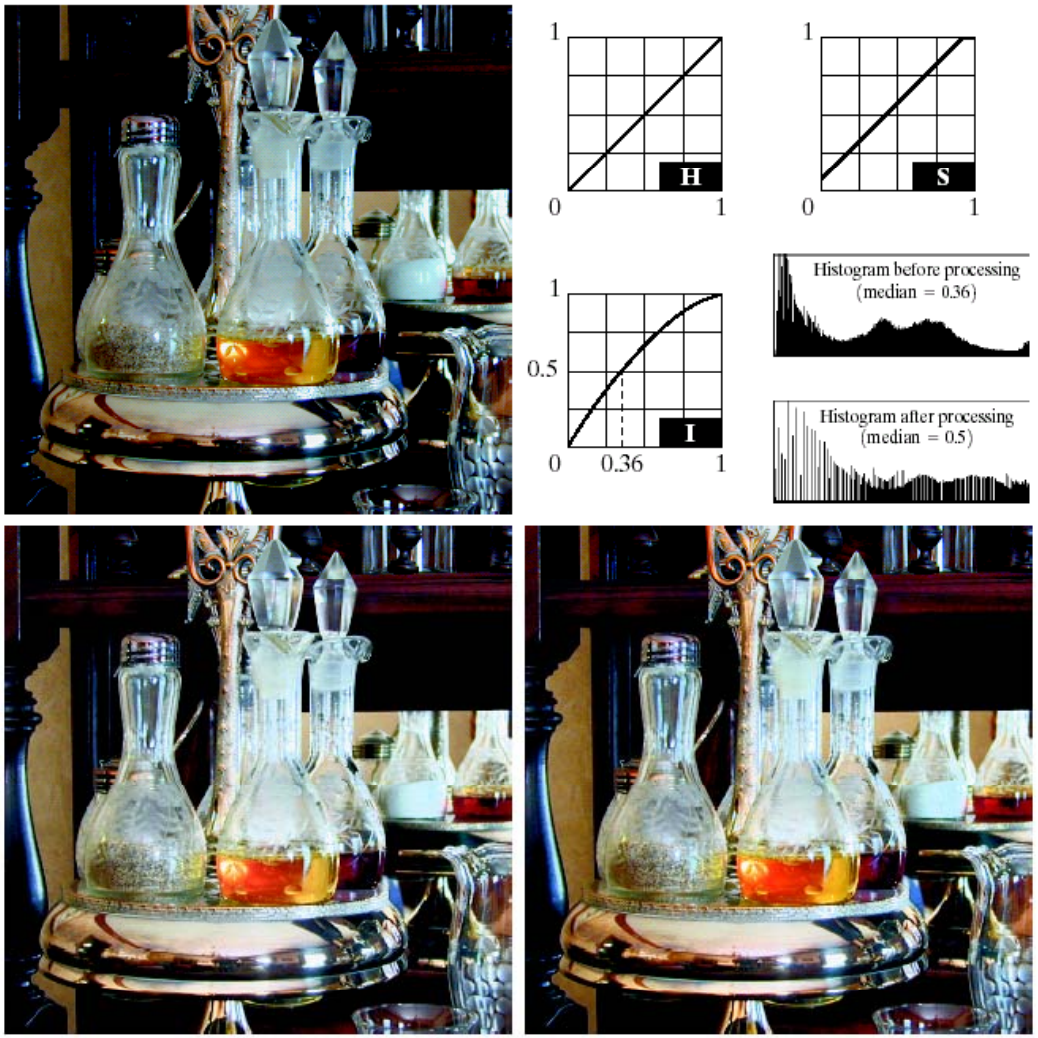


FIGURE 6.36 Color balancing corrections for CMYK color images.

Examples for Color Image (3)



a b
c d

FIGURE 6.37 Histogram equalization (followed by saturation adjustment) in the HSI color space.

Equalize intensity only

Equalize intensity and enhance saturation

Demo

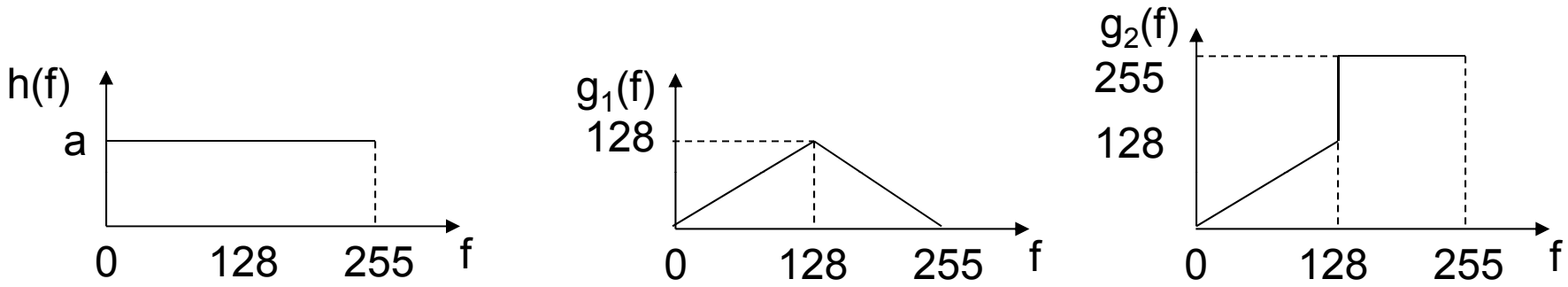
- Photoshop
  - Use “image->adjustments
    - Try brightness/contrast, curves, equalize
- Matlab
  - Imhist, histeq, imadjust, imcontrast
  - imadjdemo

Summary

- 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
- The principle of histogram specification
- Color image enhancement

Homework

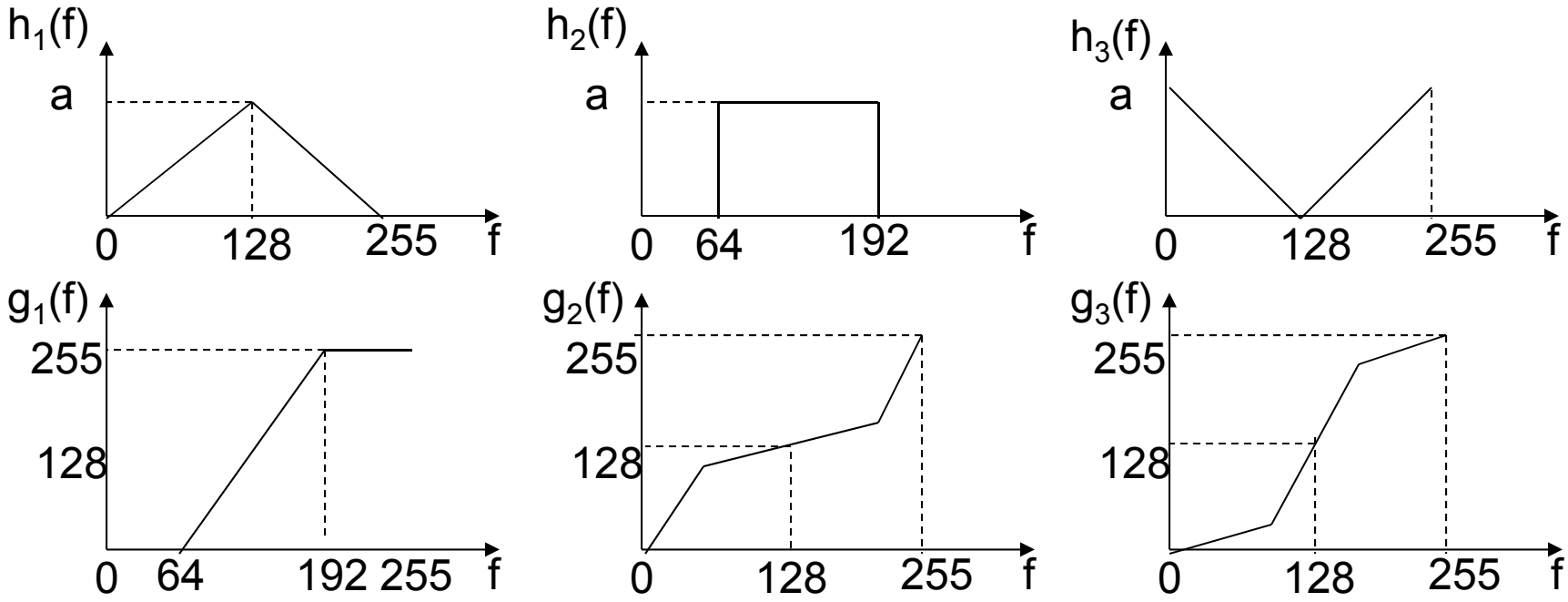
1. Following figure shows the histogram of an image and two transformation functions. Sketch the histograms of the images obtained by applying the two functions to the original image.



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.

Homework (cntd)

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. The two 8-level images have histograms h_A=[.1, .2, .3, 0, .1, 0, 0, .3], h_B=[.1, 0, .2, 0, .4, 0, .3, 0]. Find a point function g(f) that when operating on image A will produce an image C that has a histogram that is similar to h_B.



Computer Assignment

1. Write a Matlab function that can compute the histogram of a grayscale image (assuming 256 levels of gray). Try the three possible ways described in slide 7, and see which one is faster. Finalize your program with the fastest method. In a separate main program, apply the program to a test image, and display the histogram as a stem plot besides the image (using "subplot" function). You are not allowed to simply use the "imhist" or "hist" function in Matlab, although you are encouraged to compare your results with those obtained using these functions.
2. Write a Matlab program that performs histogram equalization on a grayscale image. Your program should: i) compute the histogram of the input image; ii) compute the histogram equalizing transformation function; iii) apply the function to the input image; iv) compute the histogram of the equalized image; v) display (and print) the original and equalized images, as well as their corresponding histograms, all in one figure. You are not allowed to simply use the "histeq" function in Matlab, although you are encouraged to compare your results with those obtained using these functions.
3. Play with the "imadjdemo" program in Matlab, to see the effect of different choice of the transformation functions on image contrast and brightness. Write down your observations in your report.

Readings

- Gonzalez & Woods, “Digital Image Processing”, Chapter 3 (Section 3.1 – 3.3)
- Jain, “Fundamentals of Digital Image Processing”, Chapter 7 (Section 7.1 – 7.3)