Introduction to Image Processing

Ch 3. Intensity Transformations and Spatial Filtering

Kuan-Wen Chen

Ch 3. Intensity Transformations and Spatial Filtering

Image Enhancement

- *No general theory* of image enhancement
- A certain amount of *trial and error* is usually required before a particular image enhancement approach is selected.

credit of this slide: Y. P. Hung

Ch 3. Intensity Transformations and Spatial Filtering

3.1 Background

3.2 Some Basic Intensity Transformation Functions

3.3 Histogram Processing

3.4 Fundamentals of Spatial Filtering

3.5 Smoothing (Lowpass) Spatial Filters

3.6 Sharpening (Highpass) Spatial Filters

3.7 Highpass, Bandreject, and Bandpass Filters from Low Pass Filters

3.8 Combing Spatial Enhancement Methods

3.1 Background

- Spatial Domain Methods
  - Procedures that operate directly on the spatial domain, which refer to the aggregate of pixels composing an image.

$$g(x, y) = T[f(x, y)]$$

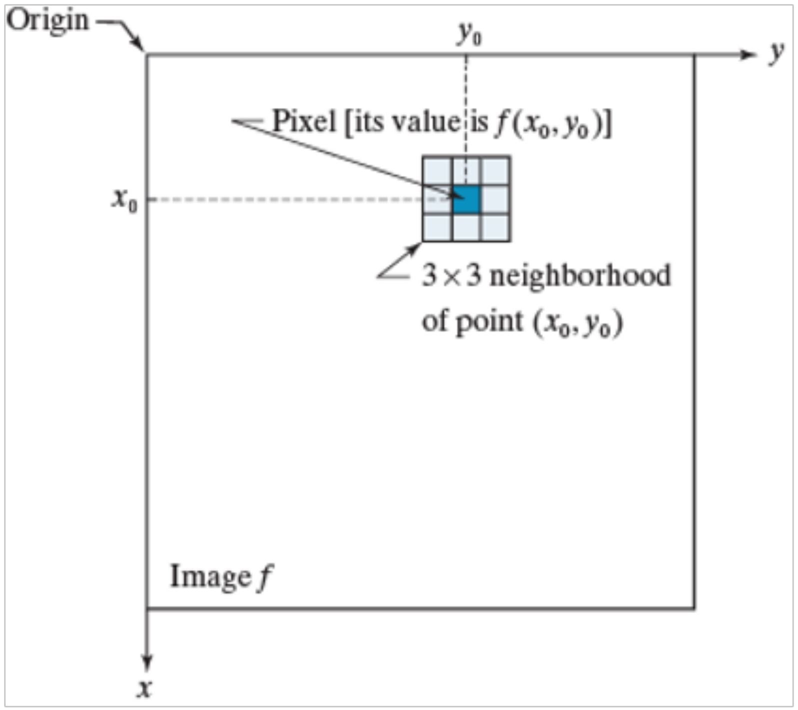
where $f(x, y)$ is the input image, $g(x, y)$ is the processed image and T is some operator defined over some neighbourhood of point (x, y)

3.1 Background

$$g(x, y) = T[f(x, y)]$$

where $f(x, y)$ is the input image, $g(x, y)$ is the processed image and T is some operator defined over some neighbourhood of point (x, y)

For example, an operator T utilizes only the pixels in the area of the image spanned by the neighborhood, e.g., a 3x3 neighborhood



3.1 Background
1x1 - Intensity Transformation or Point Processing

- The simplest spatial domain operations occur when the neighbourhood is simply the pixel itself
- In this case T is referred to as a gray level transformation function or a point processing operation
- Point processing operations take the form

$$g(x,y) = T[f(x,y)] \longrightarrow \boxed{s = T(r)}$$

(Intensity Transformation Function)

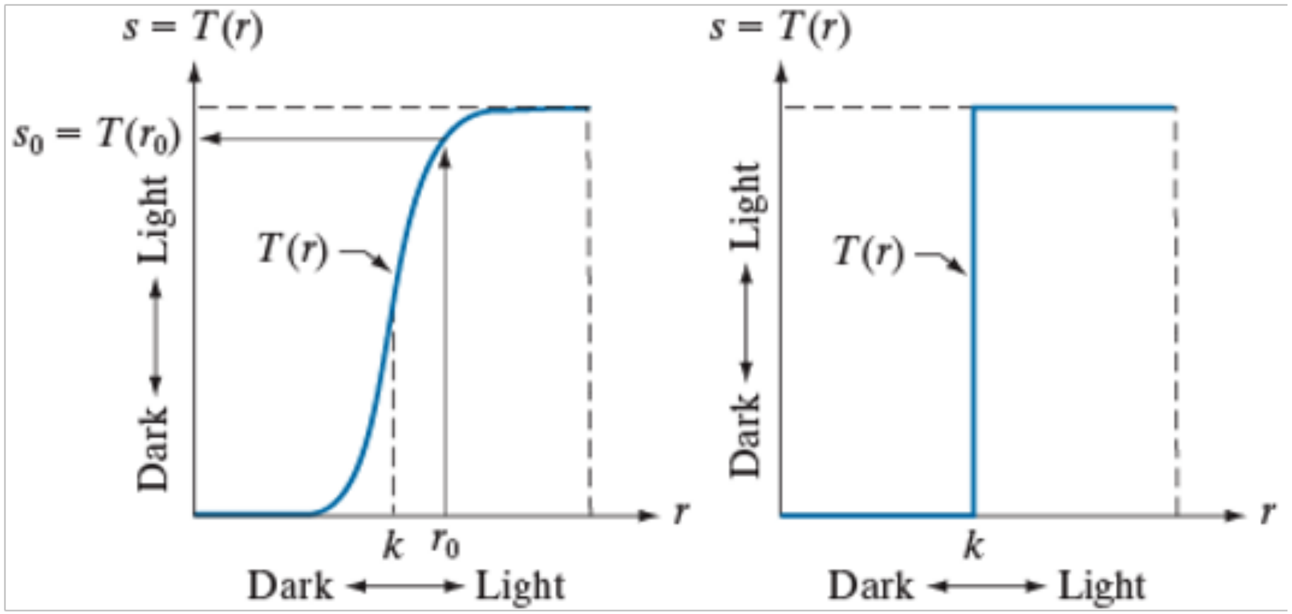
where s refers to the processed image pixel value and r refers to the original image pixel value.

credit of this slide: C. Nikou

3.1 Background
- Intensity Transformation or Point Processing

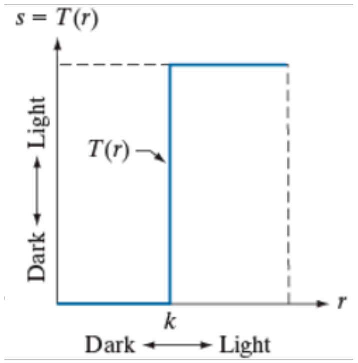
$$g(x,y) = T[f(x,y)] \longrightarrow s = T(r)$$

(Intensity Transformation Function)



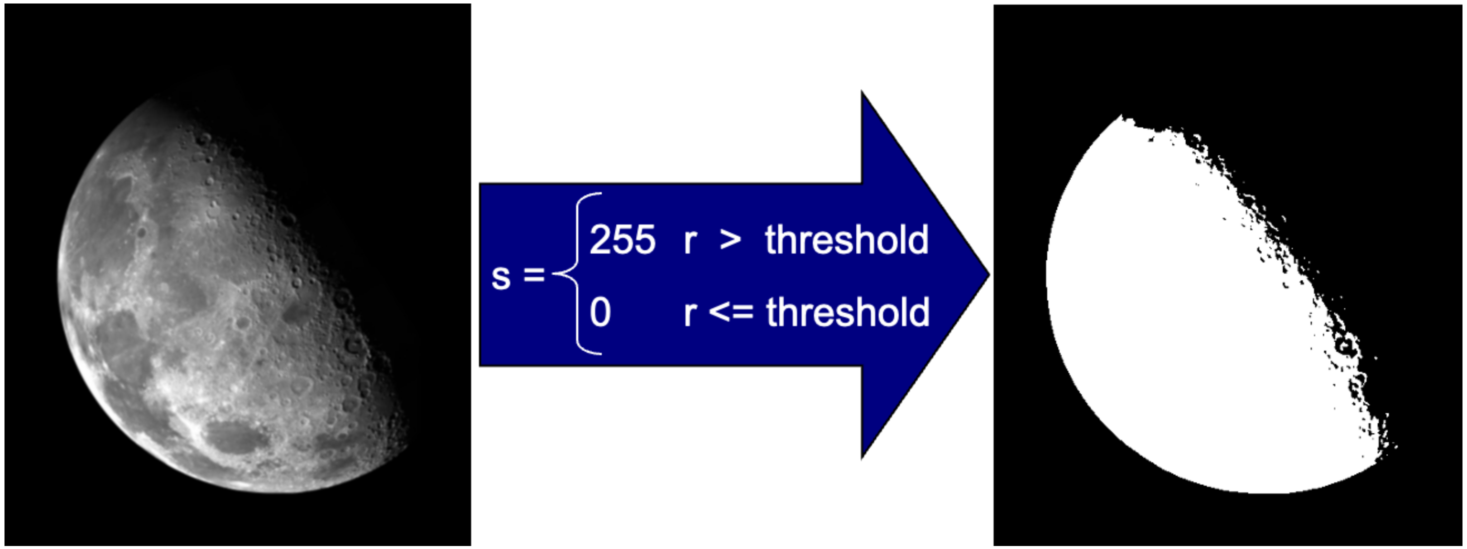
Contrast Stretching **Thresholding**

3.1 Background
- Example of Thresholding



Thresholding

Thresholding transformations are particularly useful for segmentation in which we want to isolate an object of interest from a background



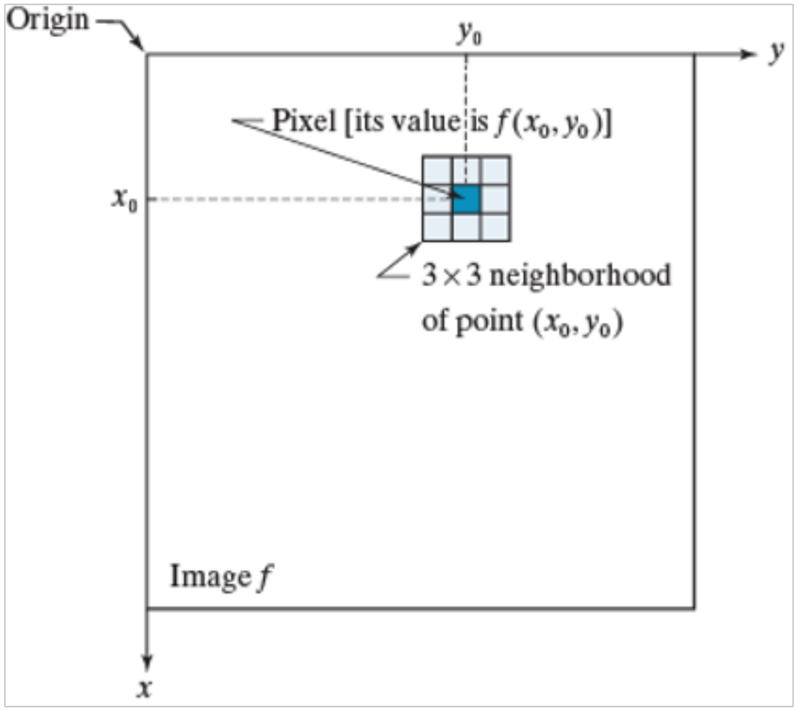
credit of this slide: C. Nikou

3.1 Background

Point Processing (3.2 - 3.3)

vs.

Mask Processing (3.4 – 3.6)



Mask Processing (Spatial Filtering)

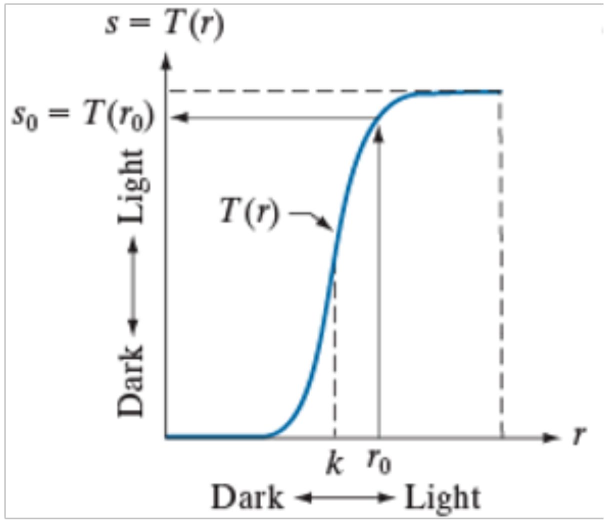
Mask (filter, kernel, template, window) -- a small 2D array in which the values of the mask coefficients determine the nature of the process, e.g., smoothing or sharpening

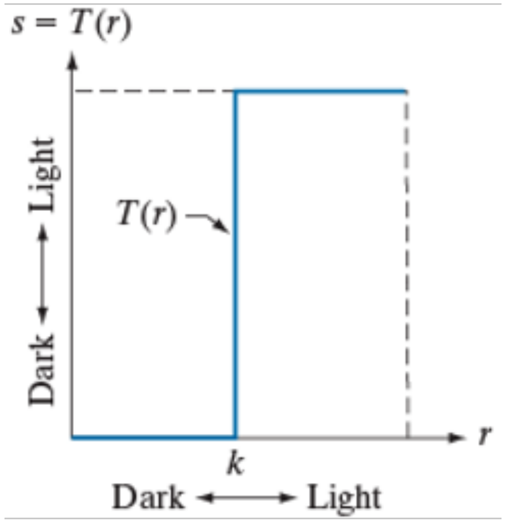
credit of this slide: Y. P. Hung

3.2 Some Basic Intensity Transformation Functions

- Image Negatives
- Log Transformations
- Power-Law Transformations
- Piecewise-Linear Transformations
  - Contrast Stretching
  - Intensity-Level Slicing
  - Bit-Plane Slicing

Implemented via table lookups
-- 256 entries

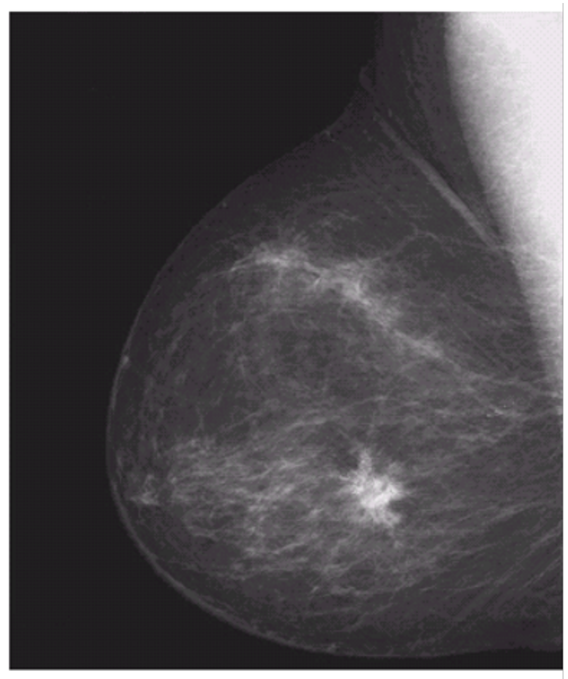




3.2 Some Basic Intensity Transformation Functions
- Image Negatives

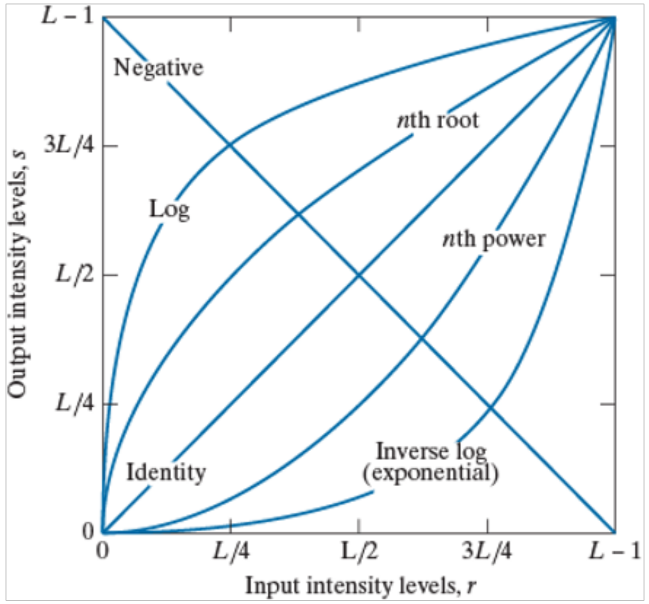
Negative images are useful for enhancing white or gray detail embedded in dark regions of an image

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



Original Image

Negative Image

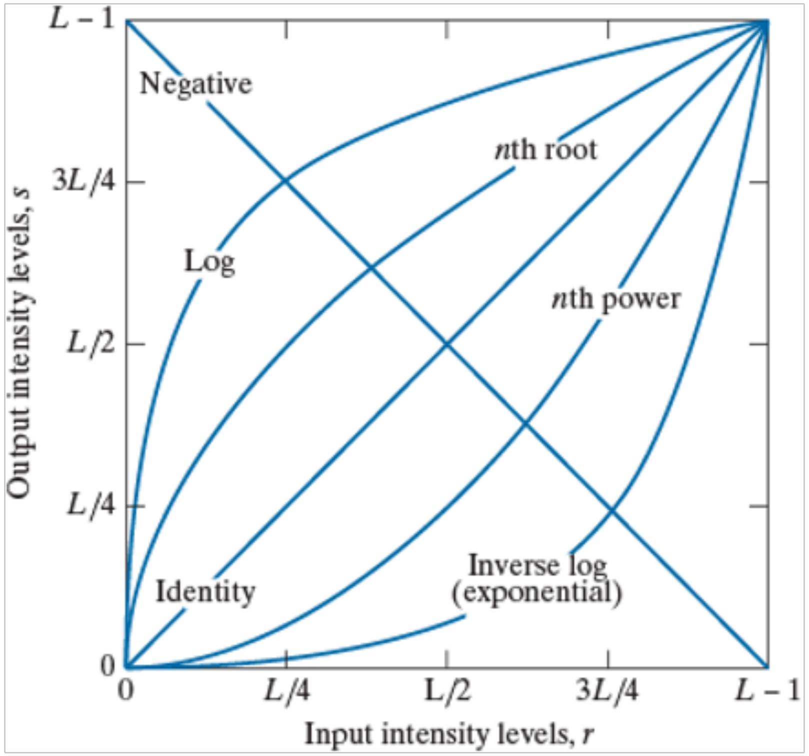


credit of this slide: C. Nikou

3.2 Some Basic Intensity Transformation Functions - Log Transformations

- The log transformation maps a narrow range of low input grey level values into a wider range of output values
- The inverse log transformation performs the opposite transformation

$$s = c \log (1 + r)$$



credit of this slide: C. Nikou

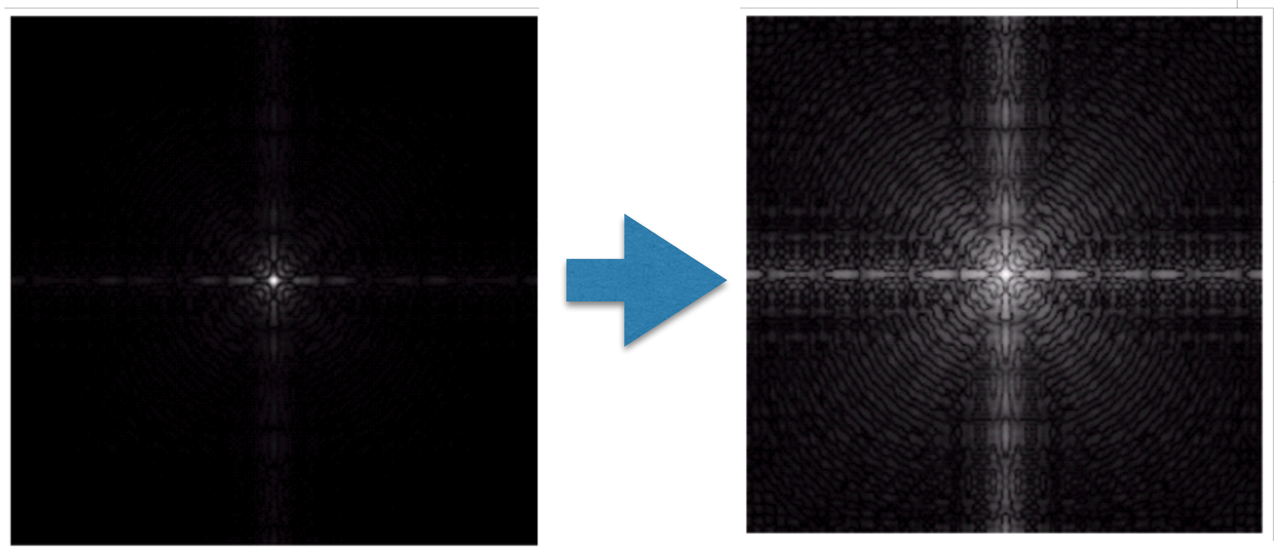
3.2 Some Basic Intensity Transformation Functions - Log Transformations

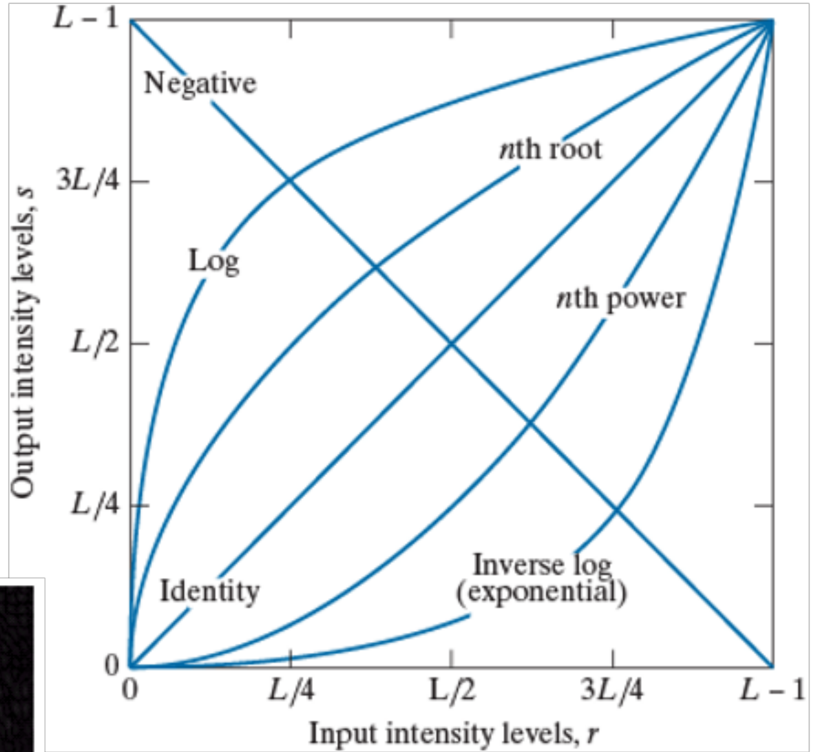
$$s = c \log (1 + r)$$

useful for compressing large dynamic range

Fourier Spectrum

$0 \sim 1.5 \times 10^6 \rightarrow 0 \sim 6.2$





Not as versatile as power-law

Most of the Fourier Spectra seen in the IP publications, including this book, have been scaled in this manner

credit of this slide: Y. P. Hung

3.2 Some Basic Intensity Transformation Functions - Power-Law (Gamma) Transformations

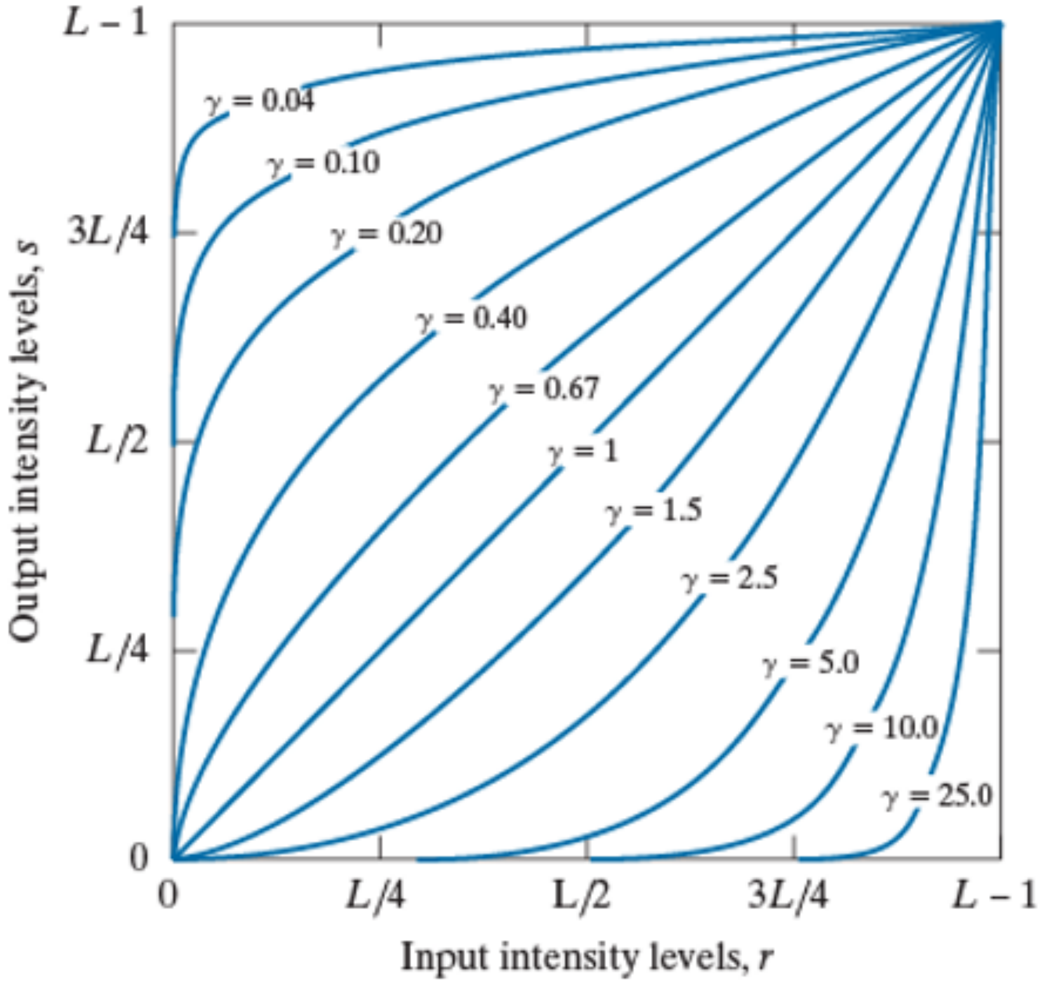
gamma

$$s = c \times r^{\gamma}$$

Another version

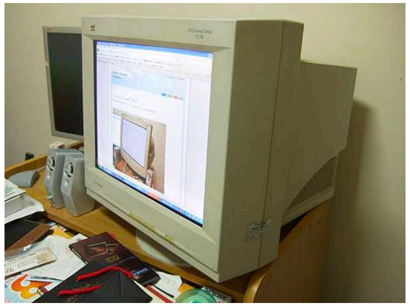
$$s = c \times (r + \varepsilon)^{\gamma}$$

offset: a measurable output when the input is zero



3.2 Some Basic Intensity Transformation Functions - Power-Law (Gamma) Transformations

$s = c \times r^{\gamma}$ Gamma Correction (Gamma Encoding)



- A variety of devices used for image capture, printing, and display respond according to a power law.
- For example, CRT: 1.8 ~ 2.5,
  - → Tend to produce images that are darker than intended.
  - → Need to preprocess the input image before inputting it into the monitor by performing **gamma correction**.

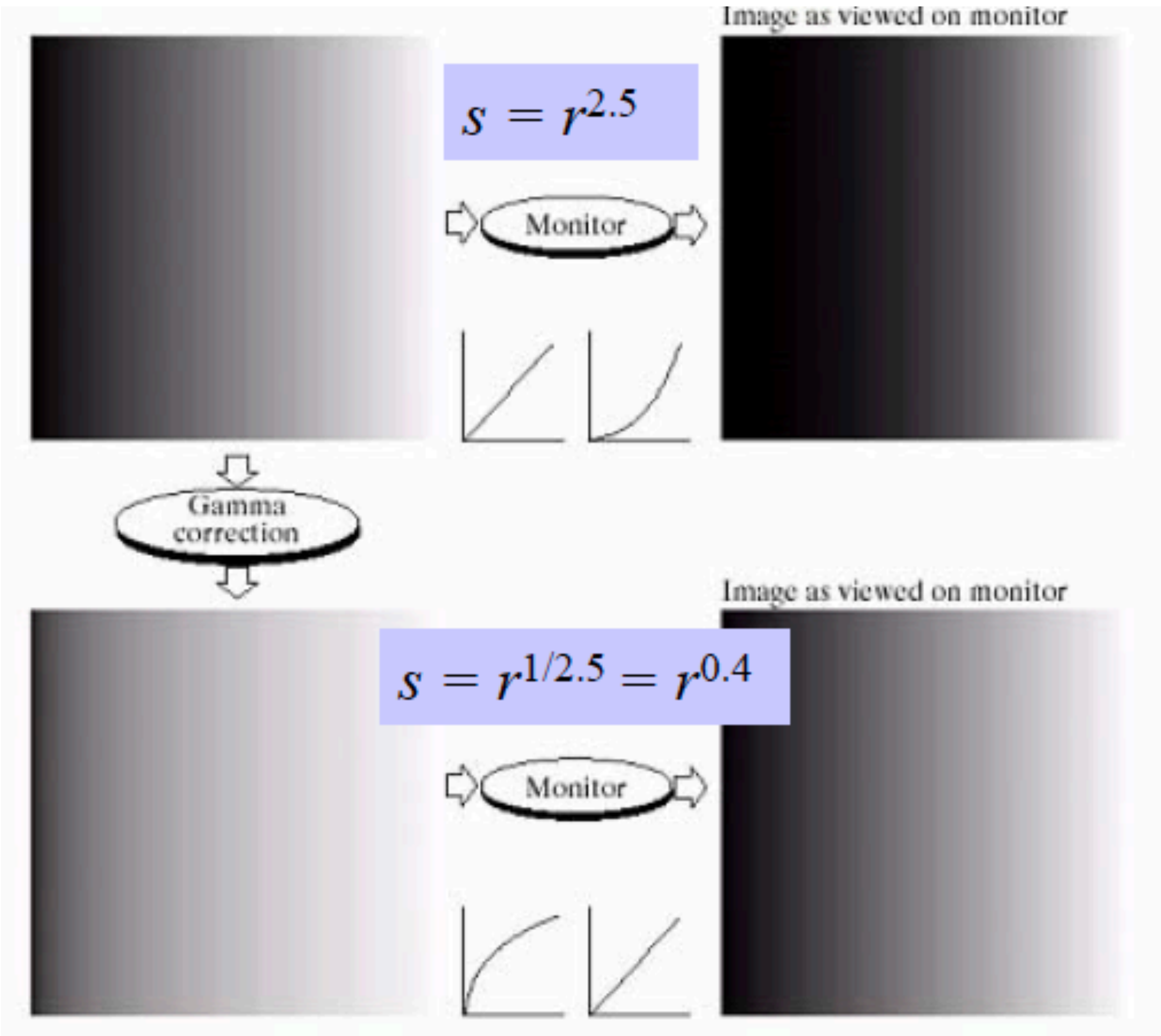
- For example, a Game Boy Advance display has a gamma between 3 and 4 depending on lighting conditions
- In LCD displays such as those on laptop computers, the relation between the signal voltage V_S and the intensity I is very nonlinear and cannot be described with gamma value. However, such displays apply a correction onto the signal voltage in order to approximately get a standard γ =2.5 behaviour.

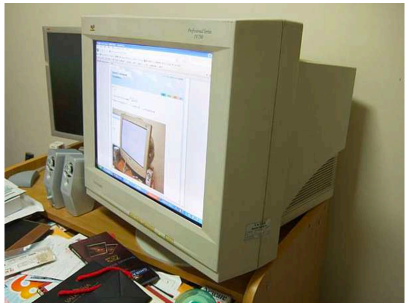
credit of this slide: Y. P. Hung

3.2 Some Basic Intensity Transformation Functions - Power-Law (Gamma) Transformations

$$s = c \times r^{\gamma}$$

Gamma Correction (Gamma Encoding)





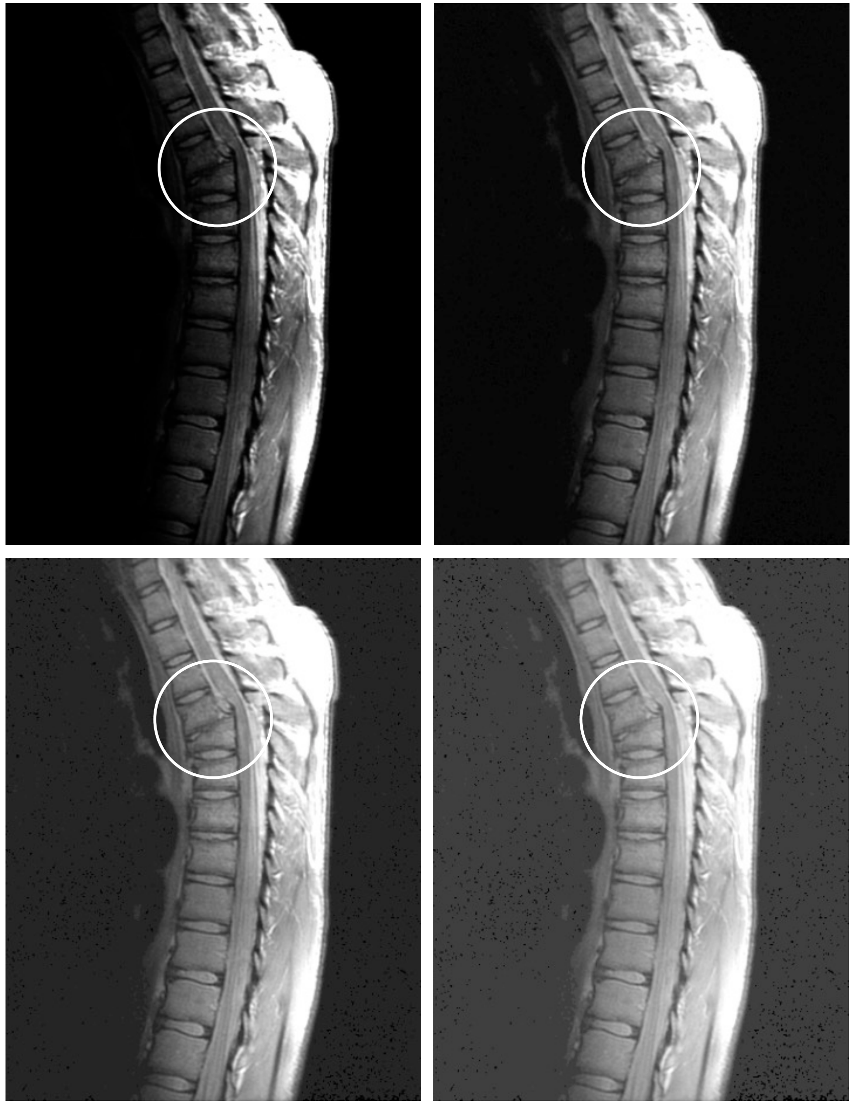
monitor with a gamma of 2.5

3.2 Some Basic Intensity Transformation Functions - Power-Law (Gamma) Transformations

$$s = c \times r^{\gamma}$$

Other examples of Gamma Correction

- MRI of a human upper thoracic spine with a fracture dislocation
- Different gamma highlight different detail



$\gamma = 0.6$

$\gamma = 0.4$

$\gamma = 0.3$

3.2 Some Basic Intensity Transformation Functions
- Power-Law (Gamma) Transformations

$s = c \times r^{\gamma}$

Other examples of Gamma Correction

- Aerial image
- Different gamma highlight different detail

$\gamma = 3.0$

$\gamma = 4.0$

$\gamma = 5.0$

3.2 Some Basic Intensity Transformation Functions
- Piecewise-Linear Transformations

Rather than using a well defined mathematical function we can use arbitrary user-defined transforms

- Contrast Stretching
- Intensity-Level Slicing
- Bit-Plane Slicing

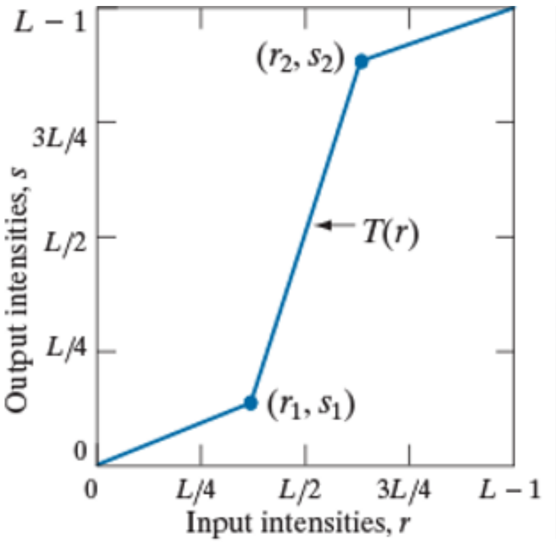
3.2 Some Basic Intensity Transformation Functions - Piecewise-Linear Transformations

- Contrast Stretching

Assume $r_2 \geq r_1$ and $s_2 \geq s_1$

→ single-valued function, and monotonically increasing

→ prevent intensity artifacts



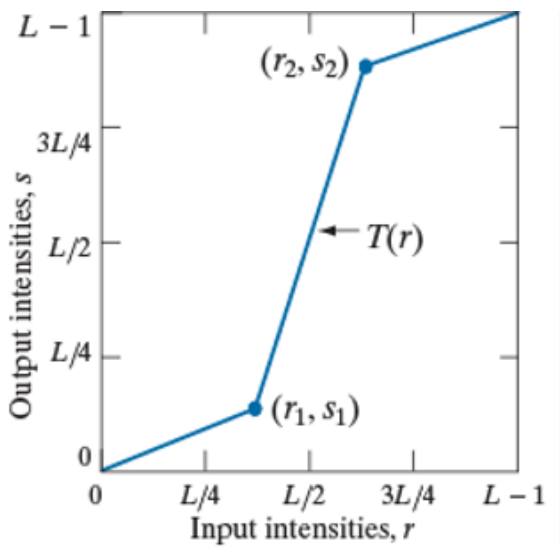
3.2 Some Basic Intensity Transformation Functions - Piecewise-Linear Transformations

- Contrast Stretching

Assume $r_2 \geq r_1$ and $s_2 \geq s_1$

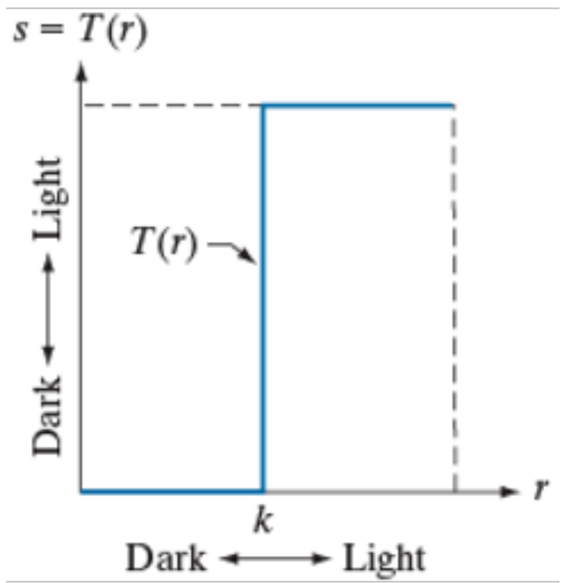
→ single-valued function, and monotonically increasing

→ prevent intensity artifacts



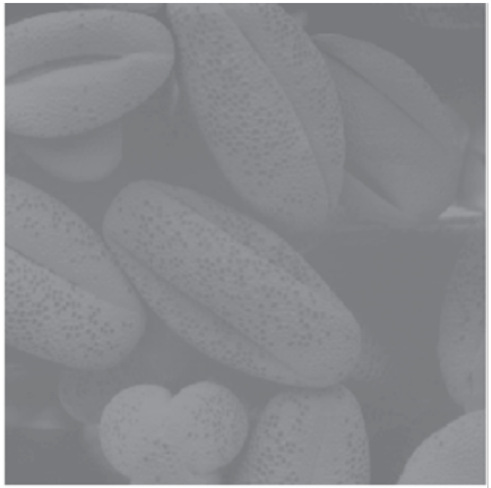
Some special cases:

- If $r_1 = s_1$ and $r_2 = s_2$
  - produce no changes
- If $r_1 = r_2$, $s_1 = 0$, and $s_2 = L - 1$
  - Thresholding



3.2 Some Basic Intensity Transformation Functions
- Piecewise-Linear Transformations

- Contrast Stretching



放大700倍的花粉

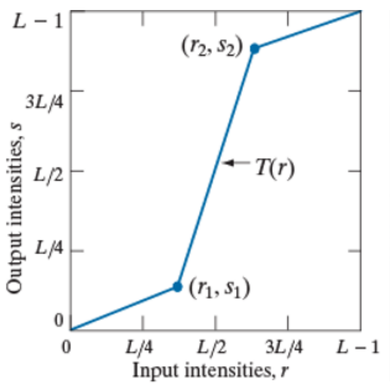
Full-Scale Contrast Stretch

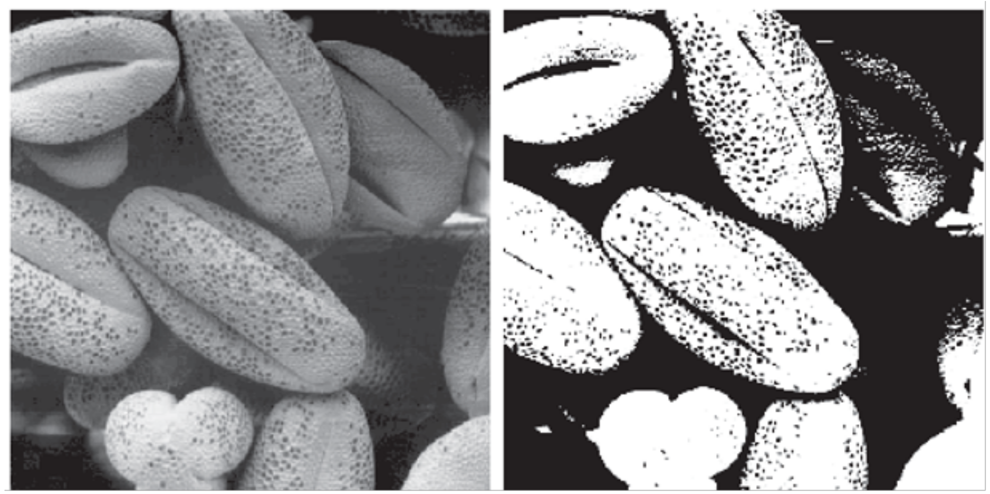
$$(r_1, s_1) = (r_{min}, 0)$$
$$(r_2, s_2) = (r_{max}, L-1)$$

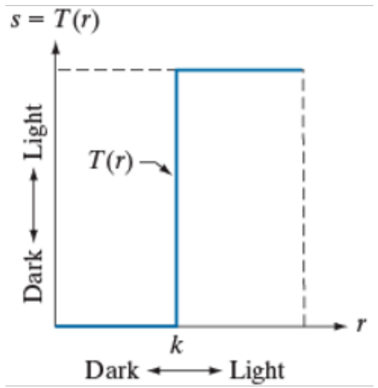
Thresholding

m is the mean intensity level

$$(r_1, s_1) = (m, 0)$$
$$(r_2, s_2) = (m, L-1)$$

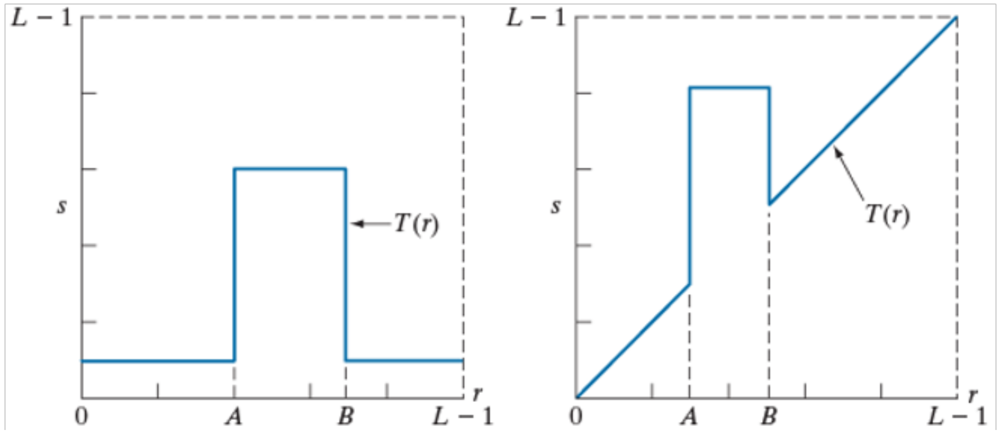




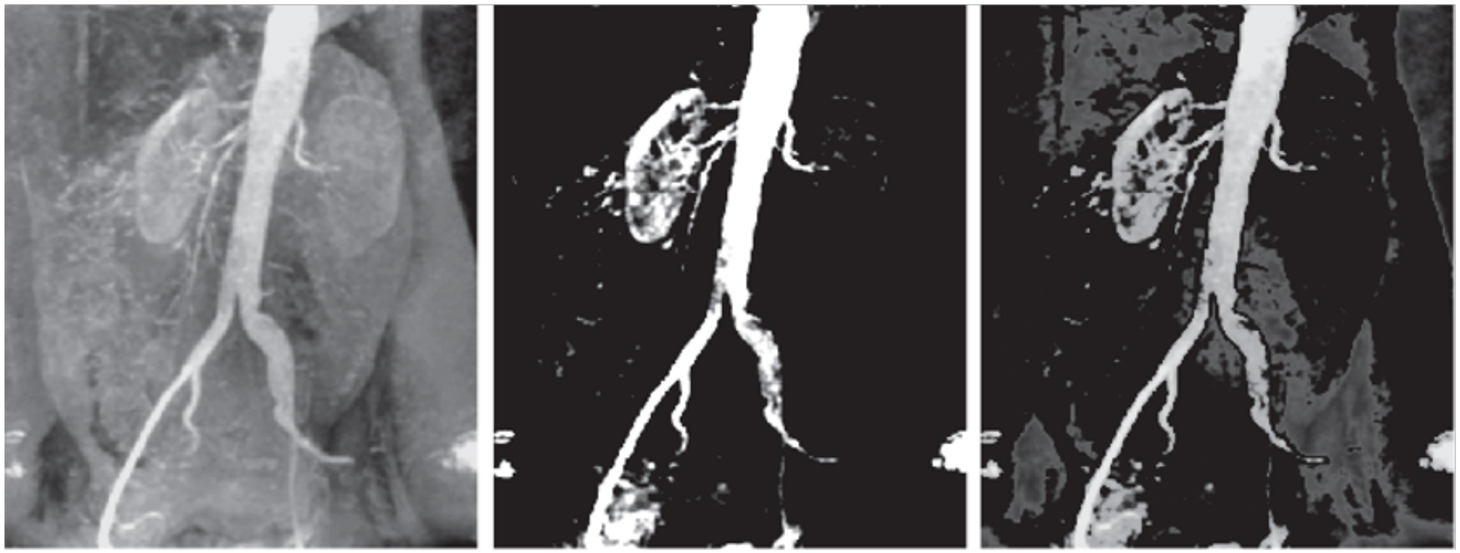


3.2 Some Basic Intensity Transformation Functions - Piecewise-Linear Transformations

- Intensity-Level Slicing



電腦斷層血管攝影術

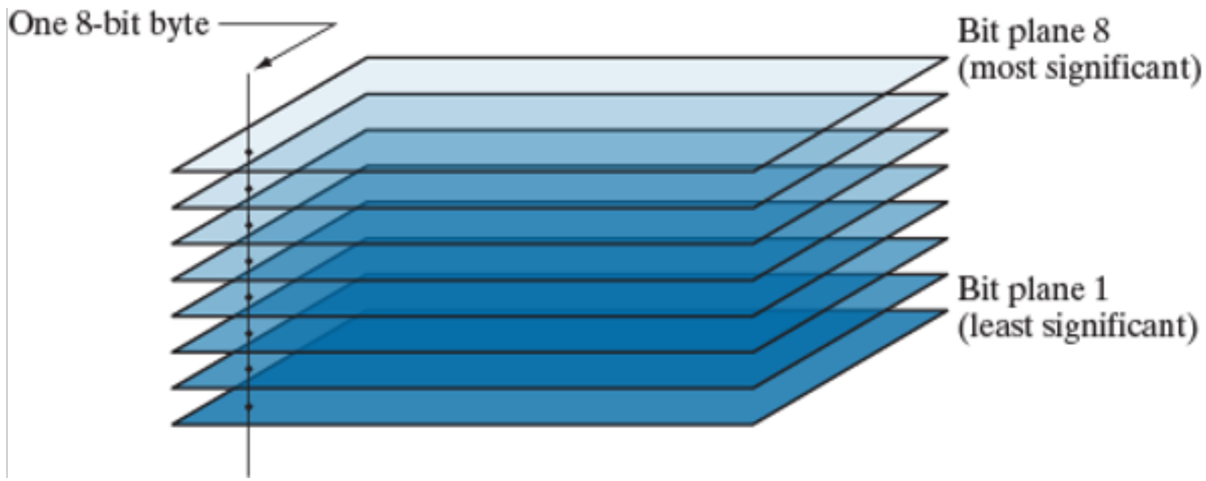


3.2 Some Basic Intensity Transformation Functions - Piecewise-Linear Transformations

- Bit-Plane Slicing

Often by isolating particular bits of the pixel values in an image we can highlight interesting aspects of that image

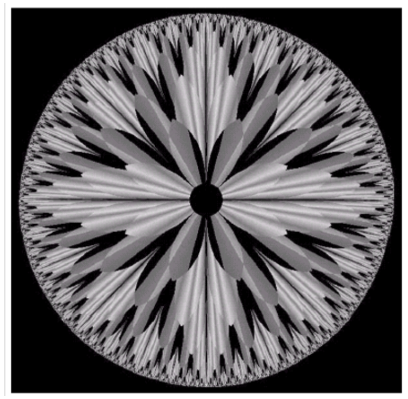
- Higher-order bits usually contain most of the significant visual information
- Lower-order bits contain subtle details

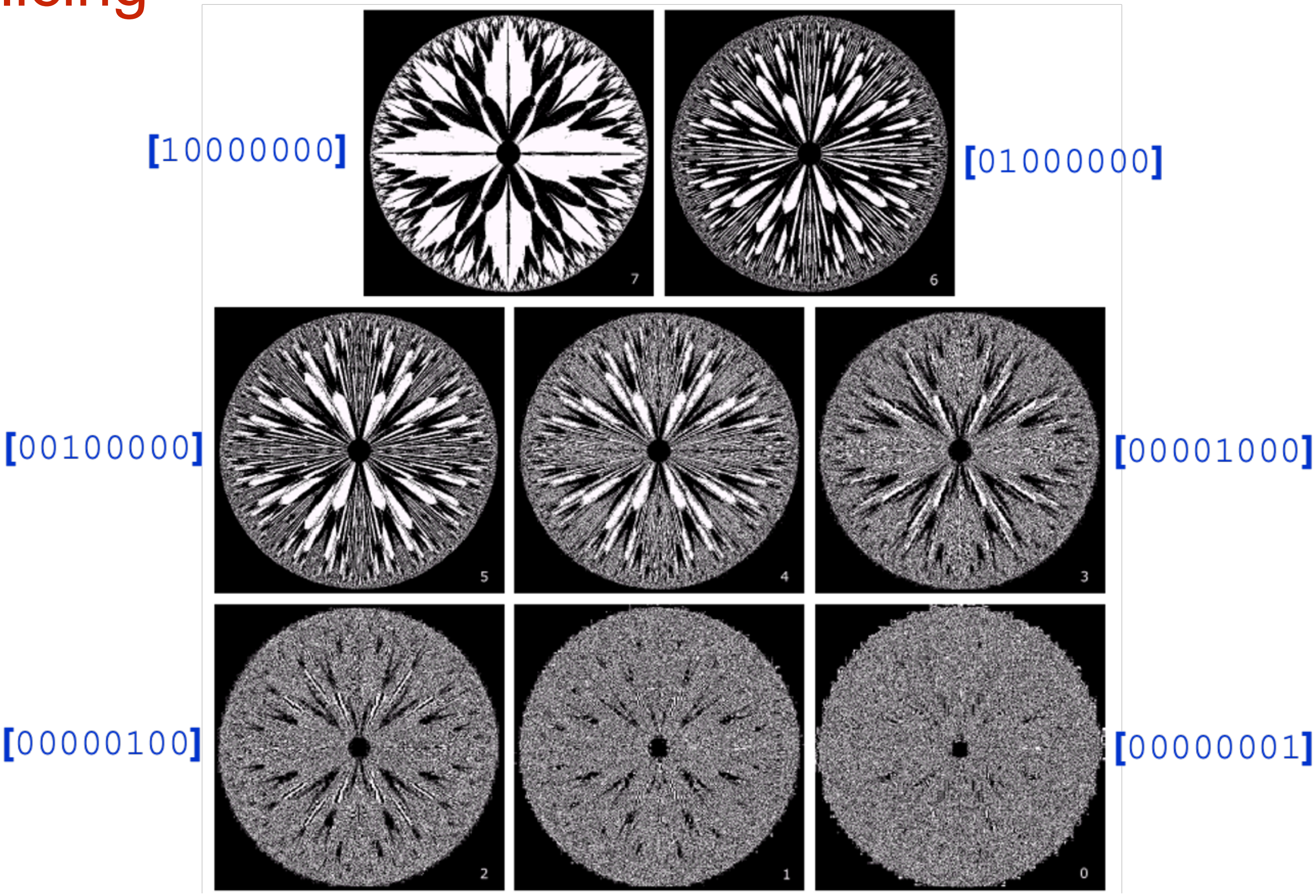


credit of this slide: C. Nikou

3.2 Some Basic Intensity Transformation Functions
- Piecewise-Linear Transformations

- Bit-Plane Slicing

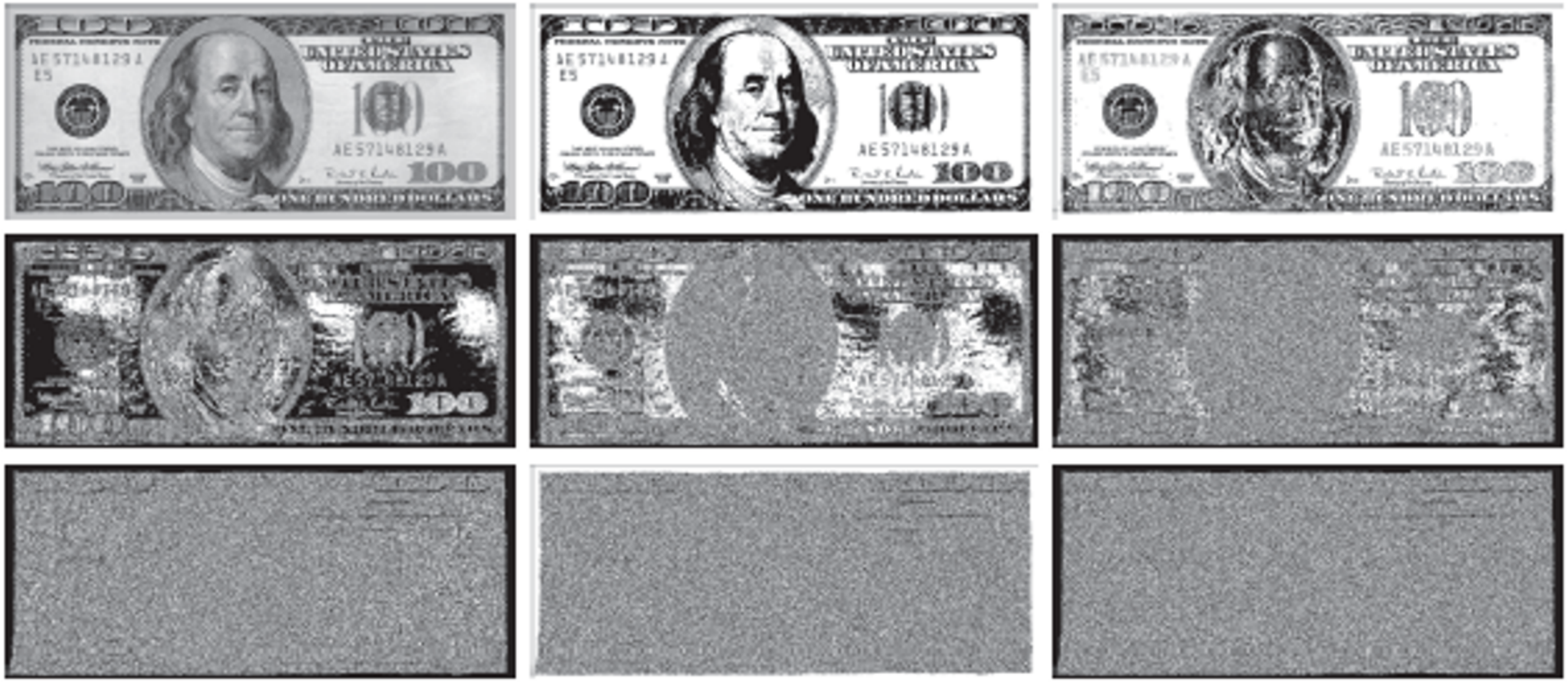




credit of this slide: C. Nikou

3.2 Some Basic Intensity Transformation Functions - Piecewise-Linear Transformations

- Bit-Plane Slicing



a	b	c
d	e	f
g	h	i

FIGURE 3.14 (a) An 8-bit gray-scale image of size 550×1192 pixels. (b) through (i) Bit planes 8 through 1, with bit plane 1 corresponding to the least significant bit. Each bit plane is a binary image..

3.2 Some Basic Intensity Transformation Functions - Piecewise-Linear Transformations

- Bit-Plane Slicing

a b c

FIGURE 3.15 Image reconstructed from bit planes: (a) 8 and 7; (b) 8, 7, and 6; (c) 8, 7, 6, and 5.

Useful for compression.

Reconstruction is obtained by:

$$I(i,j) = \sum_{n=1}^{N} 2^{n-1} I_n(i,j)$$

credit of this slide: C. Nikou

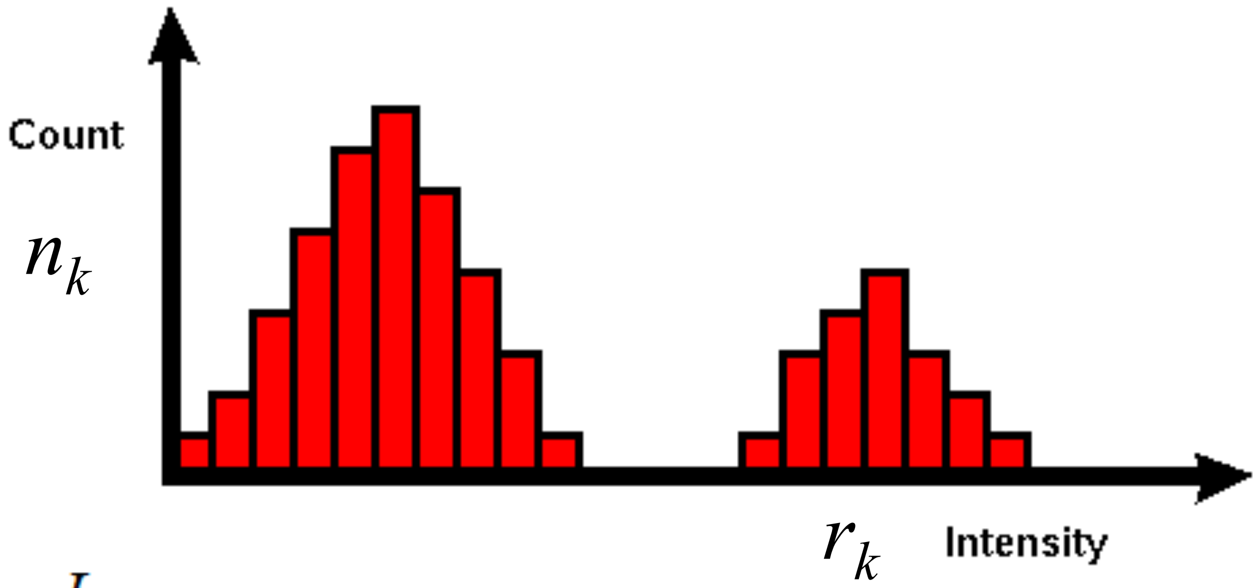
3.3 Histogram Processing - Histogram (直方圖)

- Histogram of a digital image is a distribution function

$$h(r_k) = n_k$$

  - where r_k is the kth gray level

    and n_k is the number of pixels having gray level r_k

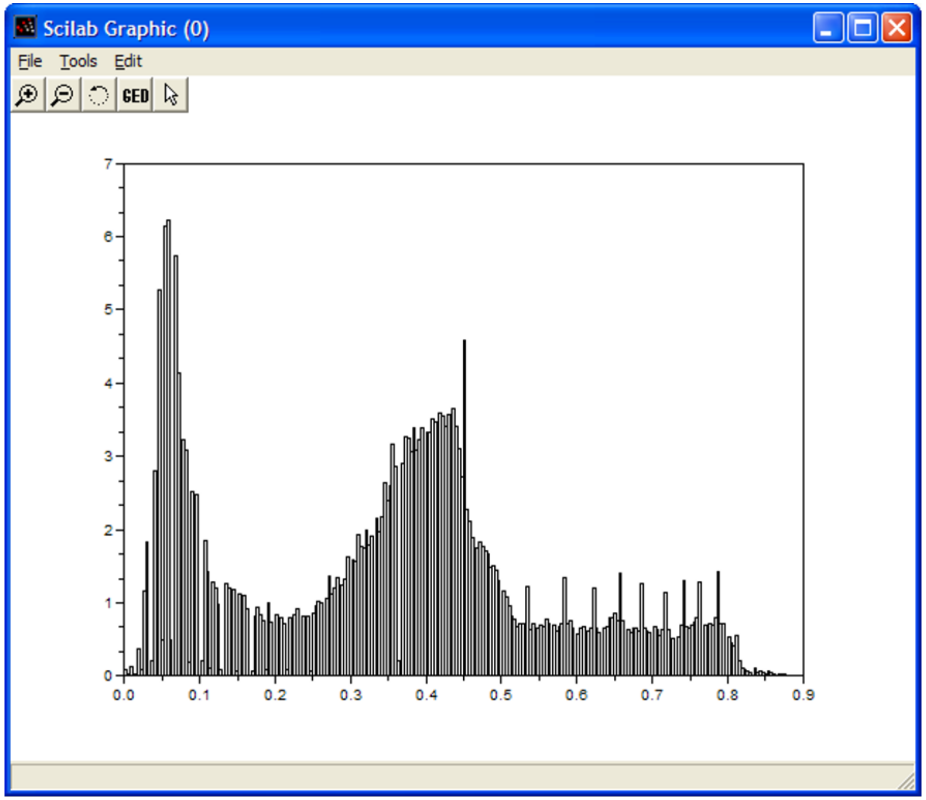


Normalized histogram

$$p(r_k) = n_k / n \ , \quad k = 1, \ \ldots, \ L$$

credit of this slide: Y. P. Hung

3.3 Histogram Processing
- Histogram (直方圖)



credit of this slide: C. Nikou

3.3 Histogram Processing
- Histogram (直方圖)

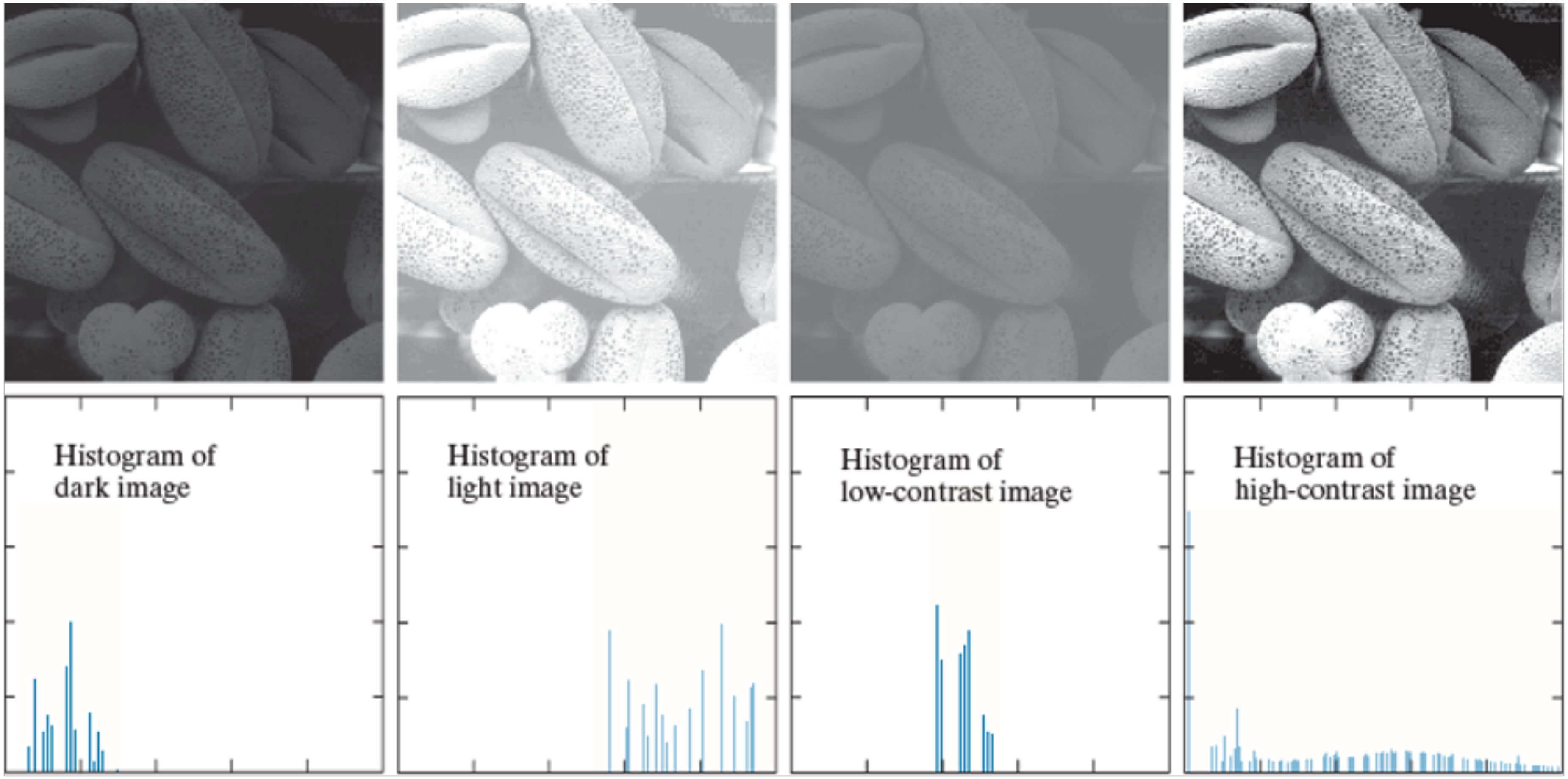
- Histogram is widely used in computer vision, because it can provide a scale, rotation, view-angle invariant descriptor to mention an object

Multi-camera object tracking

Single camera mean-shift object tracking

3.3 Histogram Processing
- Histogram (直方圖)

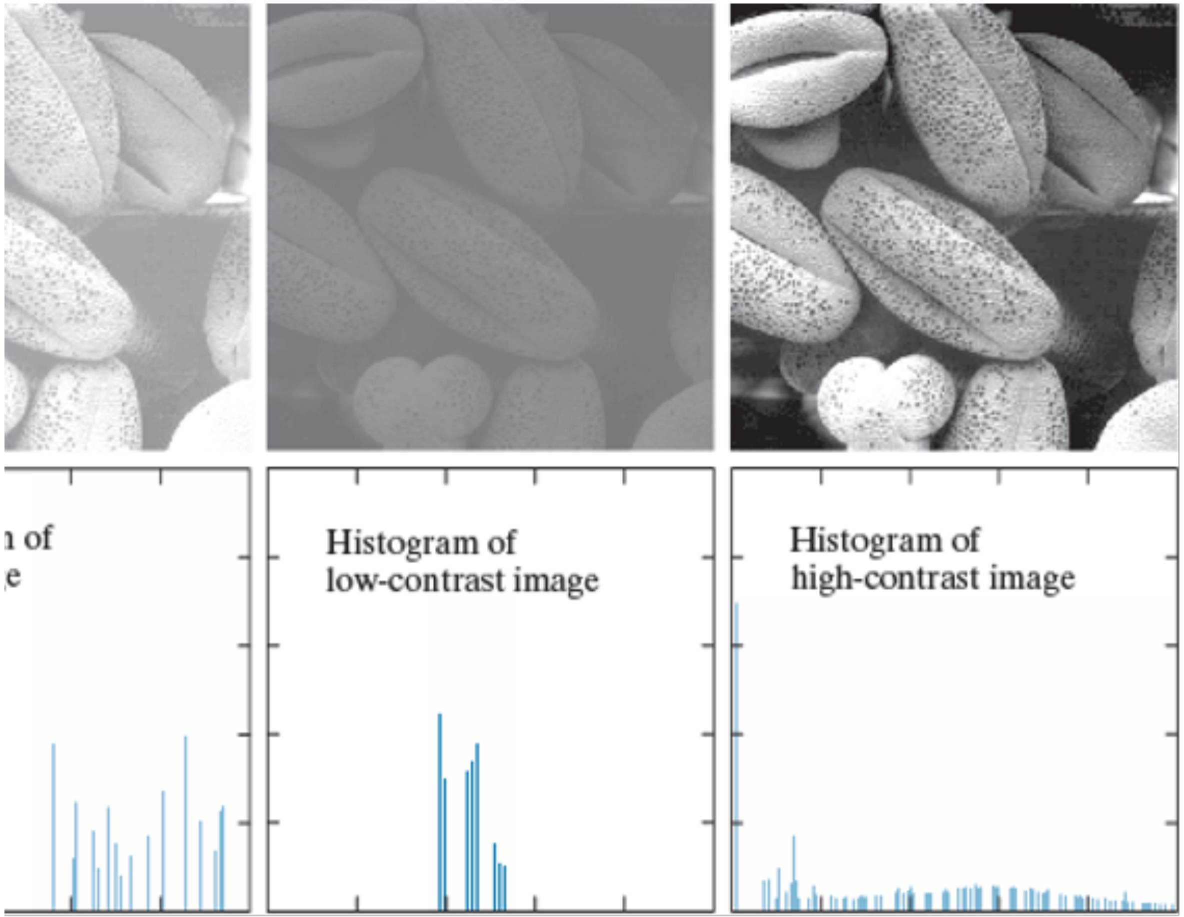
- Four image types and their corresponding histograms



Note that the high-contrast image usually has a more flat histogram

3.3 Histogram Processing
- Histogram (直方圖)

- Four image types and their corresponding histograms

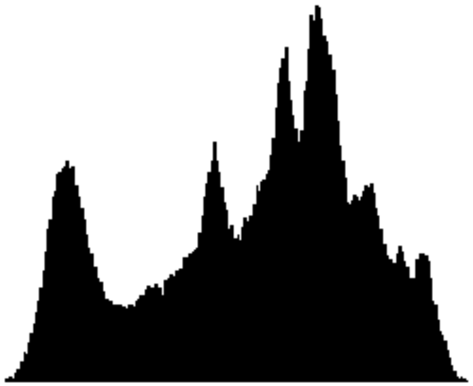


=> Histogram Equalization

Note that the high-contrast image usually has a more flat histogram

3.3 Histogram Processing
- Histogram Equalization

Image Enhancement



Histogram Equalization

To make histogram distributed uniformly

credit of this slide: Y. P. Hung

3.3 Histogram Processing - Histogram Equalization

At first, the **continuous case** will be studied:

- r is the intensity of the image in [0, L-1].
- The transformations $s = T(r)$:
  - $T(r)$ is strictly **monotonically increasing**.
  - $T(r)$ must satisfy:

$$0 \le T(r) \le L-1, \; for \;\; 0 \le r \le L-1$$

monotonically increasing: guarantees that ordering of the output intensity values will be the same as that of the input (avoids reversal of intensities)

3.3 Histogram Processing
- Histogram Equalization

At first, the **continuous case** will be studied:

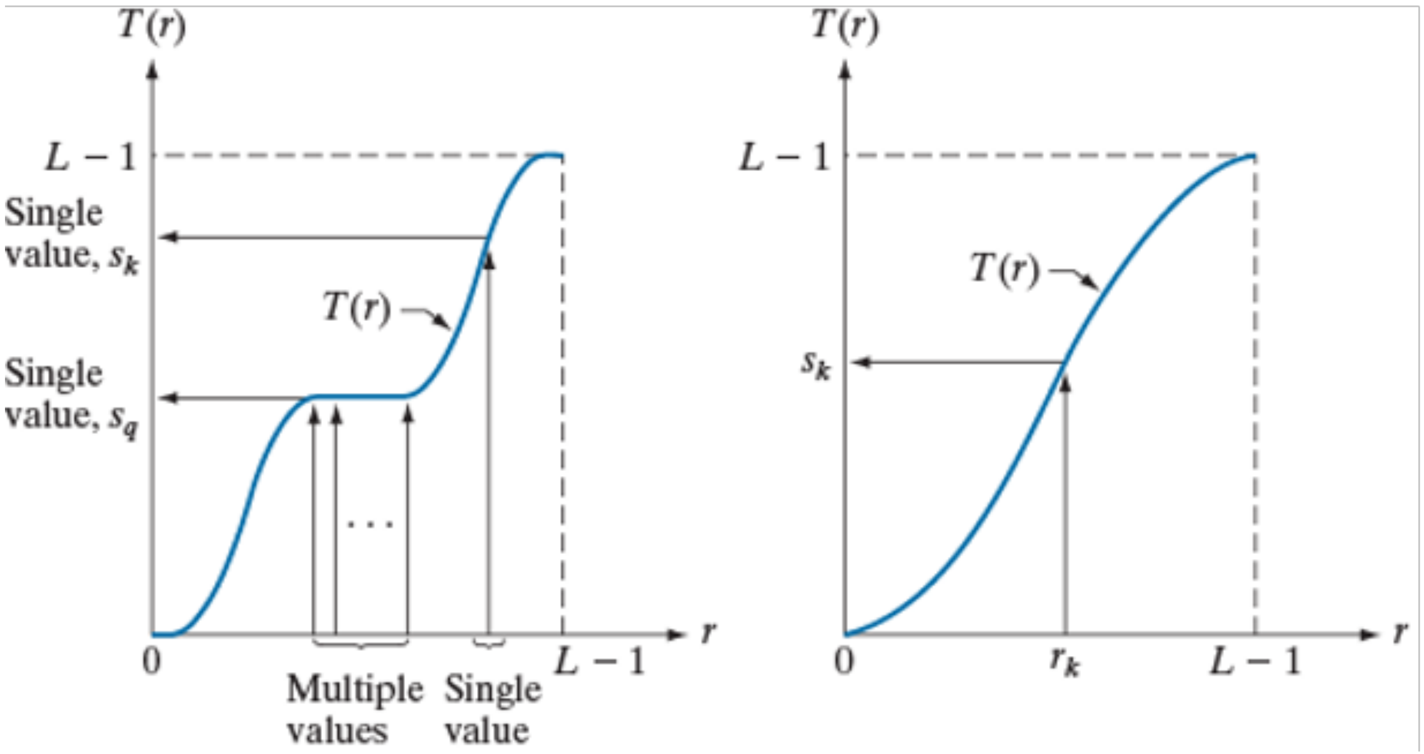
- r is the intensity of the image in [0, L-1].
- The transformations $s = T(r)$:
  - $T(r)$ is **strictly monotonically increasing**.
  - $T(r)$ must satisfy:

$$0 \leq T(r) \leq L-1, \ for \ \ 0 \leq r \leq L-1$$

strictly monotonically increasing: guarantees the mapping from s back to r will be ono-to-one

3.3 Histogram Processing - Histogram Equalization

strictly monotonically increasing: guarantees the mapping from *s* back to *r* will be ono-to-one



monotonically increasing   strictly monotonically increasing

3.3 Histogram Processing - Histogram Equalization

- We then can view intensities r and s as random variables and their histograms as probability density functions (PDFs) $p_r(r)$ and $p_s(s)$.

- A fundamental result from probability theory:
  - If $p_r(r)$ and $T(r)$ are known, and $T(r)$ is continuous and differentiable, then

$$p_s(s) = p_r(r)\left|\frac{dr}{ds}\right|$$

https://www.pbr-book.org/3ed-2018/Monte_Carlo_Integration/Transforming_between_Distributions

3.3 Histogram Processing - Histogram Equalization

$$p_s(s) = p_r(r)\left|\frac{dr}{ds}\right|$$

- The PDF of the output *s* is determined by the PDF of the input *r* and the transformation $T(r)$, which means we can determine the histogram of the output image

3.3 Histogram Processing - Histogram Equalization

- A transformation of particular importance in image processing is cumulative distribution function (CDF) of a random variable:

$$\boxed{s = T(r) = (L-1)\int_0^r p_r(w)\,dw}$$

- It satisfies the first condition as the area under the curve increases as r increases.
- It satisfies the second condition as for $r=L\text{-}1$, the integral evaluates to 1, thus the maximum $s=L\text{-}1$.

3.3 Histogram Processing - Histogram Equalization

- To find $p_s(s)$ we can compute

$$p_s(s) = p_r(r)\left|\frac{dr}{ds}\right|$$

$$s = T(r) = (L-1)\int_0^r p_r(w)\,dw$$

$$\frac{ds}{dr} = \frac{dT(r)}{dr}$$

$$= (L-1)\frac{d}{dr}\int_0^r p_r(w)\,dw$$

$$= (L-1)\,p_r(r)$$

3.3 Histogram Processing
- Histogram Equalization

Substituting this result: $$\frac{ds}{dr} = (L-1)\,p_r(r)$$

to $$p_s(s) = p_r(r)\left|\frac{dr}{ds}\right|$$

we have

$$p_s(s) = p_r(r)\left|\frac{1}{(L-1)\,p_r(r)}\right| = \frac{1}{L-1}, \quad 0 \le s \le L-1$$

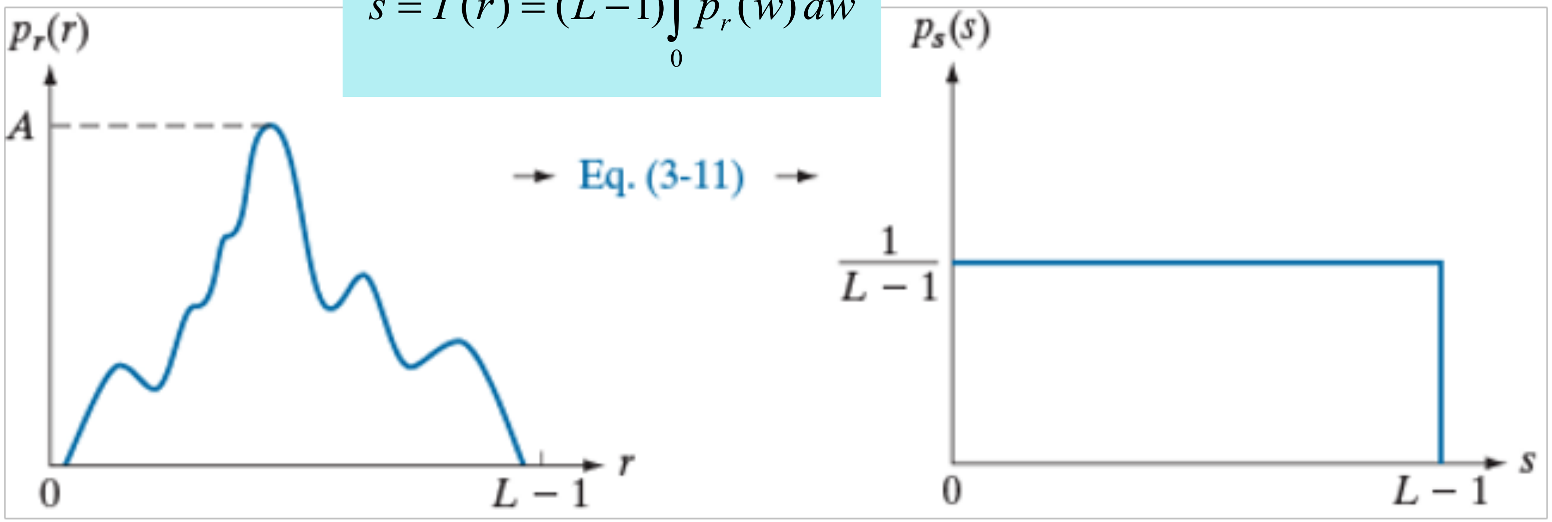
Uniform PDF

3.3 Histogram Processing
- Histogram Equalization

$$p_s(s) = p_r(r)\left|\frac{1}{(L-1)p_r(r)}\right| = \frac{1}{L-1}, \quad 0 \le s \le L-1$$

Uniform PDF

$$s = T(r) = (L-1)\int_0^r p_r(w)\,dw$$



3.3 Histogram Processing - Histogram Equalization

$$s = T(r) = (L-1)\int_0^r p_r(w)\,dw$$

- For discrete case, the formula of histogram equalization is

$$s_k = T(r_k) = (L-1)\sum_{j=0}^{k} p_r(r_j) = \frac{(L-1)}{MN}\sum_{j=0}^{k} n_j$$

where

- r_k: input intensity
- s_k: processed intensity
- n_j: the frequency of intensity j
- MN: the number of image pixels.

3.3 Histogram Processing
- Algorithm of Histogram Equalization

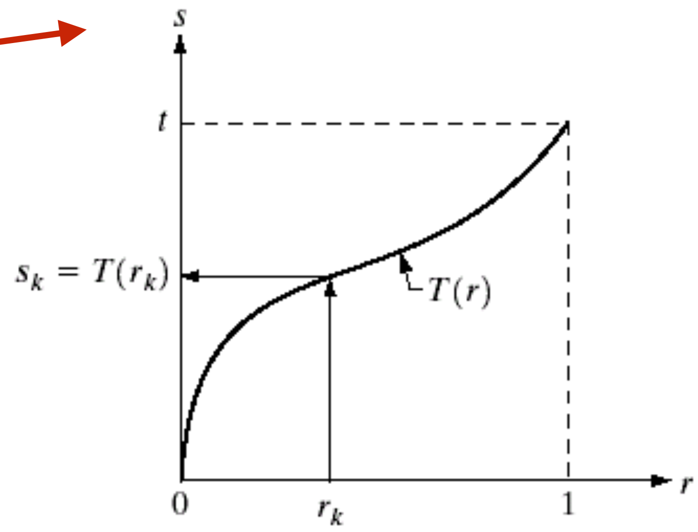
1. Compute the histogram of the input image:

$h(k) = \#\{(x,y)|f(x,y)=k\}$, *where* $k = 0$ *to* 255.

2. Compute the transformation function:

$$T(k) = 255 * \boxed{\sum_{j=0}^{k} \frac{h(j)}{n}}$$

Cumulative normalized histogram



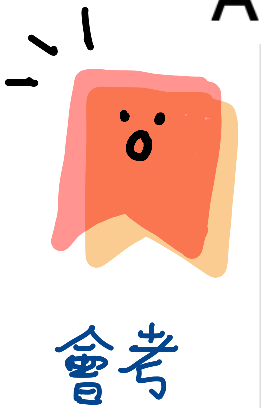
3. Transform the value of each pixel by

$$g(x,y)=T(f(x,y))$$

credit of this slide: Y. P. Hung

3.3 Histogram Processing
- Example of Histogram Equalization

A 3-bit 64x64 image has the following intensities:



r_k	n_k	$p_r(r_k) = n_k/MN$
$r_0 = 0$	790	0.19
$r_1 = 1$	1023	0.25
$r_2 = 2$	850	0.21
$r_3 = 3$	656	0.16
$r_4 = 4$	329	0.08
$r_5 = 5$	245	0.06
$r_6 = 6$	122	0.03
$r_7 = 7$	81	0.02

$$s_k = T(r_k) = (L-1)\sum_{j=0}^{k} p_r(r_j)$$

Applying histogram equalization:

$$s_0 = T(r_0) = 7\sum_{j=0}^{0} p_r(r_j) = 7p_r(r_0) = 1.33$$

$$s_1 = T(r_1) = 7\sum_{j=0}^{1} p_r(r_j) = 7p_r(r_0) + 7p_r(r_1) = 3.08$$

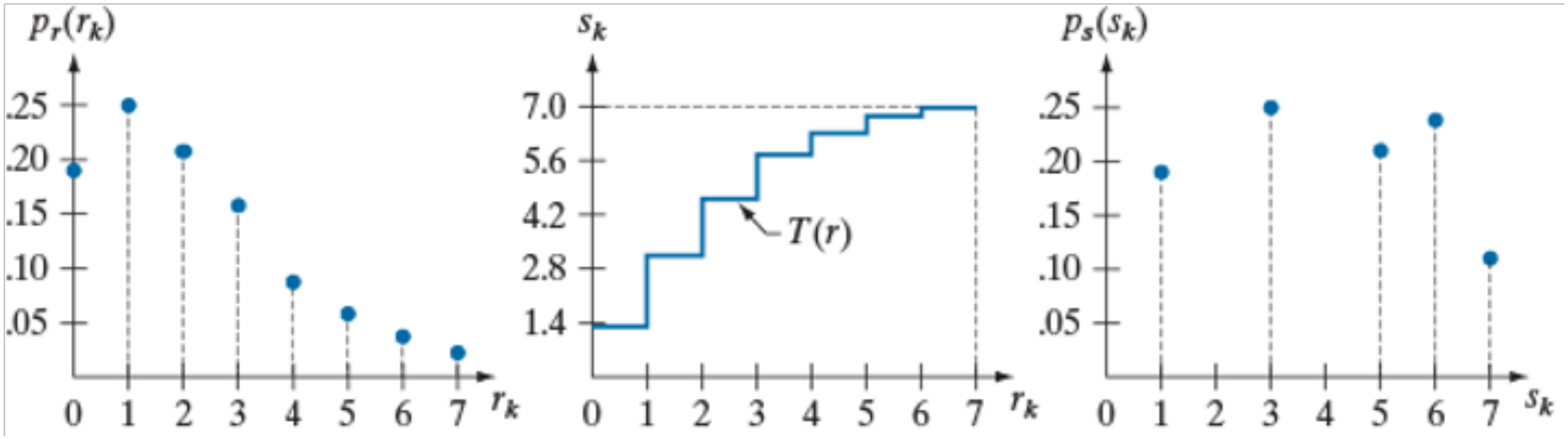
credit of this slide: C. Nikou

3.3 Histogram Processing
- Example of Histogram Equalization

Rounding to the nearest integer (四捨五入):

$$s_0 = 1.33 \rightarrow 1 \quad s_1 = 3.08 \rightarrow 3 \quad s_2 = 4.55 \rightarrow 5 \quad s_3 = 5.67 \rightarrow 6$$

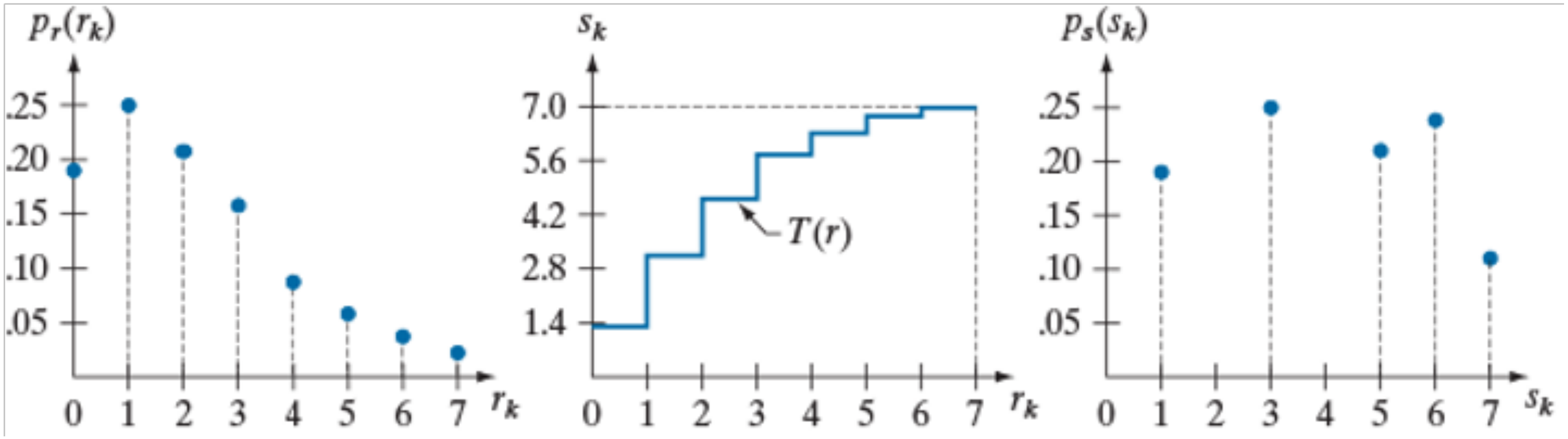
$$s_4 = 6.23 \rightarrow 6 \quad s_5 = 6.65 \rightarrow 7 \quad s_6 = 6.86 \rightarrow 7 \quad s_7 = 7.00 \rightarrow 7$$



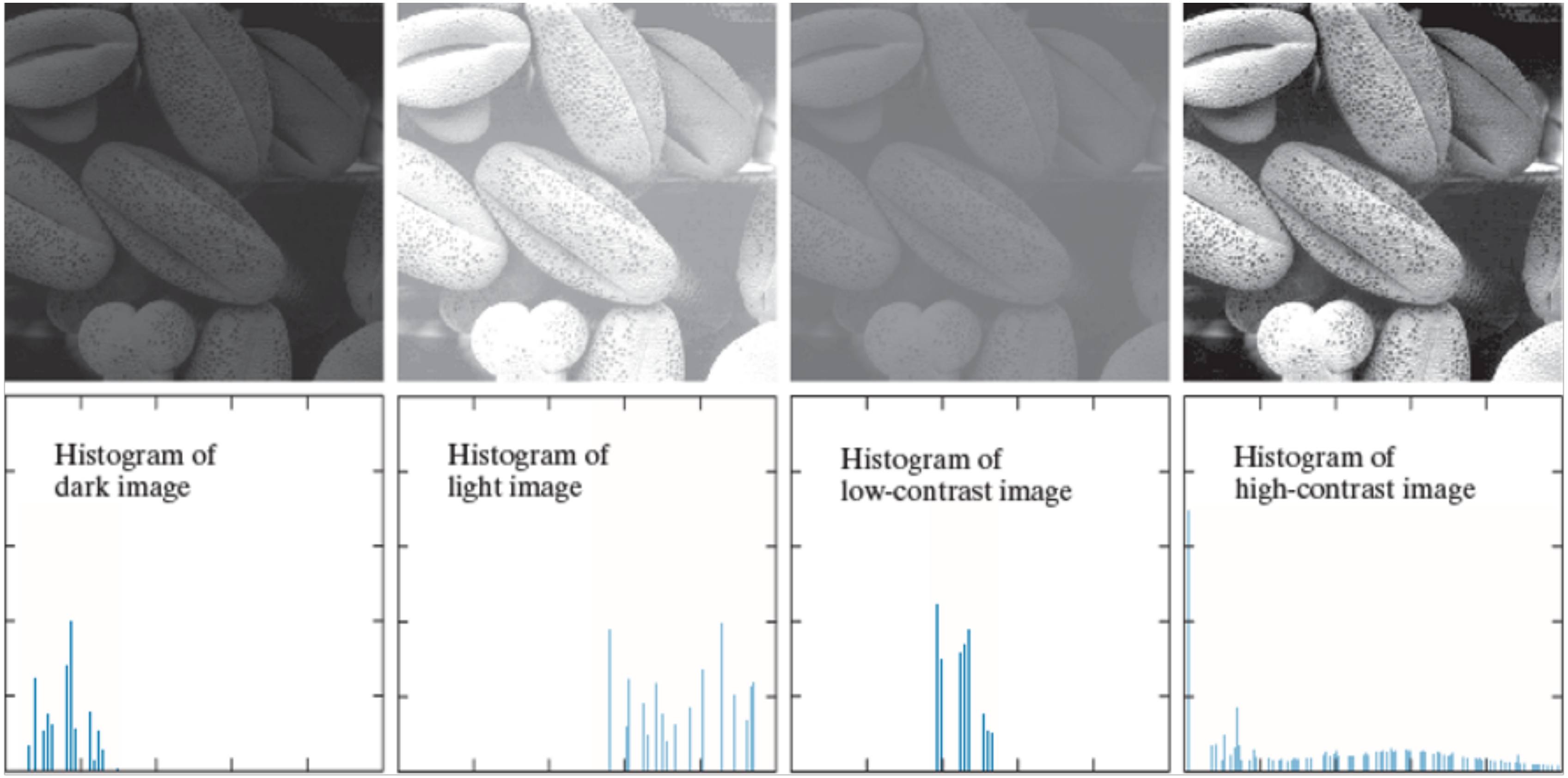
credit of this slide: C. Nikou

3.3 Histogram Processing - Example of Histogram Equalization

For discrete case, the resulting histogram will rarely be perfectly uniform. But the net result is contrast enhancement.



3.3 Histogram Processing
- Examples of Histogram Equalization



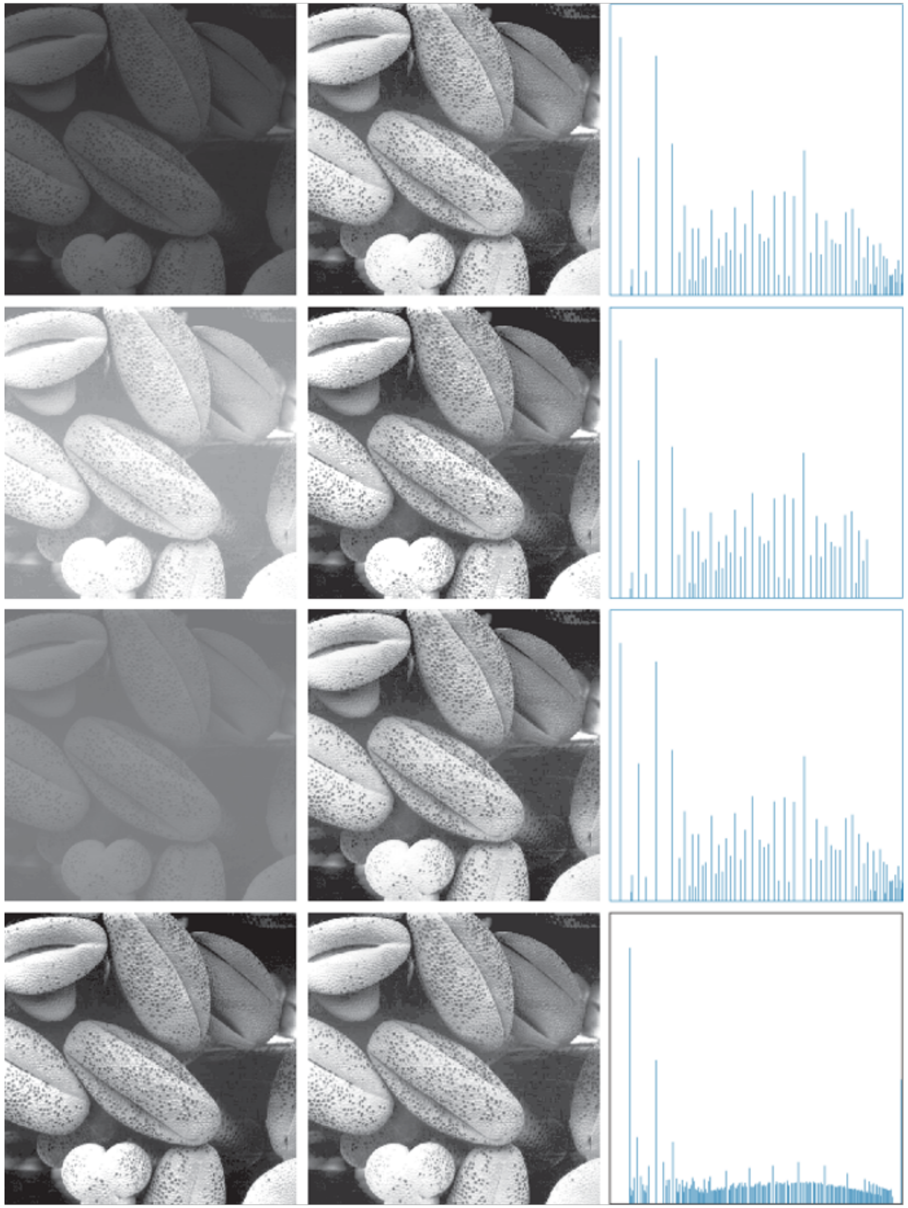
3.3 Histogram Processing
- Examples of Histogram Equalization

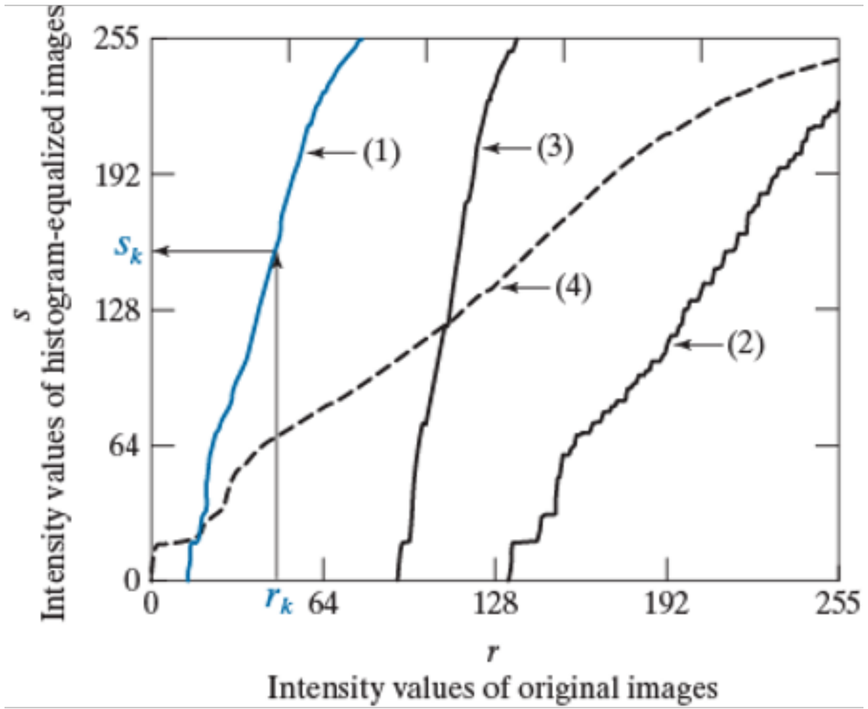
(1)

(2)

(3)

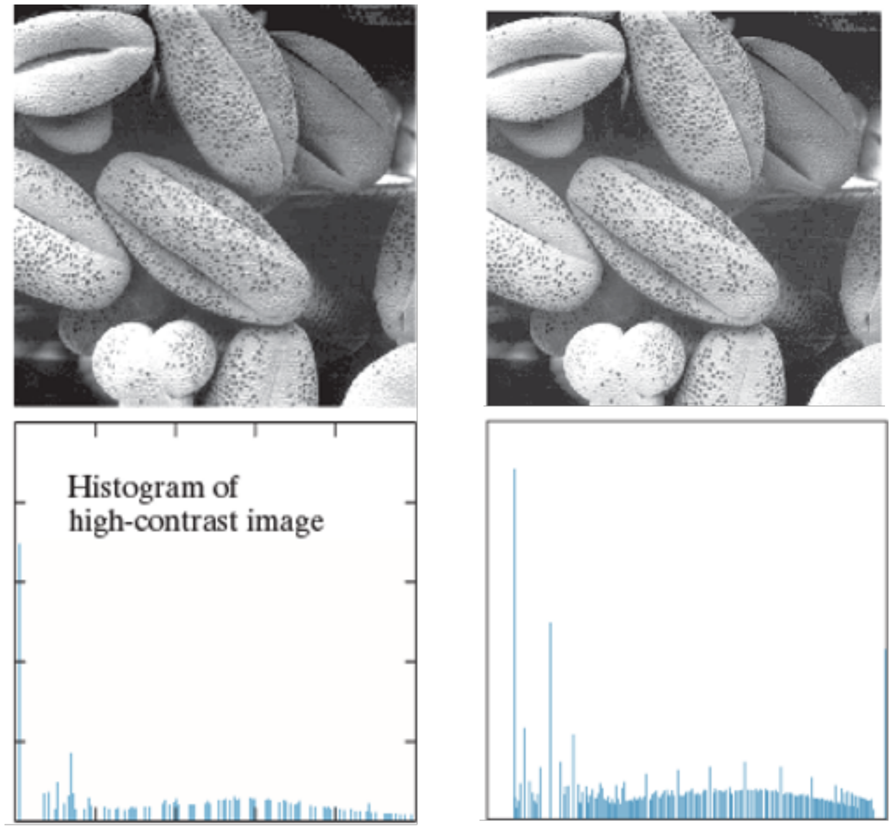
(4)





3.3 Histogram Processing
- Examples of Histogram Equalization

For discrete case, histogram bins are never reduced in amplitude, although they may increase if multiple gray levels map to the same value (thus destroying information)

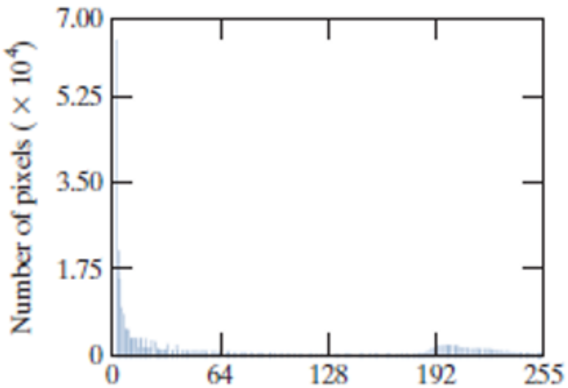


Histogram equalization does not always provide the desirable results

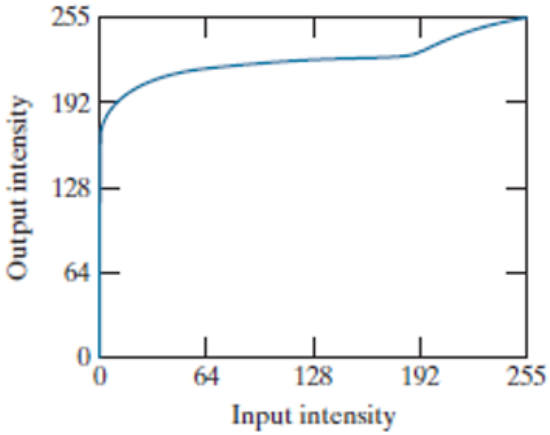
credit of this slide: Y. P. Hung

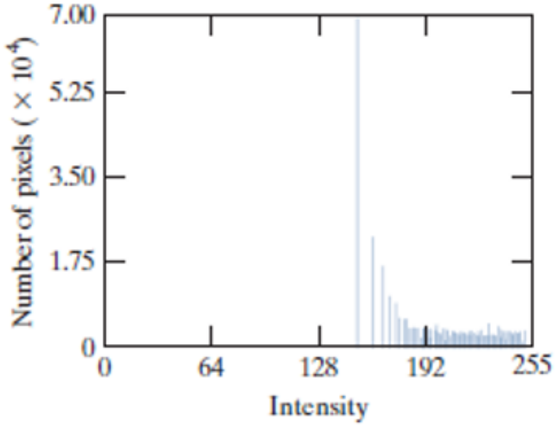
3.3 Histogram Processing - Histogram Specification

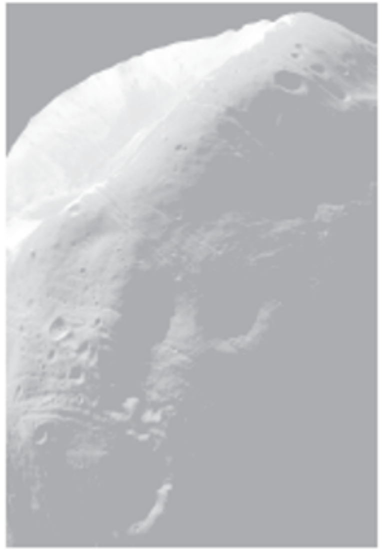
Histogram equalization does not always provide the desirable results



Original image







Histogram equalization

Want to transform an image into one that has a specific histogram

Histogram Specification

3.3 Histogram Processing
- Histogram Specification

Histogram Specification (Histogram Matching)

- **Problem statement**:
  – Given $p_r(r)$ from the image and the target histogram $p_z(z)$, estimate the transformation $z=T(r)$.

➡ The solution exploits **histogram equalization**

credit of this slide: C. Nikou

3.3 Histogram Processing
- Histogram Specification

- Equalize the initial histogram of the image:

$$s = T(r) = (L-1)\int_0^r p_r(w)\,dw$$

- Equalize the target histogram:

$$s = G(z) = (L-1)\int_0^r p_z(w)\,dw$$

$$G(z) = T(r)$$

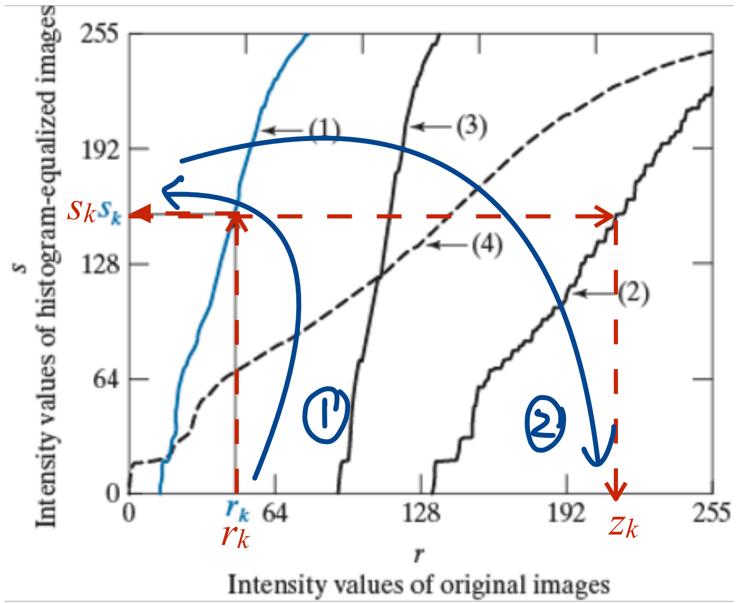
- Obtain the inverse transform:

$$z = G^{-1}(s) = G^{-1}(T(r))$$

credit of this slide: C. Nikou

3.3 Histogram Processing - Histogram Specification

- Obtain the inverse transform: $z = G^{-1}(s) = G^{-1}(T(r))$
- In practice, for every value of r in the image:
  - get its equalized transformation $s=T(r)$.
  - perform the inverse mapping $z=G^{-1}(s)$, where $s=G(z)$ is the equalized target histogram



Ex. from (1) to (2)

$r_k \rightarrow s_k \rightarrow z_k$

credit of this slide: C. Nikou

3.3 Histogram Processing - Histogram Specification

The discrete case:

- Equalize the initial histogram of the image:

$$s_k = T(r_k) = (L-1)\sum_{j=0}^{k} p_r(r_j) = \frac{(L-1)}{MN}\sum_{j=0}^{k} n_j$$

- Equalize the target histogram:

$$G(z) = T(r)$$

$$s_k = G(z_q) = (L-1)\sum_{i=0}^{q} p_z(r_i)$$

- Obtain the inverse transform: $z_q = G^{-1}(s_k) = G^{-1}(T(r_k))$

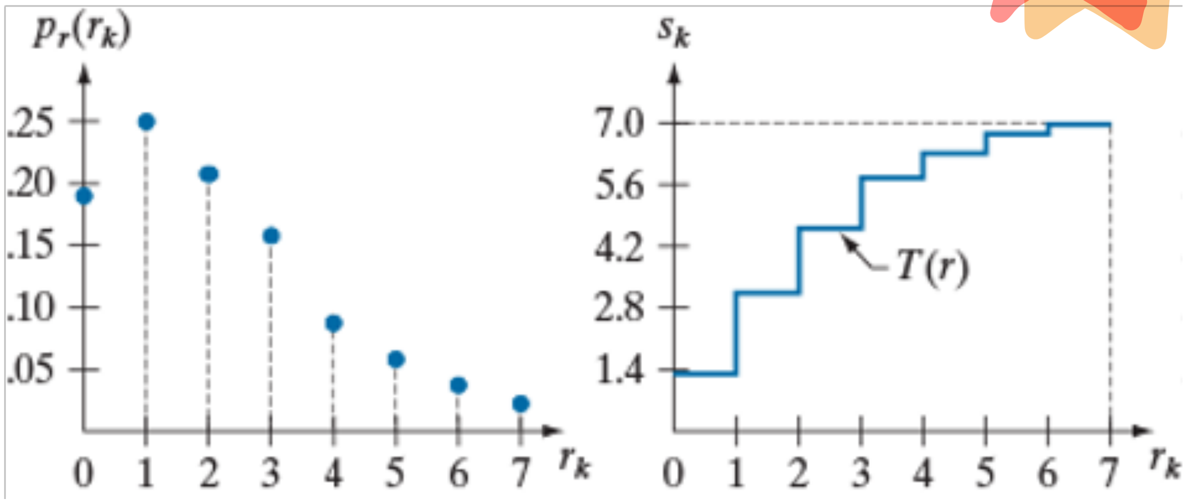
credit of this slide: C. Nikou

3.3 Histogram Processing
- Example of Histogram Specification

Consider again the 3-bit 64x64 image:

會考

r_k	n_k	$p_r(r_k) = n_k/MN$
$r_0 = 0$	790	0.19
$r_1 = 1$	1023	0.25
$r_2 = 2$	850	0.21
$r_3 = 3$	656	0.16
$r_4 = 4$	329	0.08
$r_5 = 5$	245	0.06
$r_6 = 6$	122	0.03
$r_7 = 7$	81	0.02



Histogram Equalization:

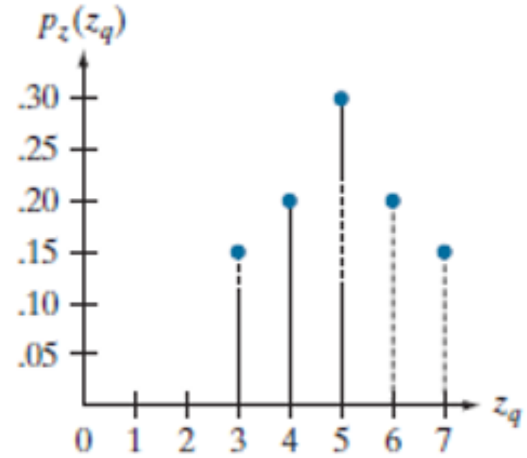
$$s_0 = 1,\ s_1 = 3,\ s_2 = 5,\ s_3 = 6,\ s_4 = 6,\ s_5 = 7,\ s_6 = 7,\ s_7 = 7$$

credit of this slide: C. Nikou

3.3 Histogram Processing
- Example of Histogram Specification

It is desired to transform this histogram to:

z_q	Specified $p_z(z_q)$	Actual $p_z(z_q)$
$z_0 = 0$	0.00	0.00
$z_1 = 1$	0.00	0.00
$z_2 = 2$	0.00	0.00
$z_3 = 3$	0.15	0.19
$z_4 = 4$	0.20	0.25
$z_5 = 5$	0.30	0.21
$z_6 = 6$	0.20	0.24
$z_7 = 7$	0.15	0.11



$$p_z(z_0) = 0.00 \quad p_z(z_1) = 0.00 \quad p_z(z_2) = 0.00 \quad p_z(z_3) = 0.15$$

$$p_z(z_4) = 0.20 \quad p_z(z_5) = 0.30 \quad p_z(z_6) = 0.20 \quad p_z(z_7) = 0.15$$

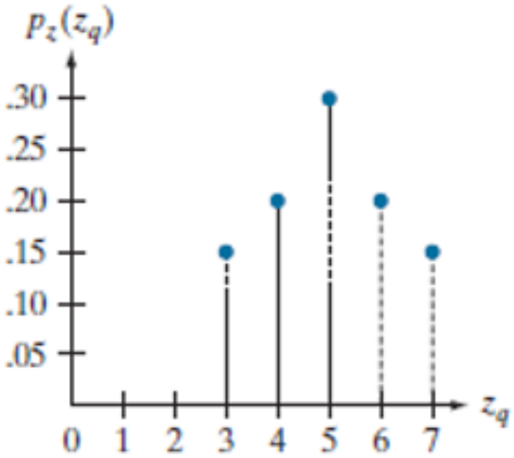
with $z_0 = 0, z_1 = 1, z_2 = 2, z_3 = 3, z_4 = 4, z_5 = 5, z_6 = 6, z_7 = 7.$

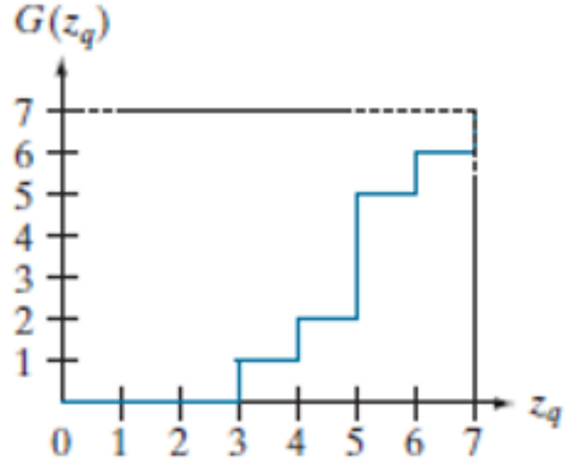
credit of this slide: C. Nikou

3.3 Histogram Processing - Example of Histogram Specification

It is desired to transform this histogram to:

z_q	Specified $p_z(z_q)$	Actual $p_z(z_q)$
$z_0 = 0$	0.00	0.00
$z_1 = 1$	0.00	0.00
$z_2 = 2$	0.00	0.00
$z_3 = 3$	0.15	0.19
$z_4 = 4$	0.20	0.25
$z_5 = 5$	0.30	0.21
$z_6 = 6$	0.20	0.24
$z_7 = 7$	0.15	0.11





Histogram Equalization:

$$G(z_0) = 0 \quad G(z_1) = 0 \quad G(z_2) = 0 \quad G(z_3) = 1$$
$$G(z_4) = 2 \quad G(z_5) = 5 \quad G(z_6) = 6 \quad G(z_7) = 7$$

credit of this slide: C. Nikou

3.3 Histogram Processing
- Example of Histogram Specification

Notice that $G(z)$ may not be strictly monotonic. We must resolve this ambiguity by choosing, e.g. the smallest value for the inverse mapping.

Perform inverse mapping: find the smallest value of z_q that is closest to s_k:

$s_k = T(r_i)$	$G(z_q)$
$s_0 = 1$	$G(z_0) = 0$
$s_1 = 3$	$G(z_1) = 0$
$s_2 = 5$	$G(z_2) = 0$
$s_3 = 6$	$G(z_3) = 1$
$s_4 = 6$	$G(z_4) = 2$
$s_5 = 7$	$G(z_5) = 5$
$s_6 = 7$	$G(z_6) = 6$
$s_7 = 7$	$G(z_7) = 7$

$$s_k \rightarrow z_q$$
$$1 \rightarrow 3$$
$$3 \rightarrow 4$$
$$5 \rightarrow 5$$
$$6 \rightarrow 6$$
$$7 \rightarrow 7$$

e.g. every pixel with value $s_0=1$ in the histogram-equalized image would have a value of 3 (z_3) in the histogram-specified image.

credit of this slide: C. Nikou

3.3 Histogram Processing
- Example of Histogram Specification

Notice that due to discretization, the resulting histogram will rarely be exactly the same as the desired histogram.

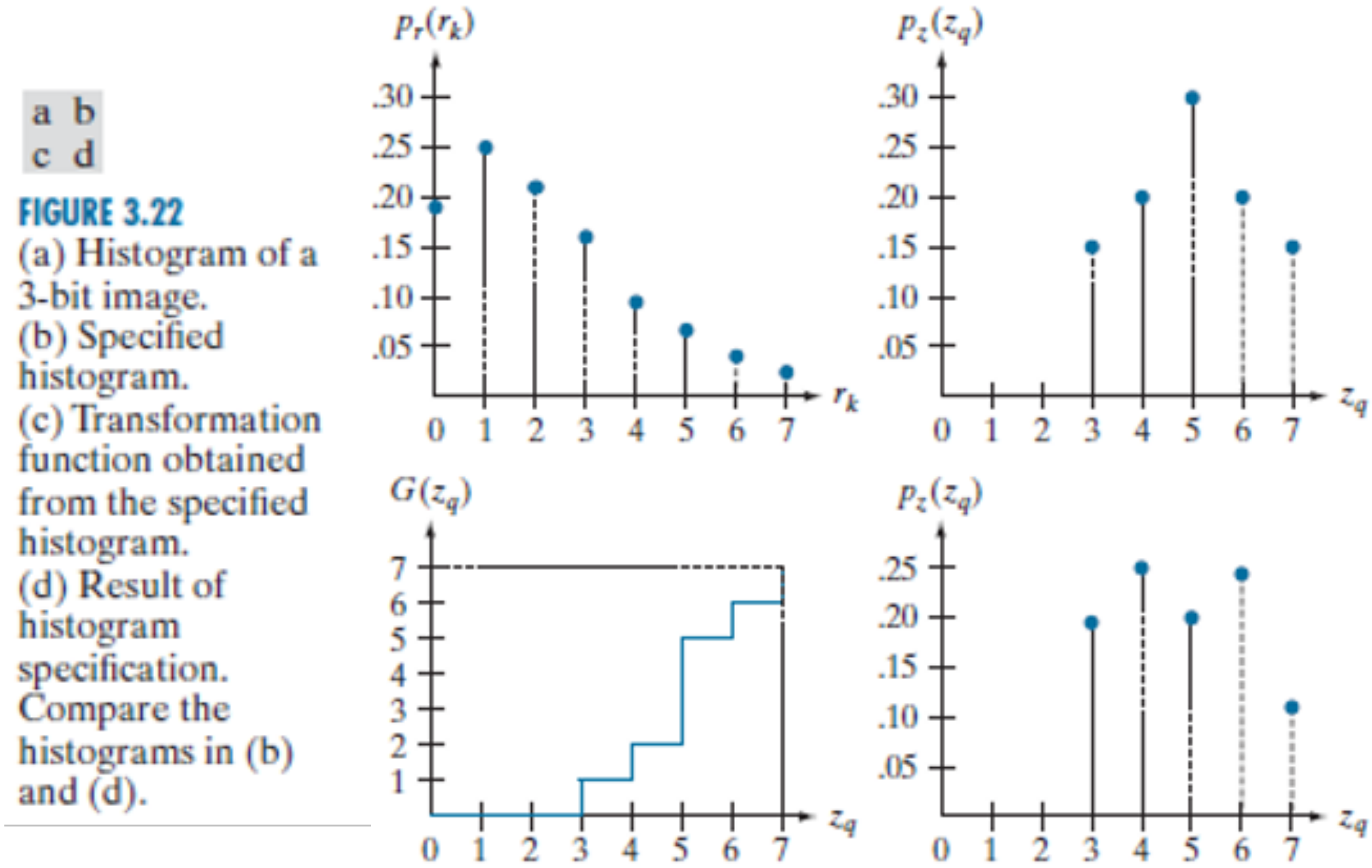
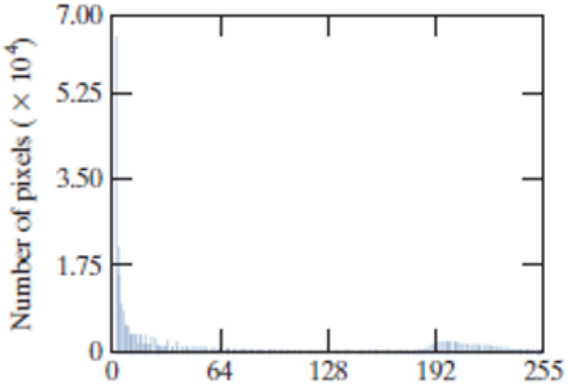
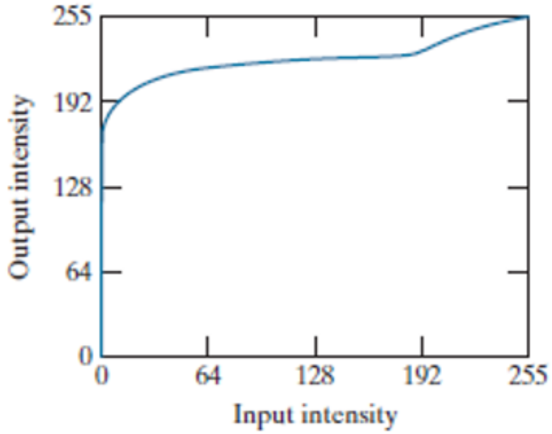


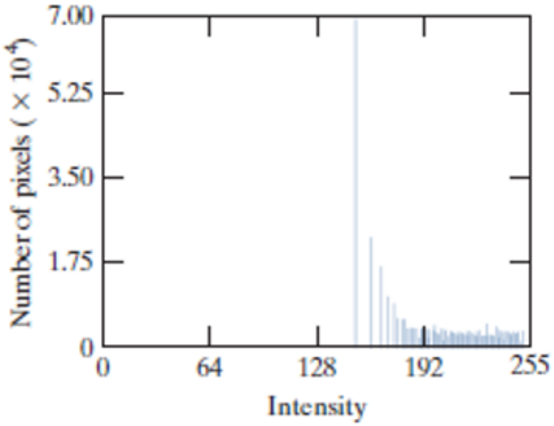
FIGURE 3.22 (a) Histogram of a 3-bit image. (b) Specified histogram. (c) Transformation function obtained from the specified histogram. (d) Result of histogram specification. Compare the histograms in (b) and (d).

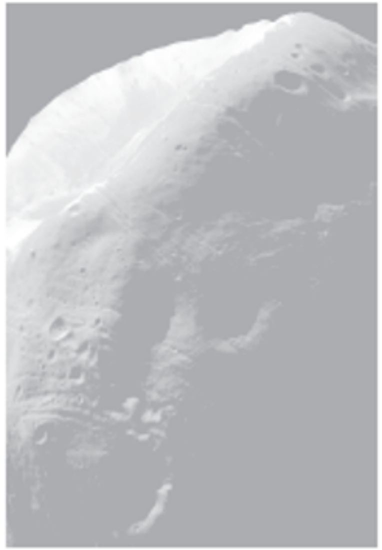
credit of this slide: C. Nikou

3.3 Histogram Processing
- Example of Histogram Specification







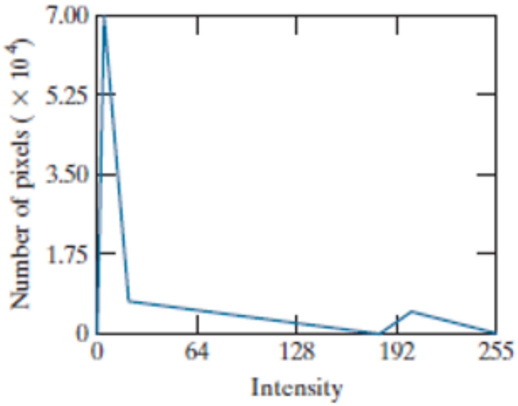


Original image

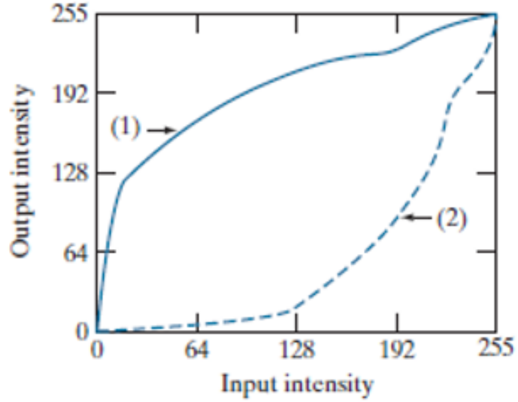
Histogram equalization

3.3 Histogram Processing
- Example of Histogram Specification

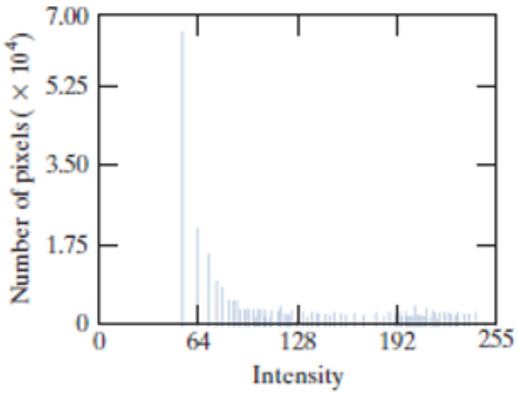
Specified histogram

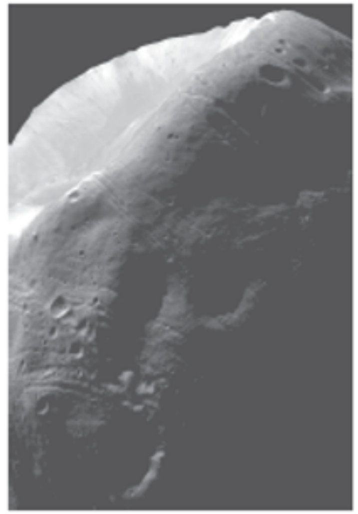


Transformation function (1) and its inverse (2)



Resulting histogram





Result Image

3.3 Histogram Processing
- Histogram Specification

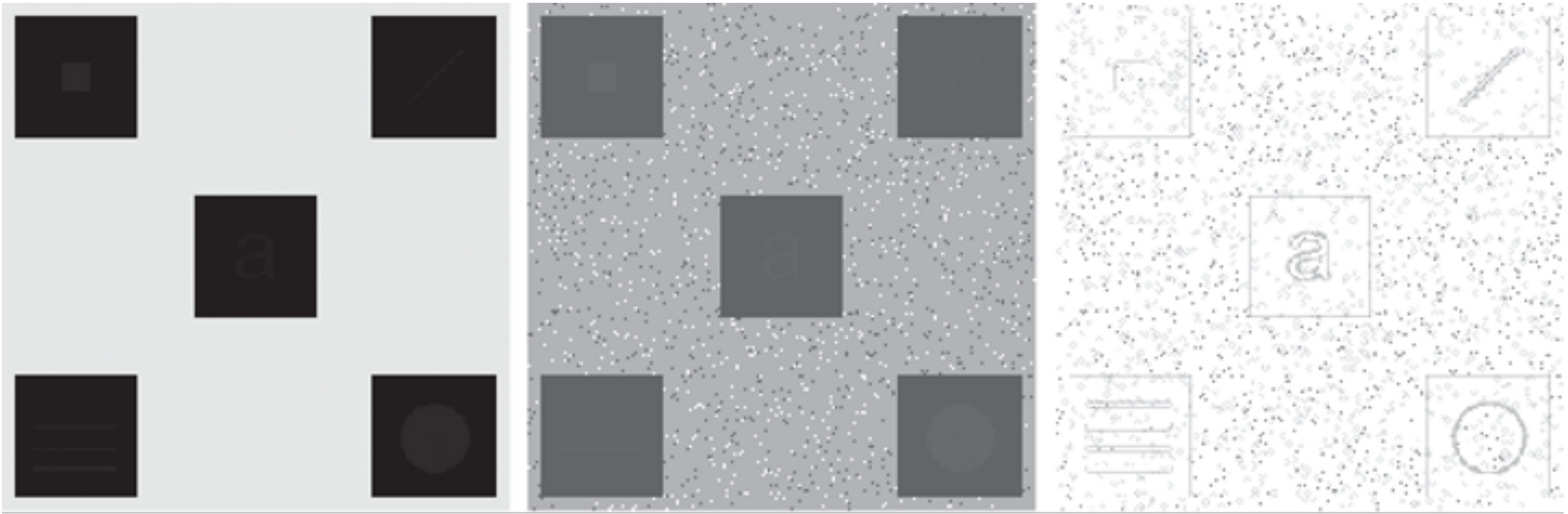
- When multiple images of the same scene, but taken under slightly different lighting conditions, are to be compared
  -- e.g., visual surveillance, image stitching, stereo, etc.

- Get high contrast images by using a specified V-shaped histogram.

- Usually, histogram specification is a trial-and-error process.

credit of this slide: Y. P. Hung

3.3 Histogram Processing
- Local Histogram Processing



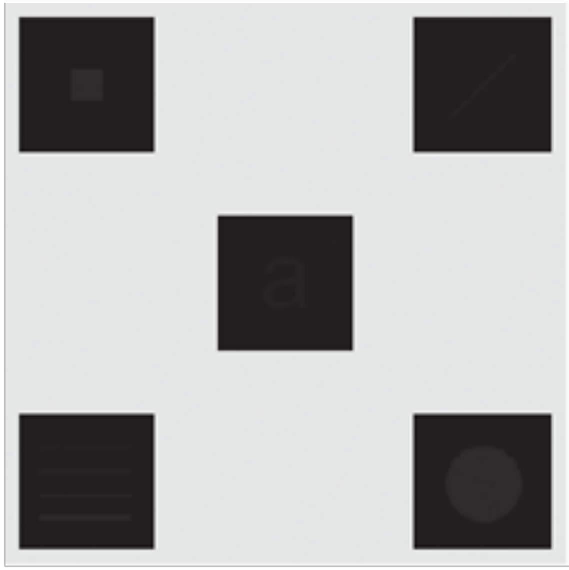
Original Image

Global Histogram Equalization

Local Histogram Equalization with a neighborhood of size 3x3

- Overlapping windows – smooth, time consuming
- Non-overlapping windows – “blocky” effect, faster

3.3 Histogram Processing
- Local Histogram Processing



Original Image

Local Enhancement based on local histogram statistics

Local Histogram Equalization with a neighborhood of size 3x3

$$g(x,y)=\begin{cases} C\,f(x,y) & \text{if } k_0 m_G \le m_{S_{xy}} \le k_1 m_G \text{ AND } k_2\sigma_G \le \sigma_{S_{xy}} \le k_3\sigma_G \\ f(x,y) & \text{otherwise} \end{cases}$$

global mean (m_G) global std. (σ_G)

$$m_{S_{xy}} = \sum_{i=0}^{L-1} r_i p_{S_{xy}}(r_i) \qquad \sigma^2_{S_{xy}} = \sum_{i=0}^{L-1} (r_i - m_{S_{xy}})^2 p_{S_{xy}}(r_i)$$

$k_0 = 0$ $k_2 = 0$

$k_1 = 0.1$ $k_3 = 0.1$

$C = 22.8$