Announcement

1/28/2025 Tuesday

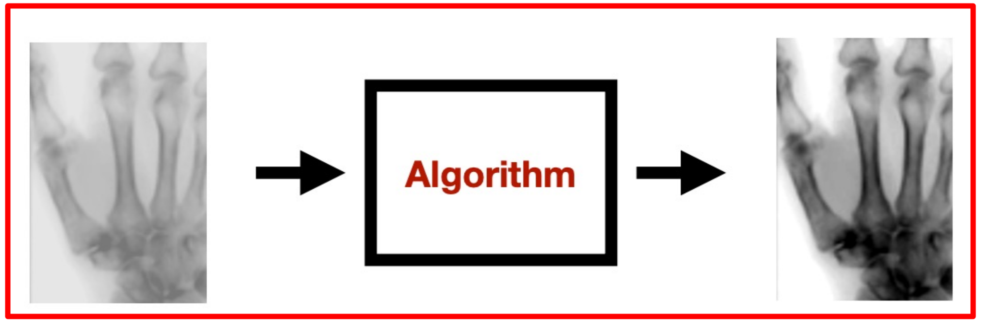
- Zoom class

- Digital Image Fundamentals
- Pixel-level processing
  - Intensity transformations

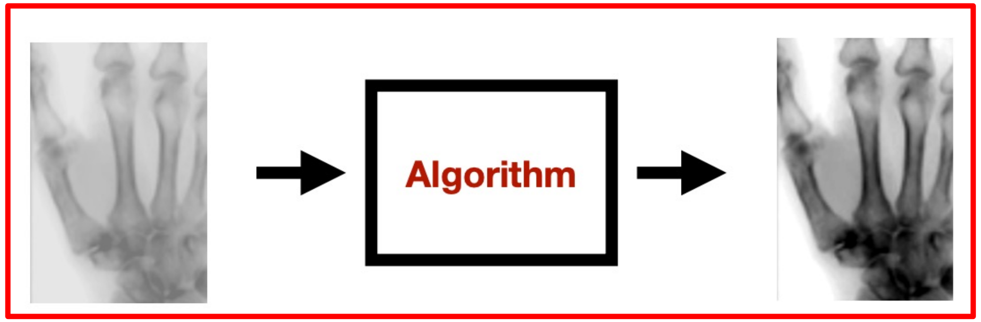
Two types of digital image processing

- Pixel-level processing
  - Same function applied to every pixel

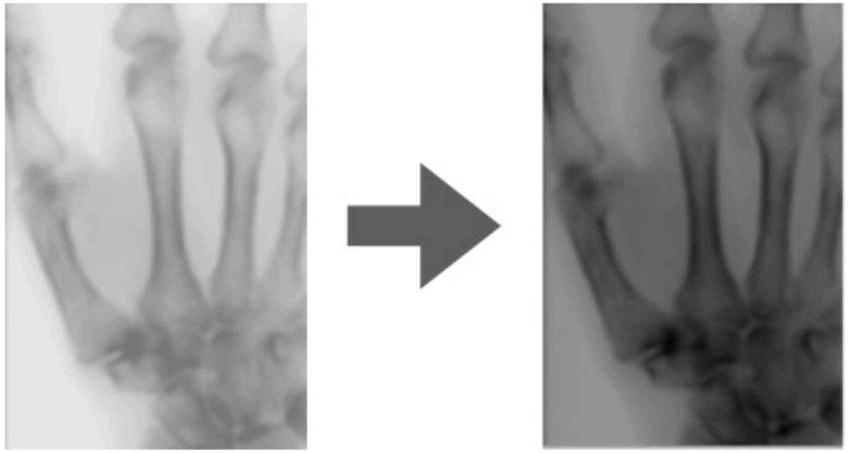
- Patch-level processing (filtering)
  - Same filter applied to sub-regions/patches



Pixel-level processing



- Same function applied to every pixel

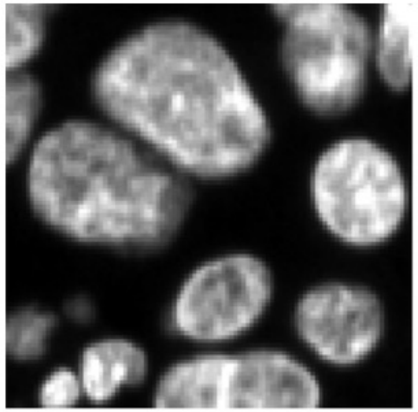


243	222	171
203	145	137
209	132	193

123	102	51
83	25	17
89	12	73

$$f(x) = x - 120$$

Example: image thresholding



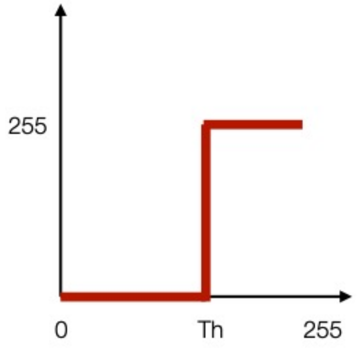
Image

Lookup table

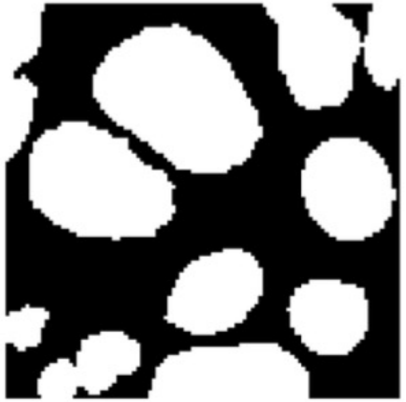
p_{in}	p_{out}
0	0
...	...
Th-1	0
Th	255
...	...
255	255

```
I[I >= Th] = 255
I[I < Th]  =  0
```

Visualization



Binary segmentation:
Find the threshold



Binary segmentation

Patch-level processing (filtering)

- Define the patch-level function (input size)

243	222	171
203	145	137
209	132	193

→

121	111	85
101	193	68
104	66	96

$$f : pa \rightarrow p'$$

Input **patch** Output pixel

$$p' = \text{median}(pa)$$

- Slide through every patch

30	31	32	3	4
0	6	99	30	30
99	35	33	32	98
0	90	90	36	31
32	31	0	90	90

	?			

Example: median filter

Patch-level processing (filtering)

- Define the patch-level function (input size)

243	222	171
203	145	137
209	132	193

121	111	85
101	193	68
104	66	96

$$f : pa \rightarrow p'$$

Input **patch** Output pixel

$$p' = \text{median}(pa)$$

- Slide through every patch

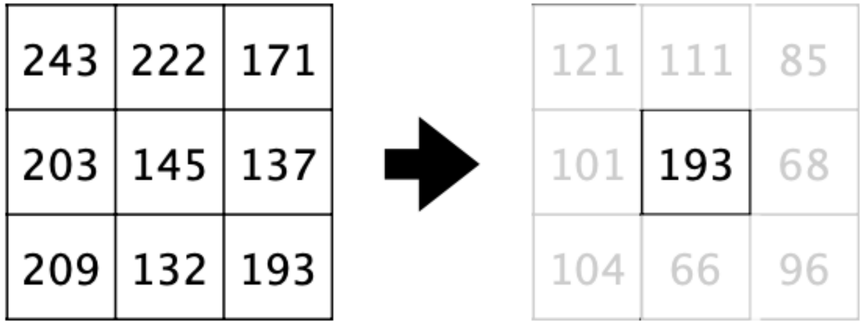
30	31	32	3	4
0	6	99	30	30
99	35	33	32	98
0	90	90	36	31
32	31	0	90	90

	32			

Example: median filter

Patch-level processing (filtering)

- Define the patch-level function (input size)

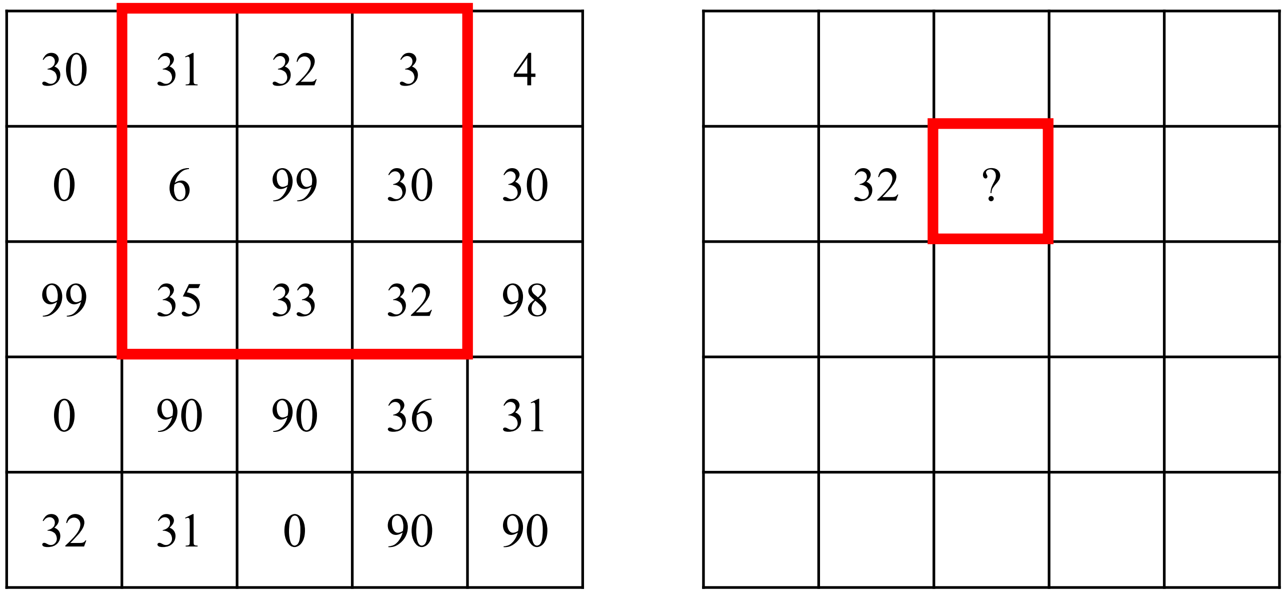


$$f : pa \to p'$$

Input **patch** Output pixel

$$p' = \text{median}(pa)$$

- Slide through every patch



Example: median filter

Patch-level processing (filtering)

- Define the patch-level function (input size)

243	222	171
203	145	137
209	132	193

121	111	85
101	193	68
104	66	96

$$f : pa \rightarrow p'$$

Input **patch** Output pixel

$$p' = \text{median}(pa)$$

- Slide through every patch

30	31	32	3	4
0	6	99	30	30
99	35	33	32	98
0	90	90	36	31
32	31	0	90	90

	32	32		

Example: median filter

Patch-level processing (filtering)

- Define the patch-level function (input size)

243	222	171
203	145	137
209	132	193

121	111	85
101	193	68
104	66	96

$$f : pa \rightarrow p'$$

Input **patch** Output pixel

$$p' = \mathrm{median}(pa)$$

- Slide through every patch

30	31	32	3	4
0	6	99	30	30
99	35	33	32	98
0	90	90	36	31
32	31	0	90	90

	32	32	$\boxed{32}$	

Example: median filter

Patch-level processing (filtering)

- Define the patch-level function (input size)

243	222	171
203	145	137
209	132	193

$\rightarrow$

121	111	85
101	193	68
104	66	96

$$f : pa \rightarrow p'$$

Input **patch** Output pixel

$$p' = \text{median}(pa)$$

- Slide through every patch

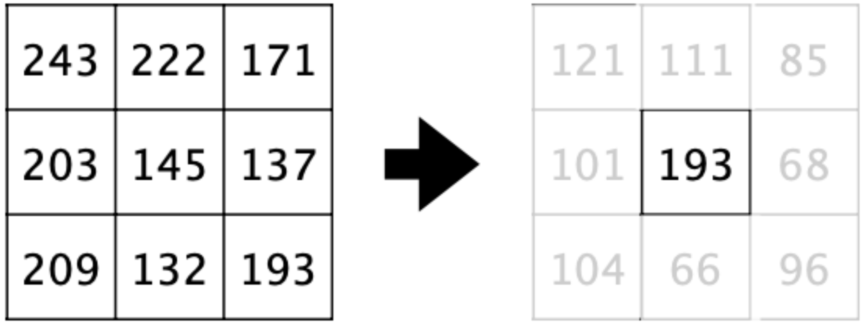
30	31	32	3	4
0	6	99	30	30
99	35	33	32	98
0	90	90	36	31
32	31	0	90	90

	32	32	32	
	35			

Example: median filter

Patch-level processing (filtering)

- Define the patch-level function (input size)



$$f : pa \rightarrow p'$$

Input **patch** Output pixel

$$p' = \text{median}(pa)$$

- Slide through every patch

30	31	32	3	4
0	6	99	30	30
99	35	33	32	98
0	90	90	36	31
32	31	0	90	90

	32	32	32	
	35	35		

Example: median filter

Patch-level processing (filtering)

- Define the patch-level function (input size)

243	222	171
203	145	137
209	132	193

➡

121	111	85
101	193	68
104	66	96

$$f : pa \to p'$$

Input **patch** Output pixel

$$p' = \text{median}(pa)$$

- Slide through every patch

30	31	32	3	4
0	6	99	30	30
99	35	33	32	98
0	90	90	36	31
32	31	0	90	90

	32	32	32	
	35	35	36	
	33	35	36	

Example: median filter

Example: denoising

Ideal image

Real-world image

Median filter result

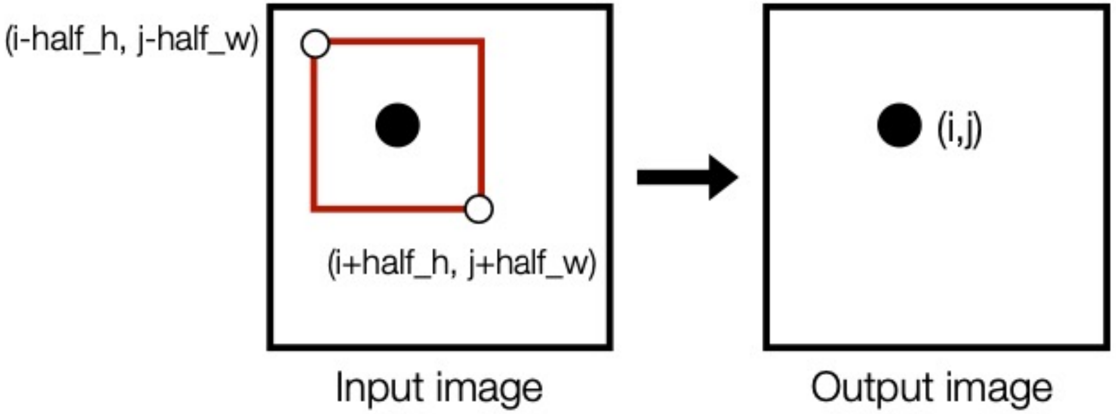
Pixel-level vs. patch-level

Pixel-level function

```
for i in range(rows):
    for j in range(cols):
        I[i,j] = f(I[i,j])
```

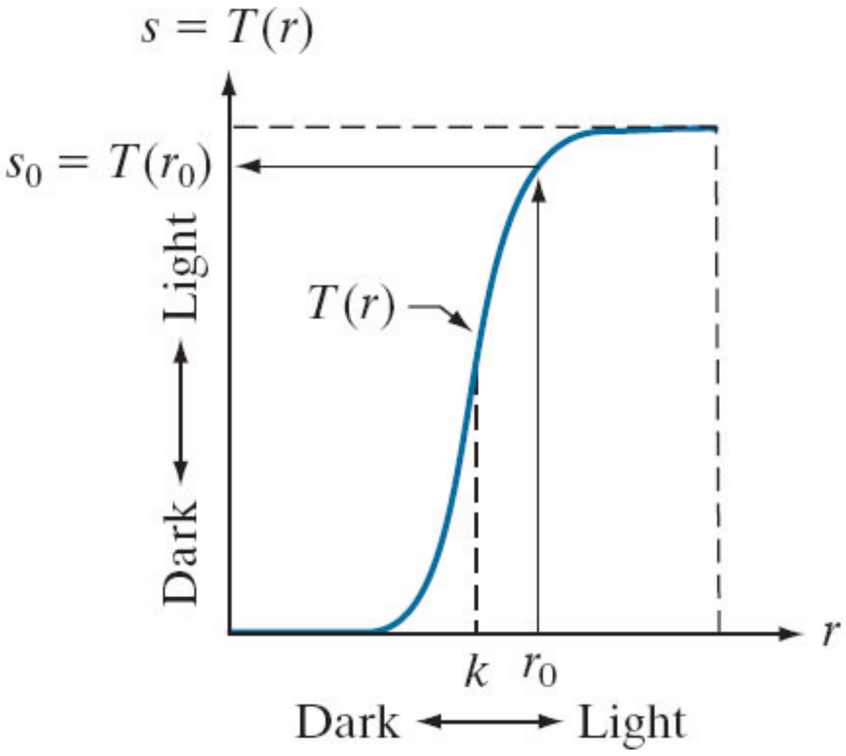
Patch-level function

```
for i in range(row_0, row_1):
    for j in range(col_0,col_1):
        I[i, j] = f(I[i-half_h:i+half_h+1, \
                    j-half_w:j+half_w+1])
```



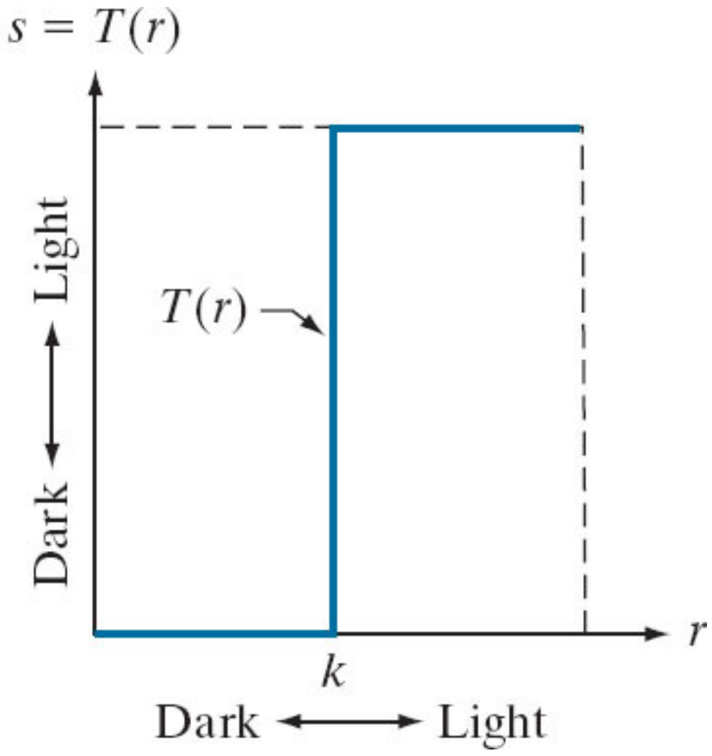
Intensity transformation

- 1x1 Neighborhood → Intensity Transformation → Image Enhancement



Contrast stretching function

Soft thresholding



Thresholding function

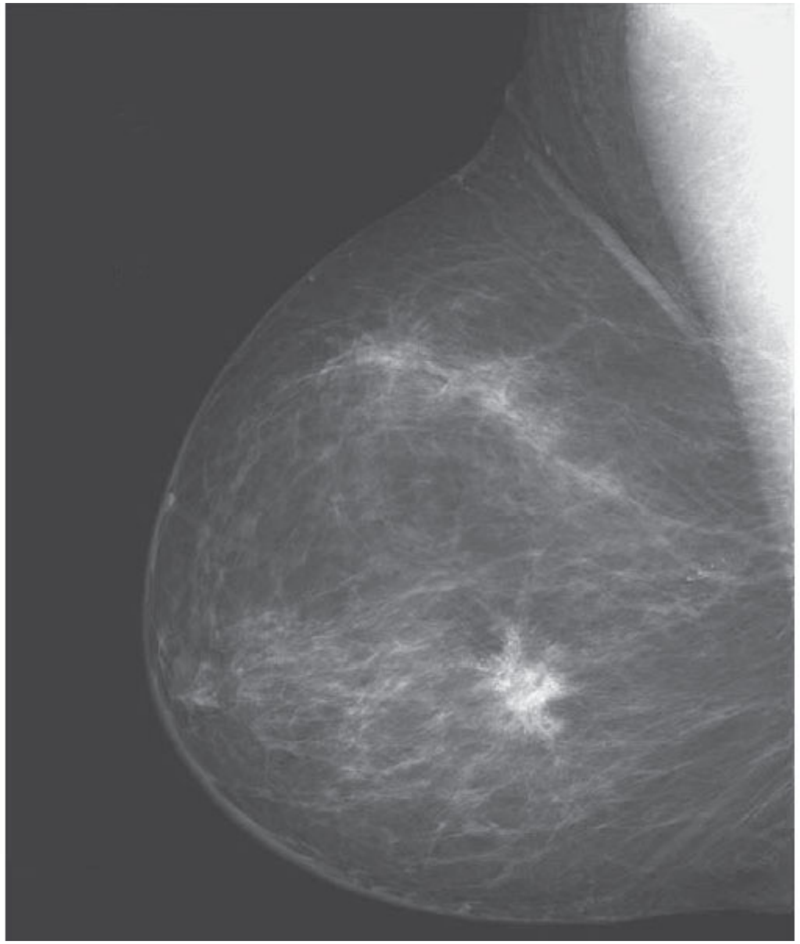
Hard thresholding

Some basic intensity transformation functions

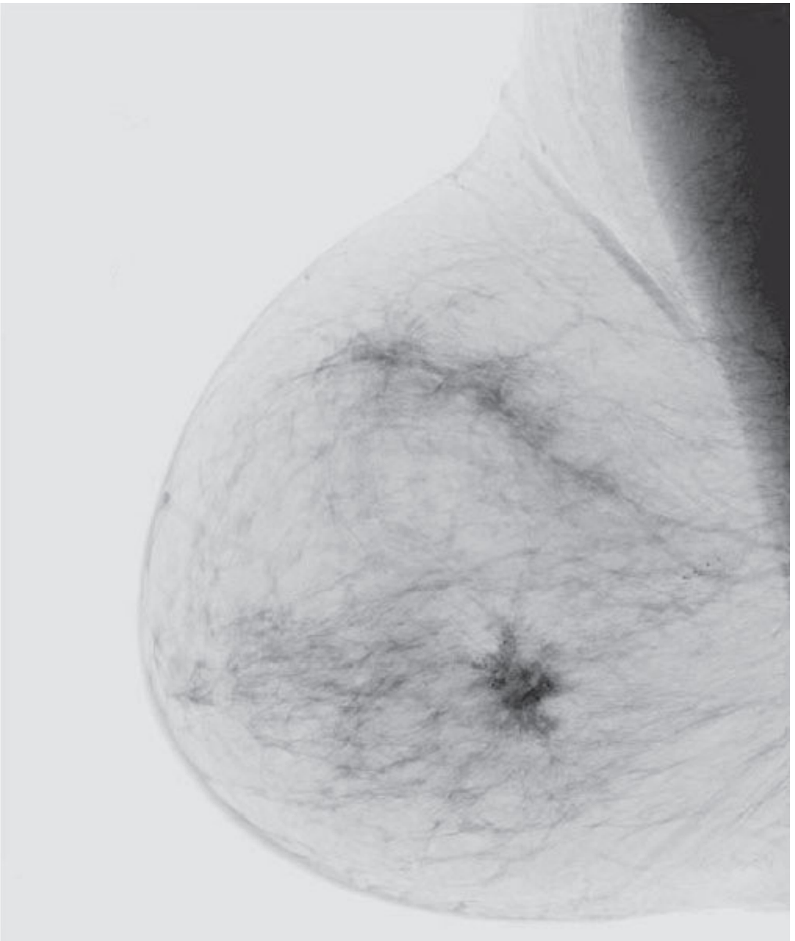
- Thresholding – Logistic function
- Log transformation
- Power-law (Gamma correction)
- Piecewise-linear transformation
- Histogram processing

Some basic intensity transformation functions

- Image negative: $S = L - 1 - r$



A digital mammogram

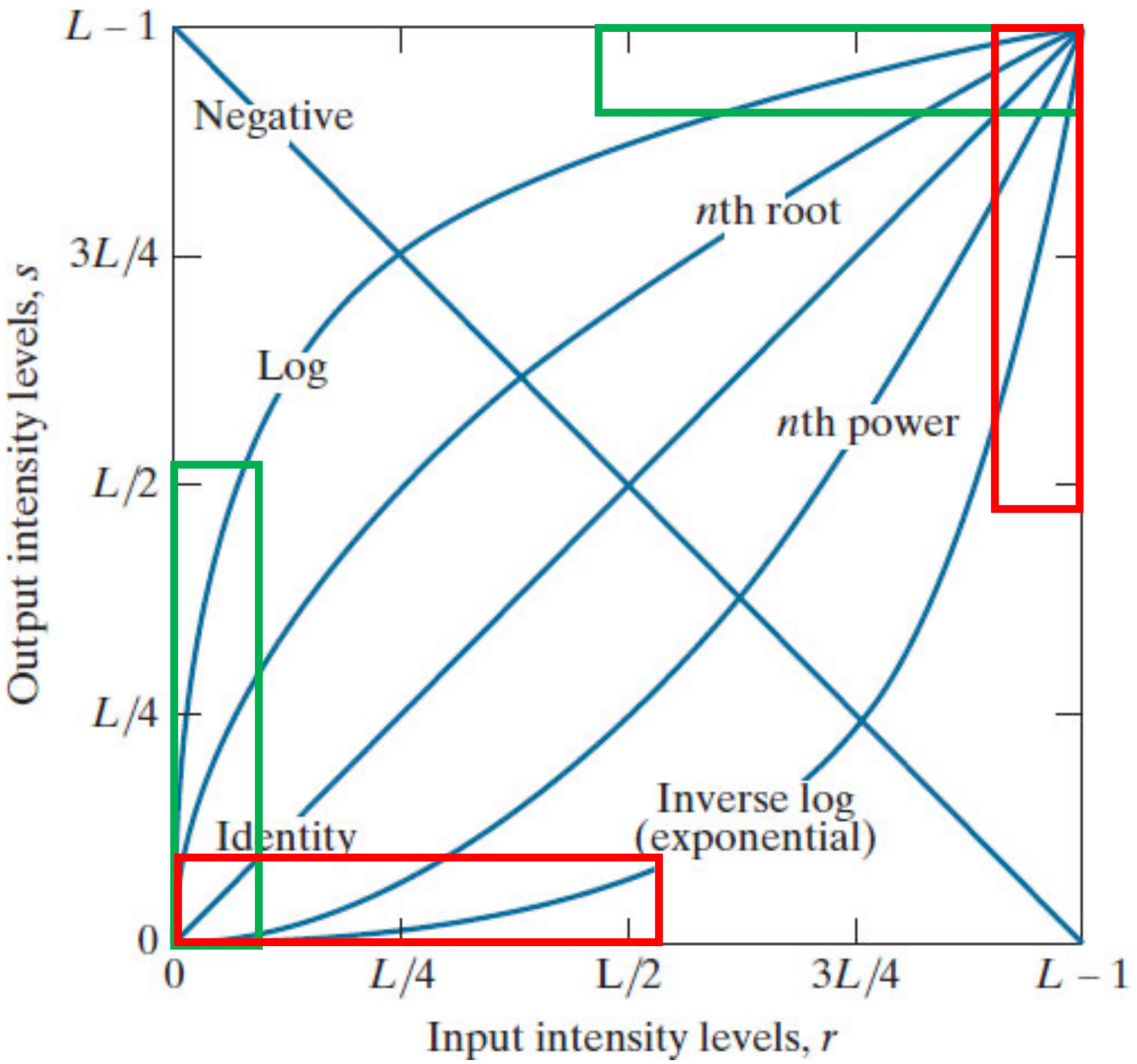


Negative image

Log transformation

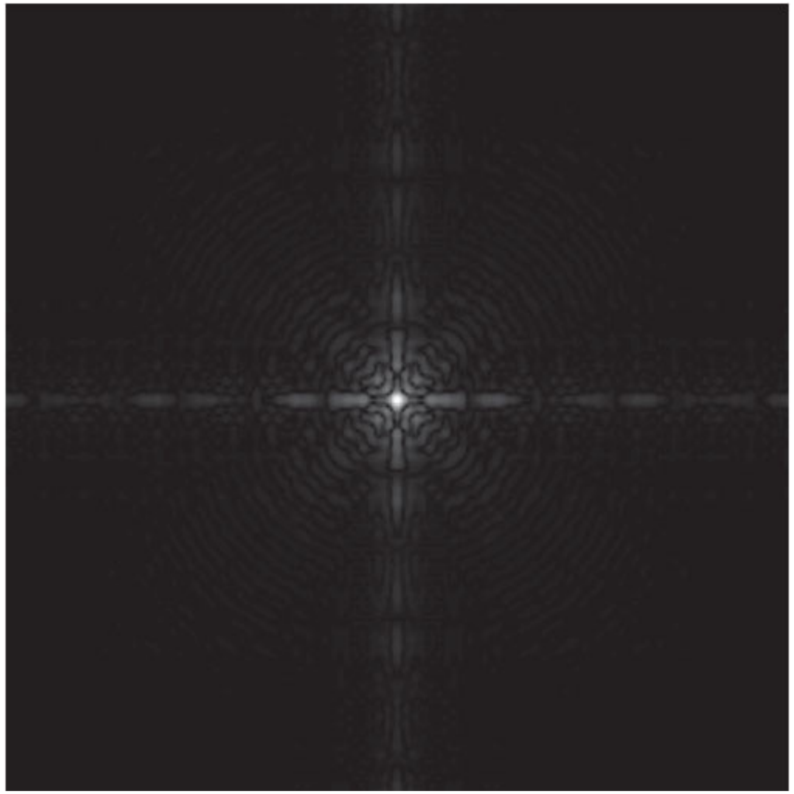
- Log function: $s = c\log(1 + r) \quad r \geq 0$
  - Stretch low intensity levels
  - Compress high intensity levels

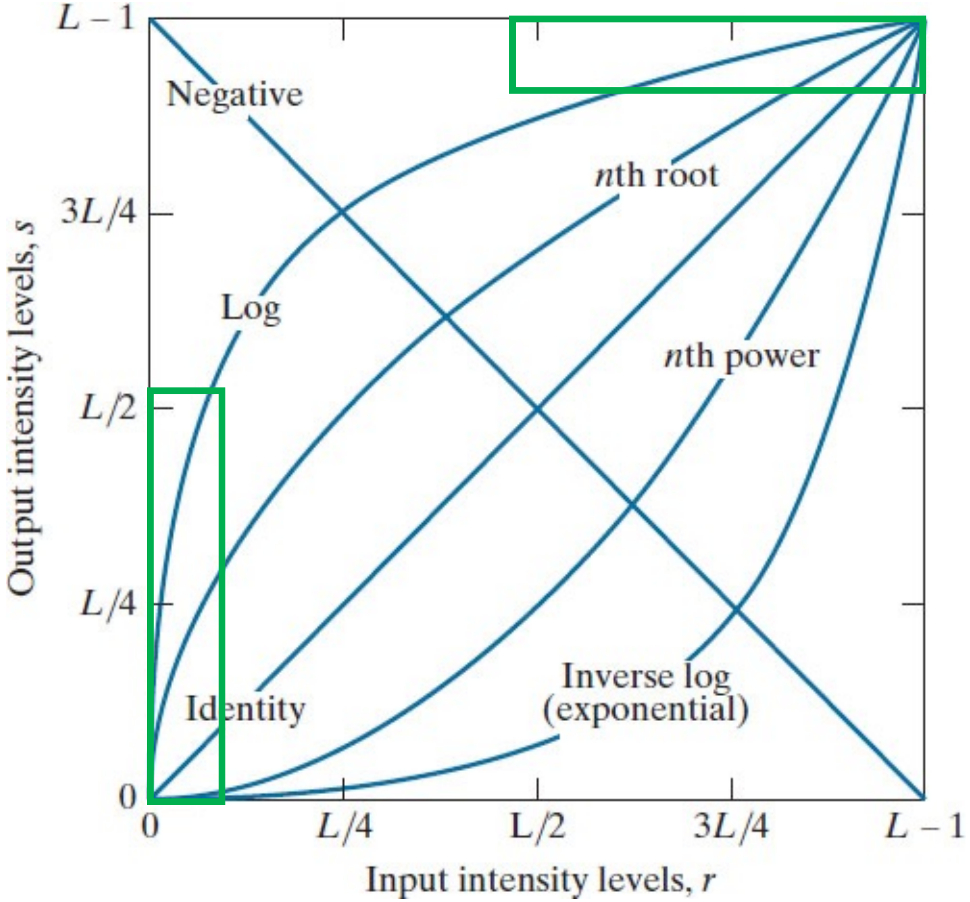
- Inverse log function: $s = c\log^{-1}(r)$
  - Stretch high intensity levels
  - Compress low intensity levels



Log transformation

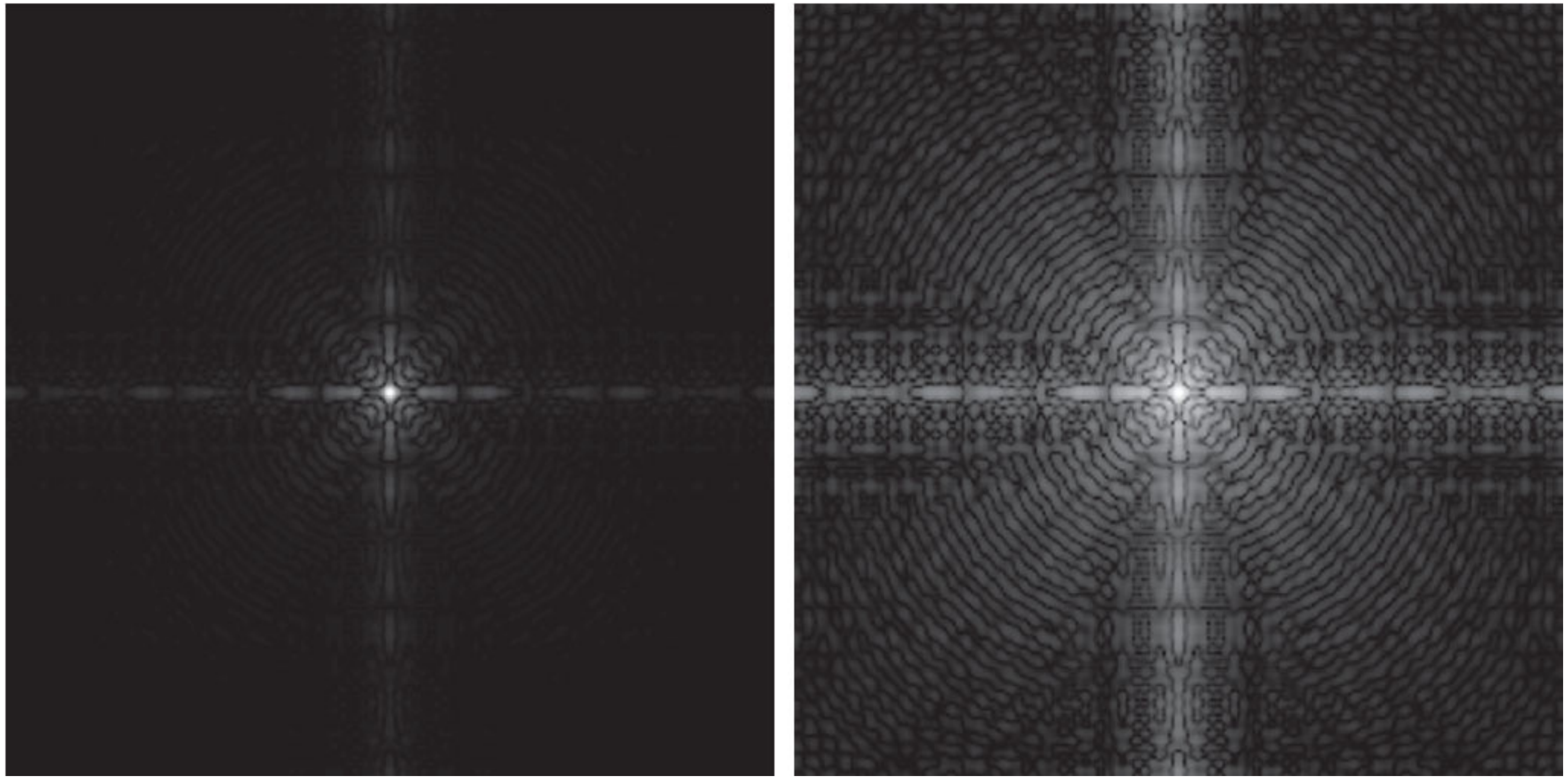
- If log transformation is applied to the following image, how would it change?





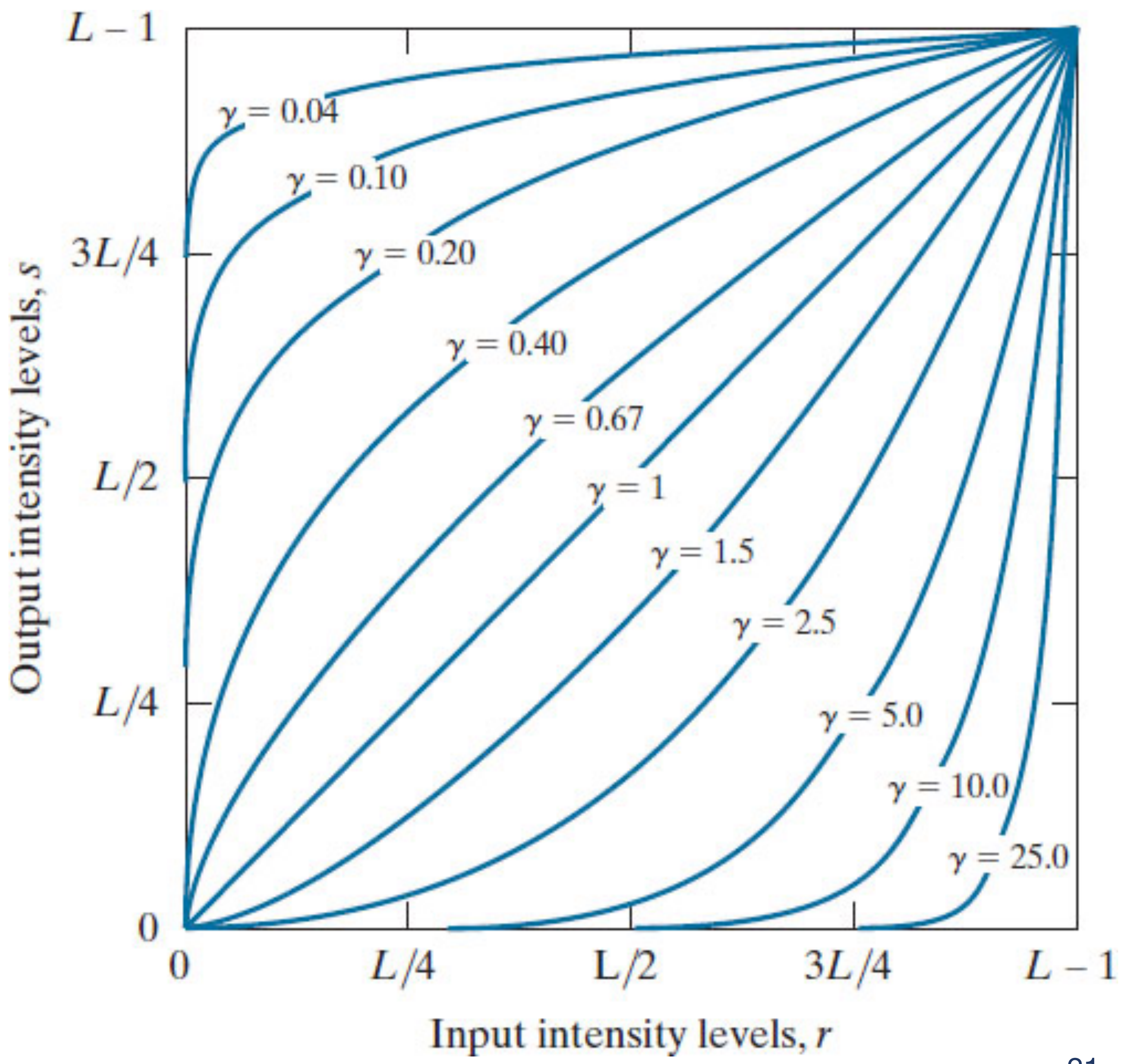
Log transformation

- If log transformation is applied to the following image, how would it change?



Power-law (gamma) transformations

- $s = cr^{\gamma}$
  - More versatile than log transformation
  - Performed by a lookup table



Power-law (gamma) transformations

- Monitors have an intensity-to-voltage response with a power function

Image of a human retina

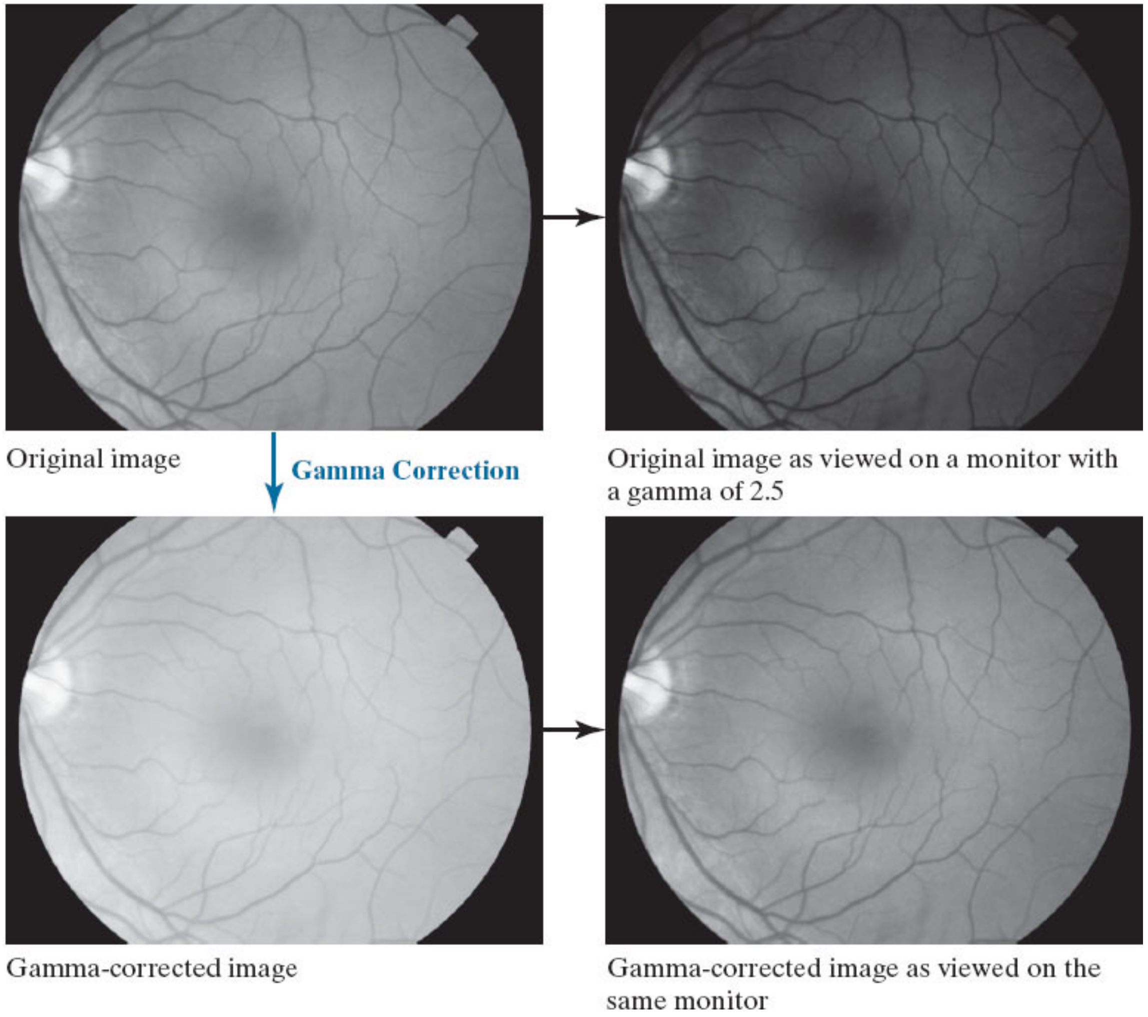


Image as it appears on a monitor

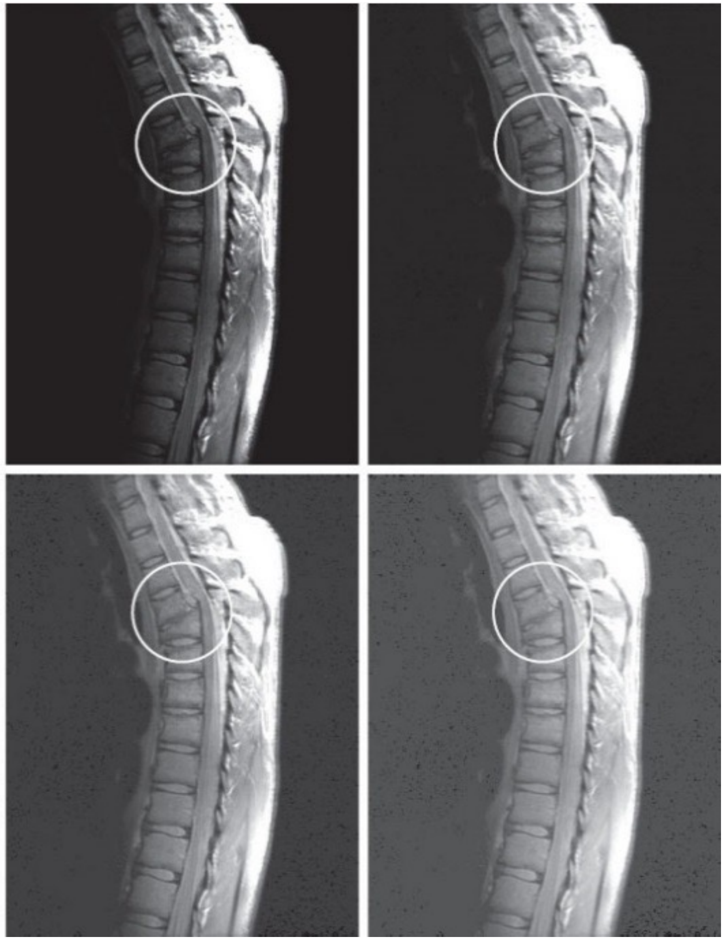
Gamma-corrected image

$$s = r^{\frac{1}{2.5}}$$

Corrected image, as it appears on the same monitor

Power-law transformations for contrast manipulation

$s = r^{0.6}$



$s = r^{0.4}$ $s = r^{0.3}$

Washed-Out Appearance caused by a small γ value

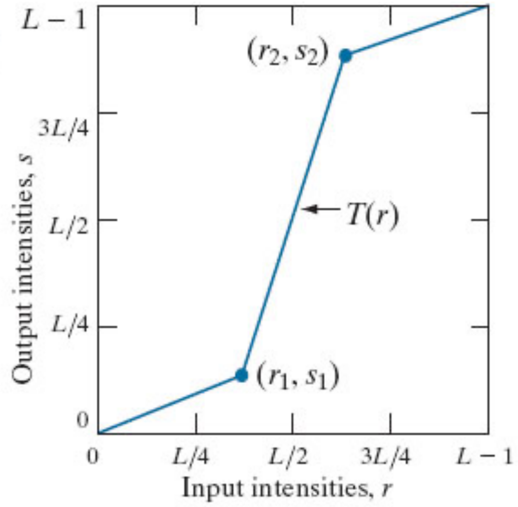
$s = r^{3}$

$s = r^{4}$ $s = r^{5}$

Washed-Out Appearance reduced by a large γ value

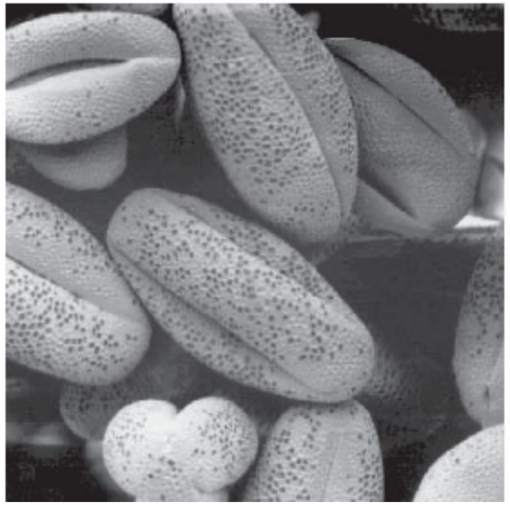
Piecewise-linear transformation functions: contrast stretching

Piecewise linear transformation function



A low-contrast electron microscope image of pollen

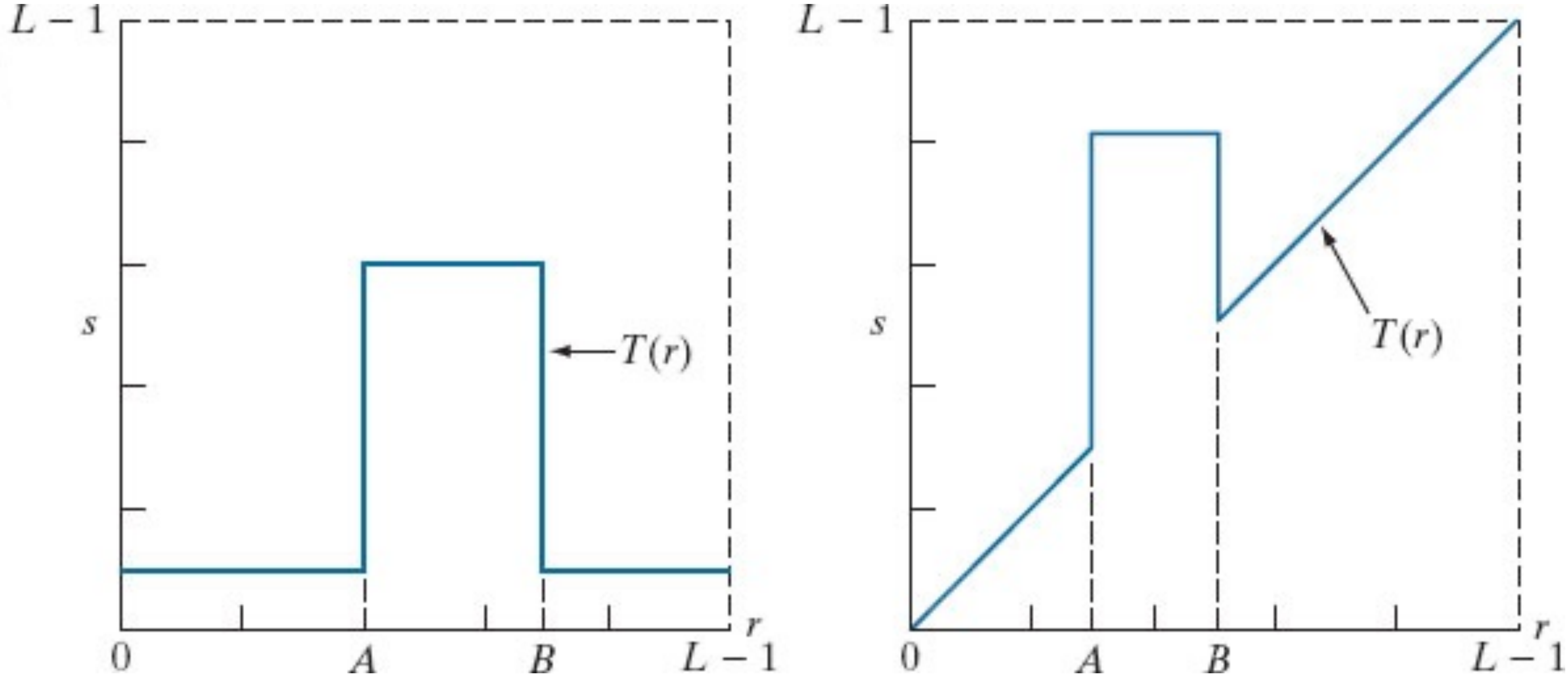
Result of contrast stretching



Result of thresholding

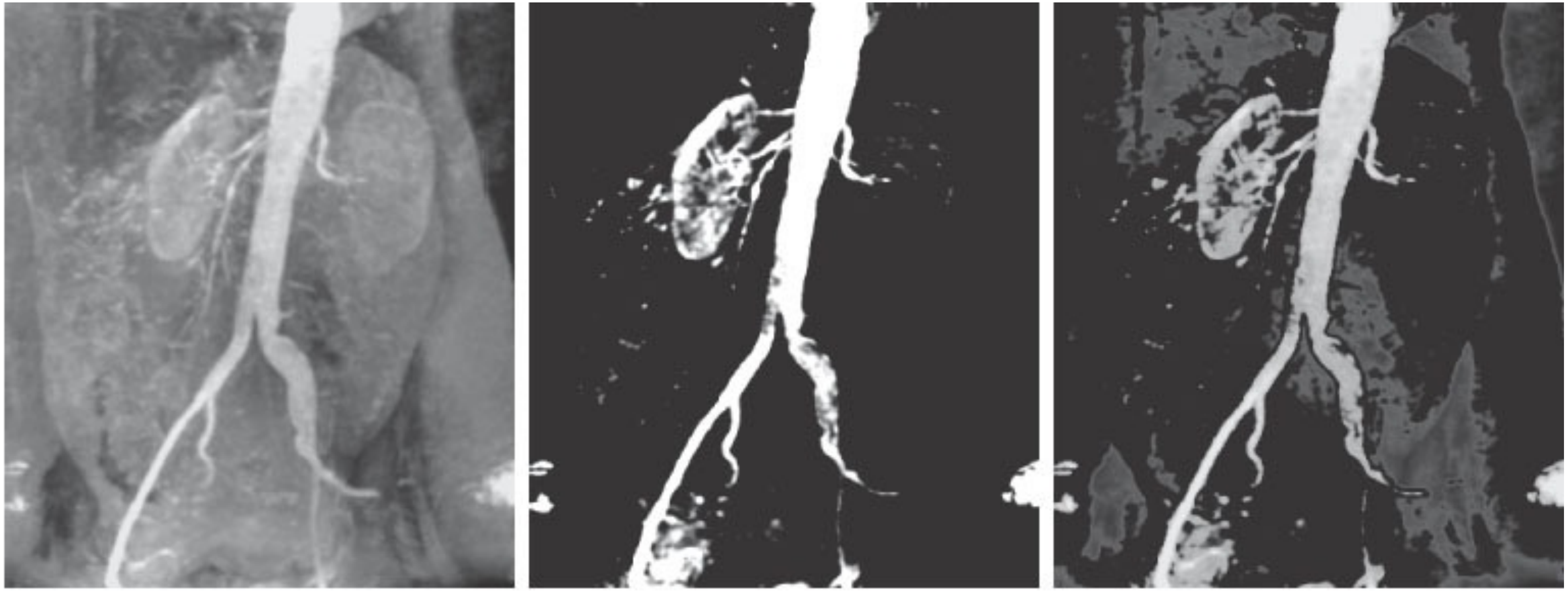
Contrast stretching examples

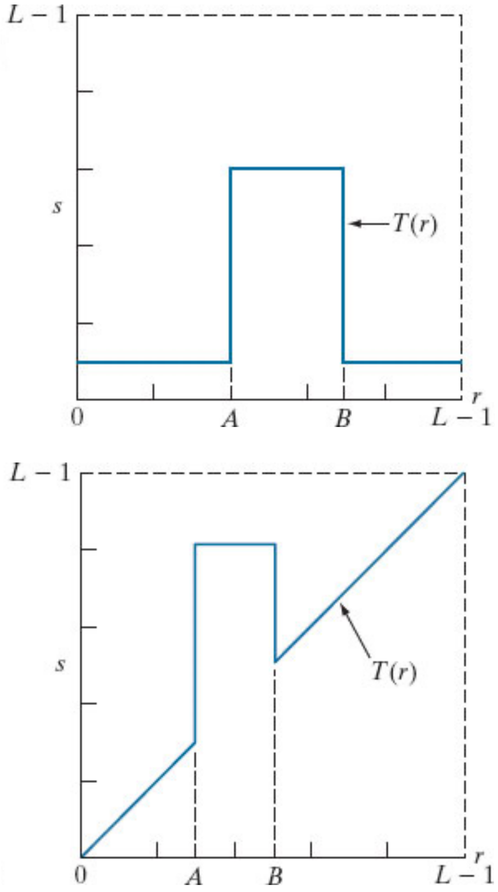
What are the effects of these two transformation functions?



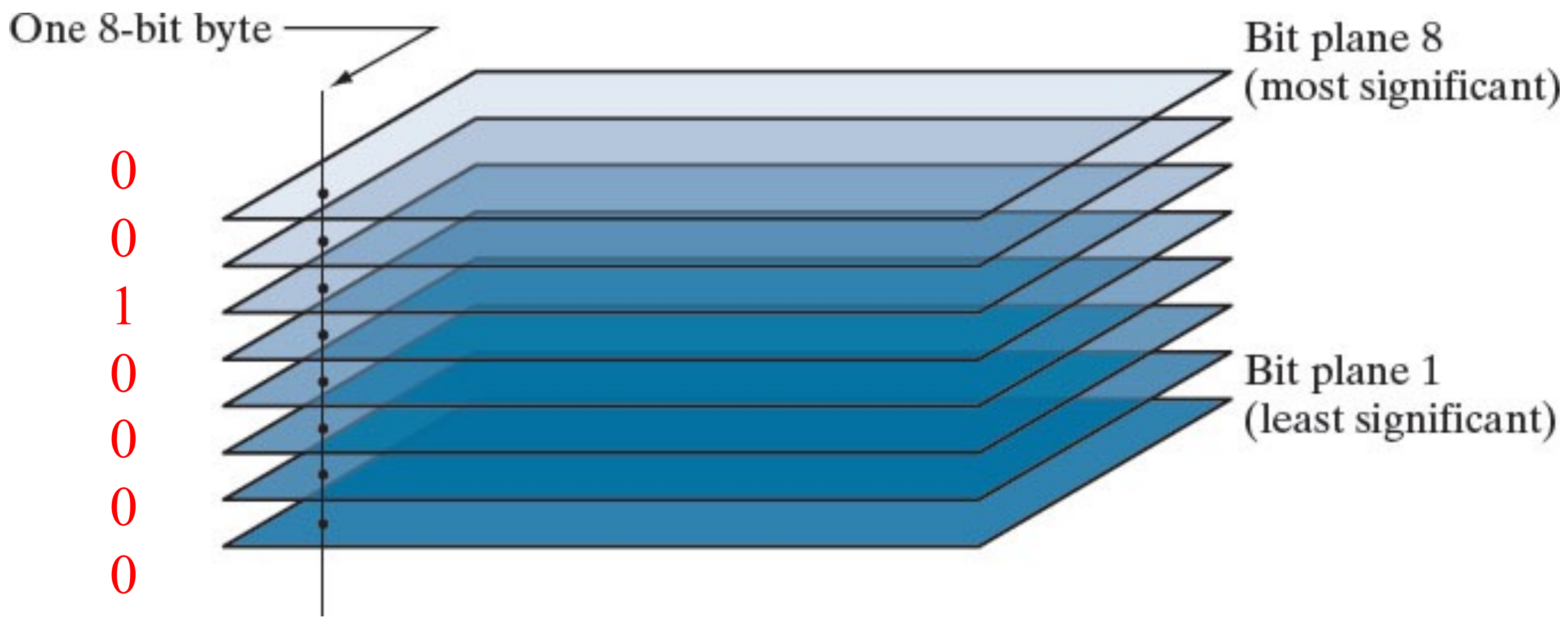
Contrast stretching examples

- An example of intensity-level slicing





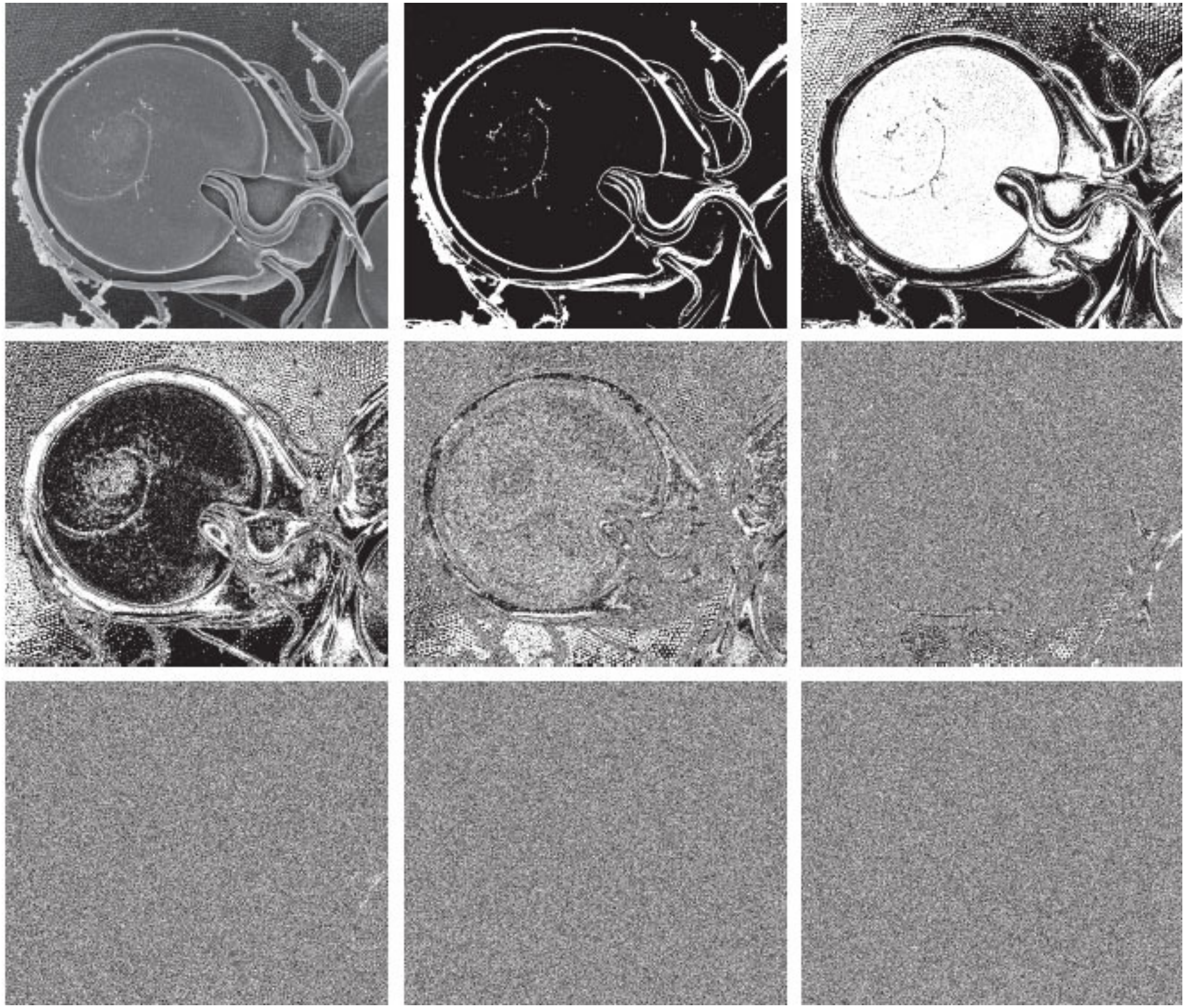
Piecewise-linear transformation functions: bit-plane slicing



1	0
1	0
1	1
1	0
1	0
1	0
1	0
1	0
255	32

Bit-planes of an 8-bit image.

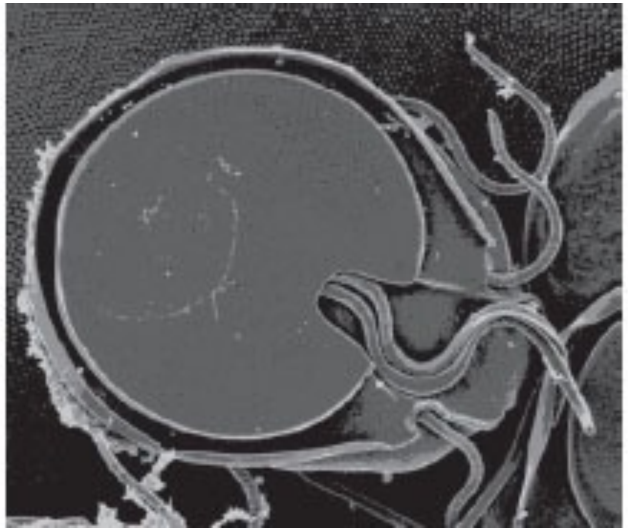
An example



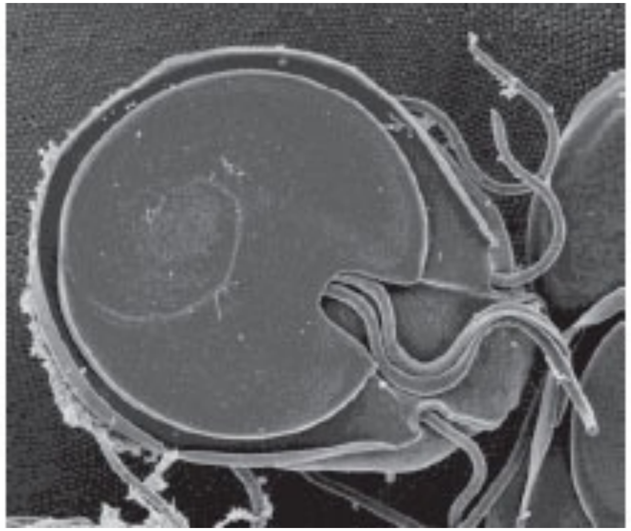
Each bit plane is a binary image

Which one is plane 8?

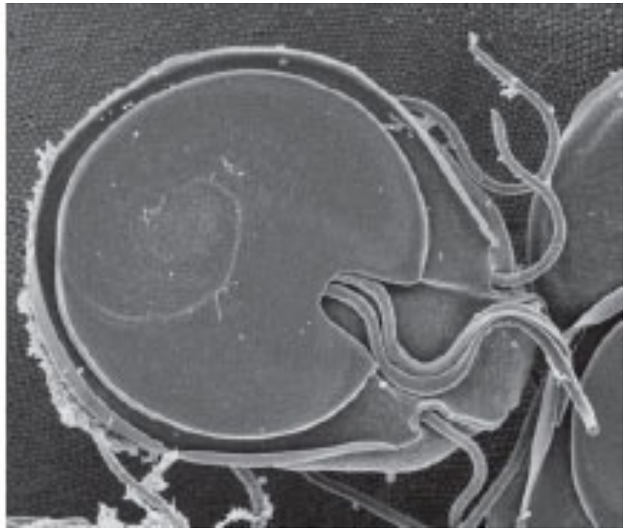
Use for image compression



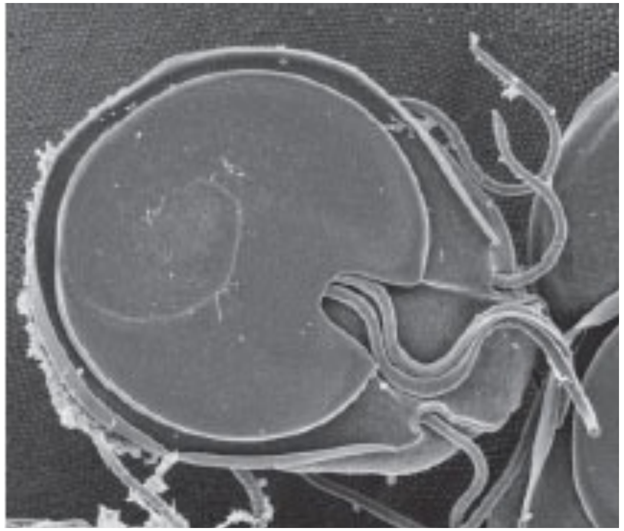
Planes 8 and 7



Planes 8, 7, and 6



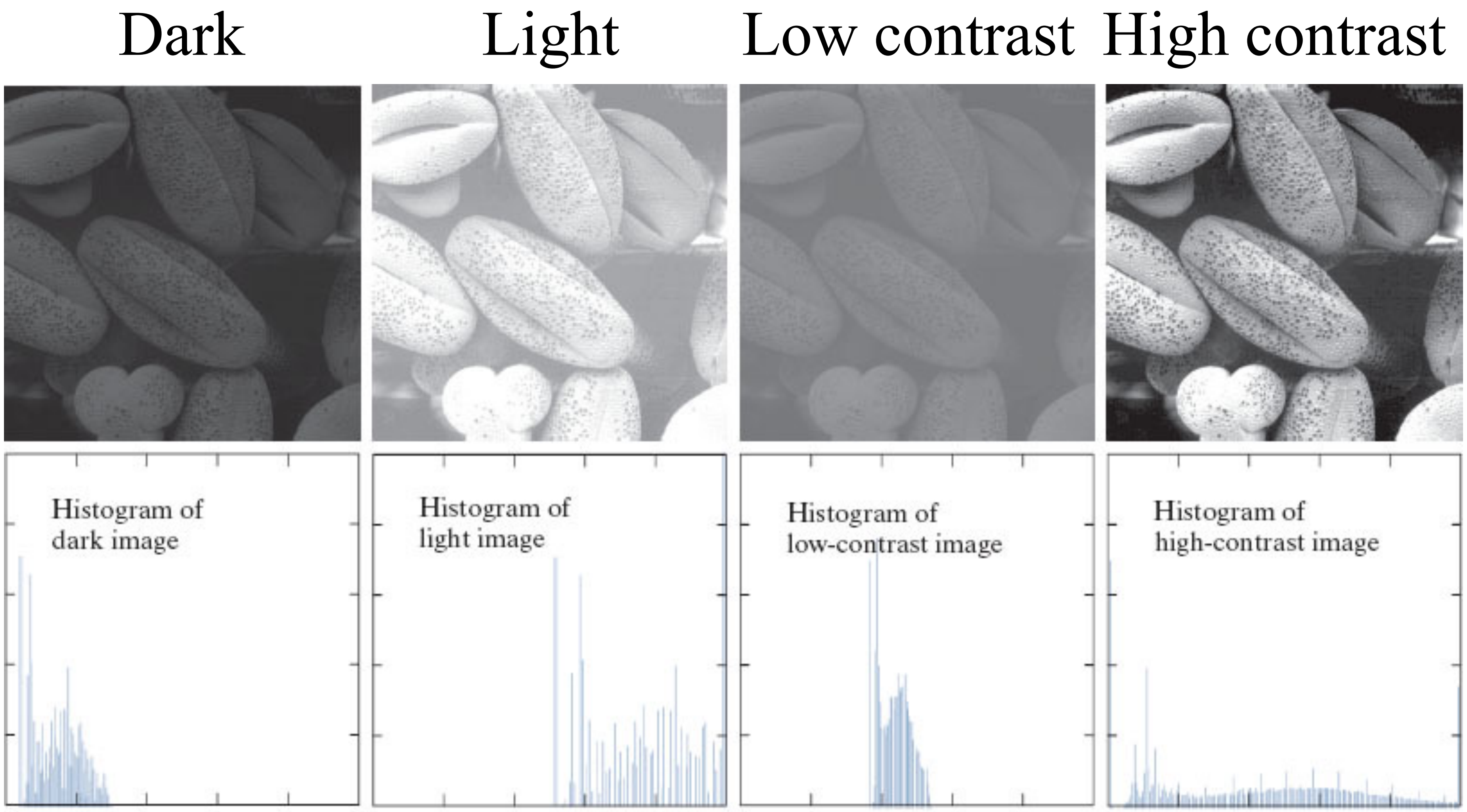
Planes 8, 7, 6, and 5



Original image

Less bit planes are sufficient to obtain an acceptable details, while require half of the storage

Histogram processing

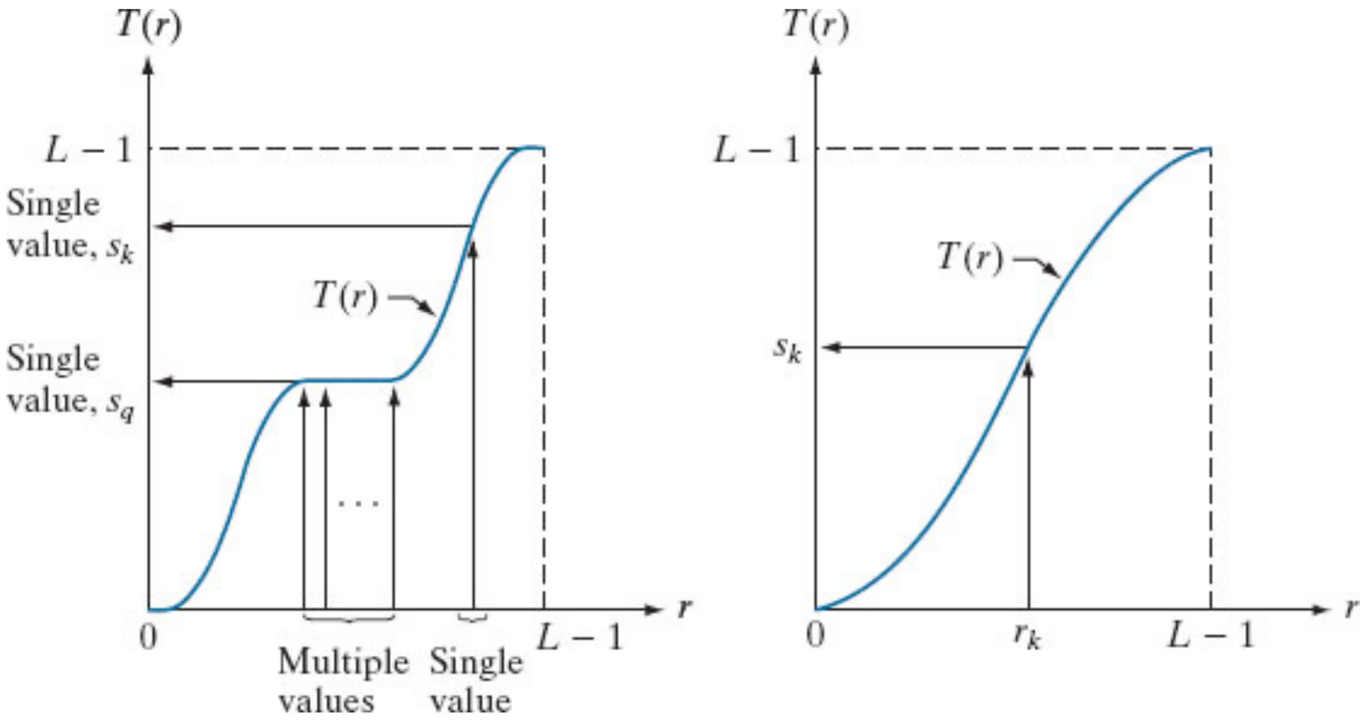


- Histogram
  $h(r_k) = n_k$

- Normalized histogram
  - $p(r_k) = \frac{n_k}{MN}$
  - $\sum_0^{255} p(r_k) = 1$

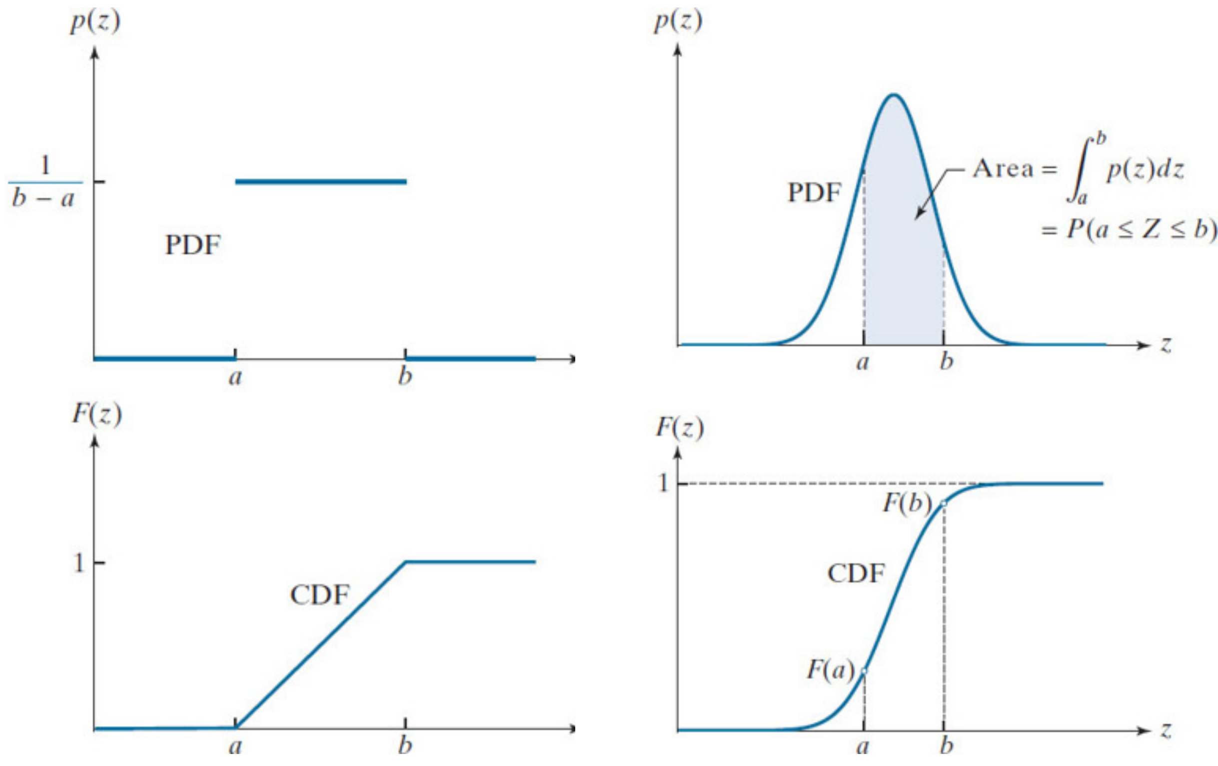
Transformation function

- $s = T(r) \quad 0 \le r \le L - 1$
- A valid transformation function must satisfy two conditions:
  - (a) $T(r)$ is monotonically increasing, i.e., $T(r_1) \ge T(r_2)$ if $r_1 > r_2$
  - (b) $0 \le T(r) \le L - 1$, i.e., the same range as input
  - (a') $T(r)$ is strictly monotonic: one-t—one mapping $r = T'(s)$



PDF & CDF

- Probability density function
  - Likelihood of a random variable taking on a specific value within a continuous range
    - $p(x) \geq 0$
    - $\int_{-\infty}^{\infty} p(x) = 1$
    - $P(a \leq x \leq b) = \int_{a}^{b} p(x)$
- Cumulative distribution function
  - Cumulative probability that a random variable X is less than or equal to a given value x: $F(x) = \int_{-\infty}^{x} p(t)dt$
    - $F(x)$ is a non-decreasing function
    - $F(x) \in [0,1]$ for all x



Histogram processing

- If $T(r)$ is continuous and differentiable over the range of r, then

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

Probability density function of intensity value

$s = T(r)$

$P(s \in [s, s + ds]) = P(r \in [r, r + dr])$

$p_s(s)ds = p_r(r)dr$

Histogram equalization

- A special transformation function

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

Cumulative distribution function of r

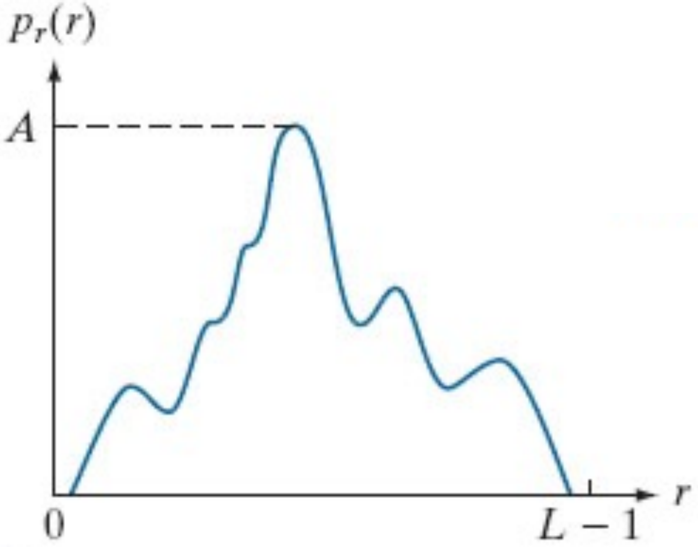
Is it a valid transformation function?

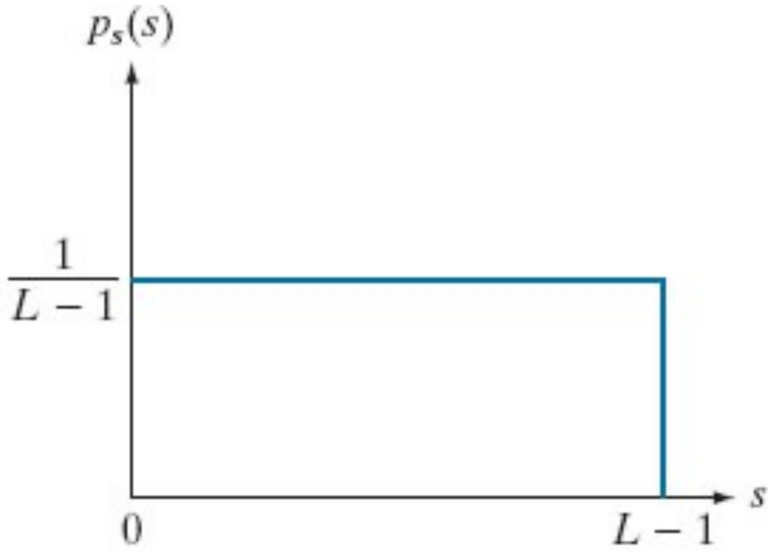
(a) $T(r)$ is monotonically increasing, i.e., $T(r_1) \geq T(r_2)$ if $r_1 > r_2$
(b) $0 \leq T(r) \leq L-1$, i.e., the same range as input

Histogram equalization

$s = T(r) = (L-1)\int_0^r p_r(w)dw \rightarrow p_s(s) = \frac{1}{L-1}$

How to prove it?





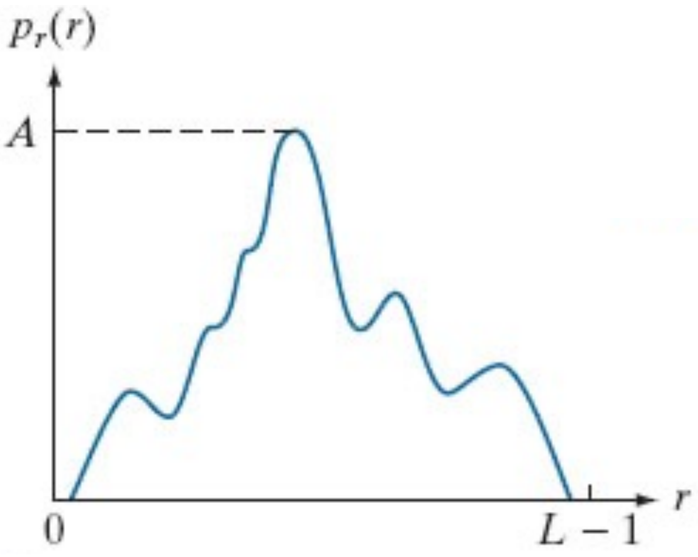
Histogram equalization

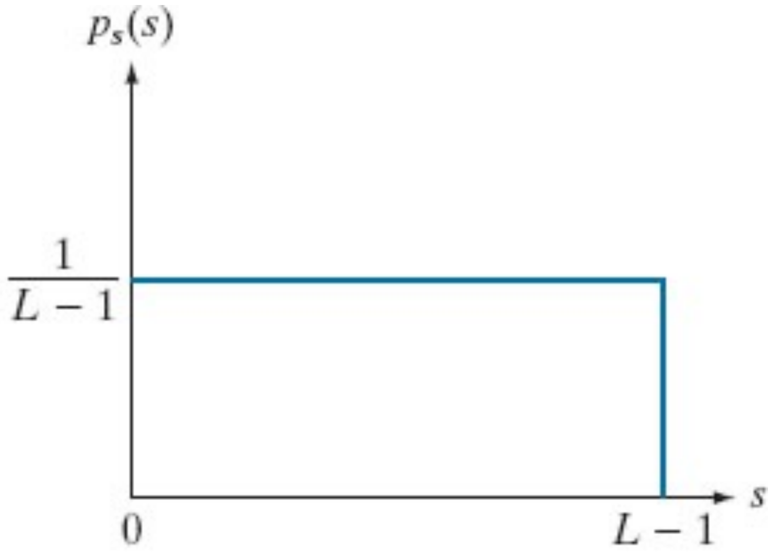
$s = T(r) = (L-1)\int_0^r p_r(w)dw \rightarrow p_s(s) = \frac{1}{L-1}$

How to prove it?

$$\frac{ds}{dr} = \frac{dT(r)}{dr}$$
$$= (L-1)\frac{d}{dr}\left[\int_0^r p_r(w)dw\right]$$
$$= (L-1)p_r(r)$$

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





An example

Suppose that the (continuous) intensity values in an image have the PDF

$$p_r(r) = \begin{cases} \dfrac{2r}{(L-1)^2} & \text{for } 0 \le r \le L-1 \\ 0 & \text{otherwise} \end{cases}$$

$$s = T(r) = (L-1)\int_0^r p_r(w)\,dw = \frac{2}{L-1}\int_0^r w\,dw = \frac{r^2}{L-1}$$

$$p_s(s) = p_r(r)\left|\frac{dr}{ds}\right| = \frac{2r}{(L-1)^2}\left|\left[\frac{ds}{dr}\right]^{-1}\right|$$

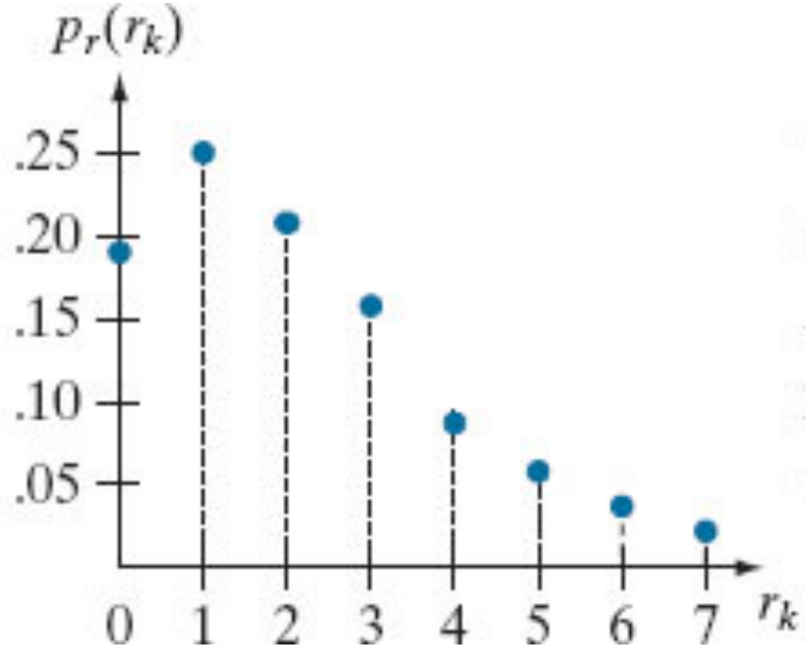
$$= \frac{2r}{(L-1)^2}\left|\left[\frac{d}{dr}\frac{r^2}{L-1}\right]^{-1}\right| = \frac{2r}{(L-1)^2}\left|\frac{(L-1)}{2r}\right| = \frac{1}{L-1}$$

Histogram equalization – discrete case

$$p_r(r_k) = \frac{n_k}{MN} \quad k = 0,1,2,\dots,L-1$$

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

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



Intensity distribution and histogram values for a 3-bit 64x64 digital image

Histogram equalization – discrete case

$$p_r(r_k) = \frac{n_k}{MN} \quad k = 0,1,2,\dots,L-1$$

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

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

Intensity distribution and histogram values for a 3-bit 64x64 digital image

$$s_0 = T(r_0) = (L-1)\sum_{j=0}^{0} p_r(r_j)$$

$$= (8-1) * 0.19 = 1.33 \rightarrow 1$$

$$s_7 = T(r_7) = (L-1)\sum_{j=0}^{7} p_r(r_j)$$

$$= (8-1) * 1 = 7$$

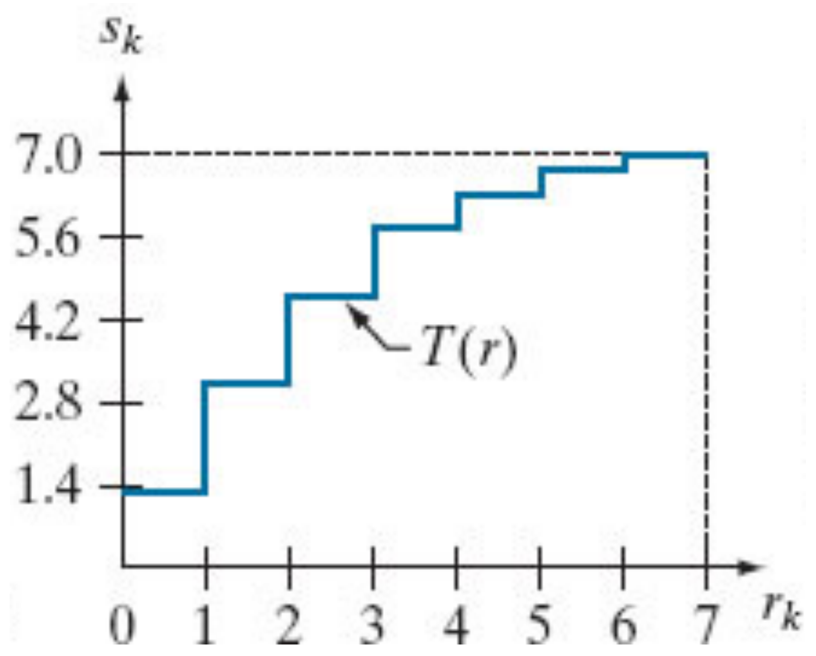
Histogram equalization – discrete case

$$p_r(r_k) = \frac{n_k}{MN} \quad k = 0,1,2,\ldots,L-1$$

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

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

Intensity distribution and histogram values for a 3-bit 64x64 digital image



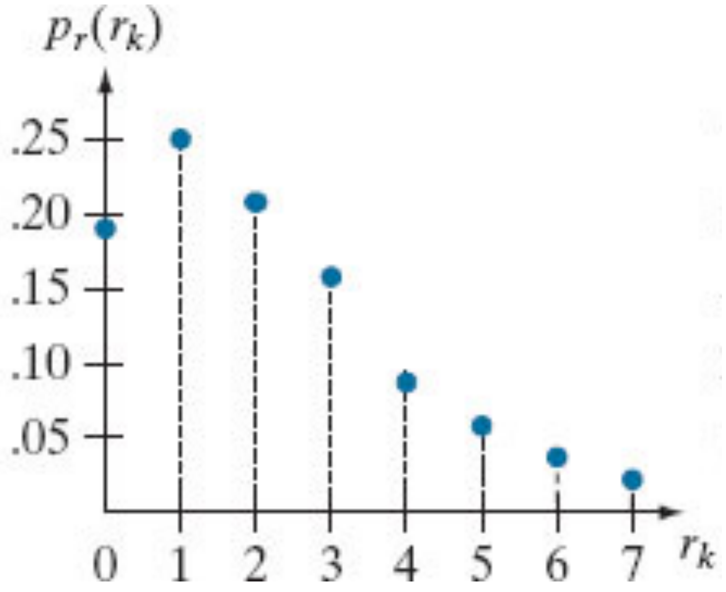
Histogram equalization – discrete case

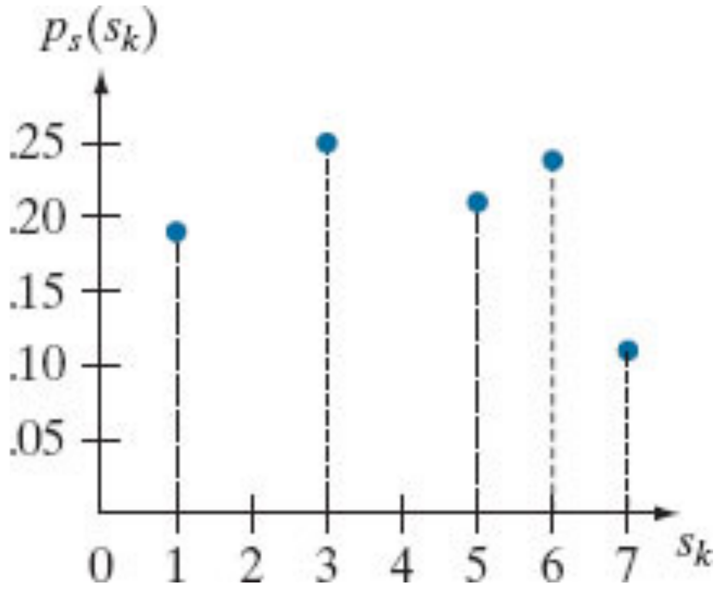
$$p_r(r_k) = \frac{n_k}{MN} \quad k = 0,1,2,\ldots,L-1$$

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

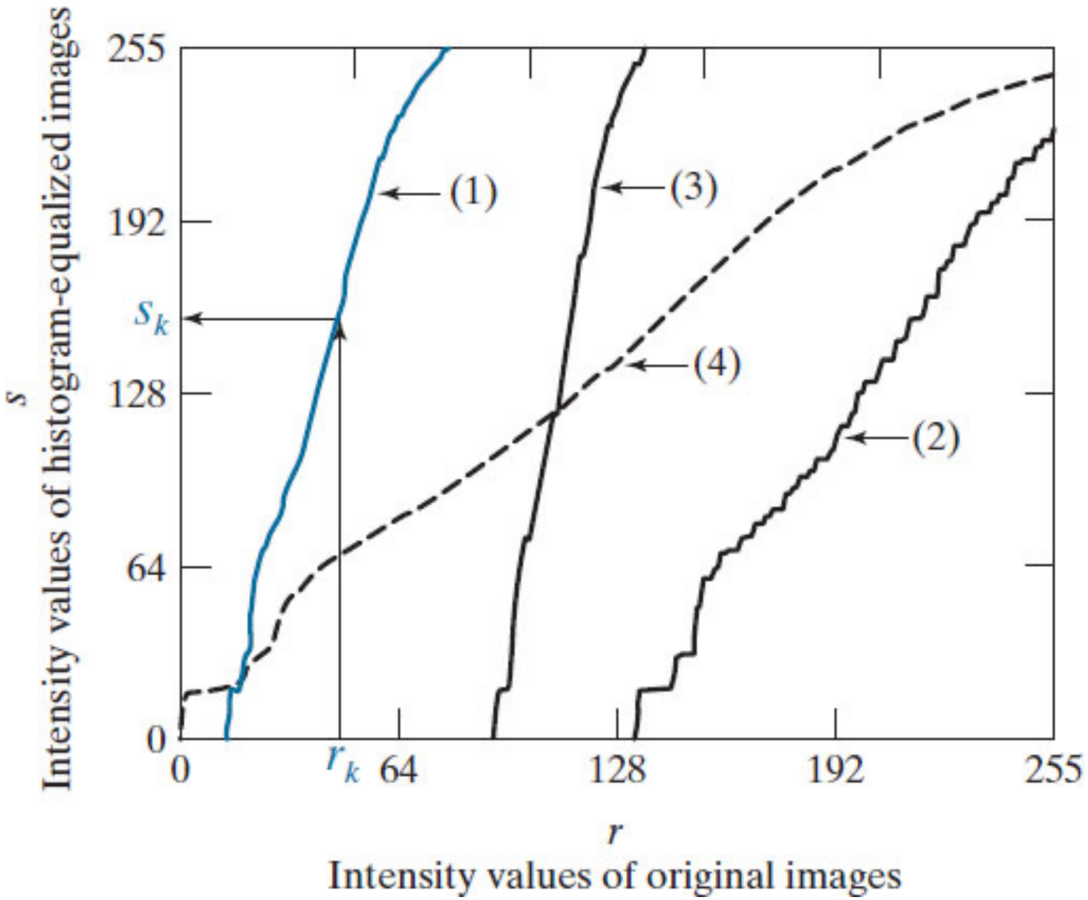
r_k	n_k	$p_r(r_k) = n_k/MN$
$r_0 = 0$	790	0.19
$r_1 = 1$	1023	0.25
$r_2 = 2$	850	0.21
$r_3 = 3$	656	0.16
$r_4 = 4$	329	0.08
$r_5 = 5$	245	0.06
$r_6 = 6$	122	0.03
$r_7 = 7$	81	0.02

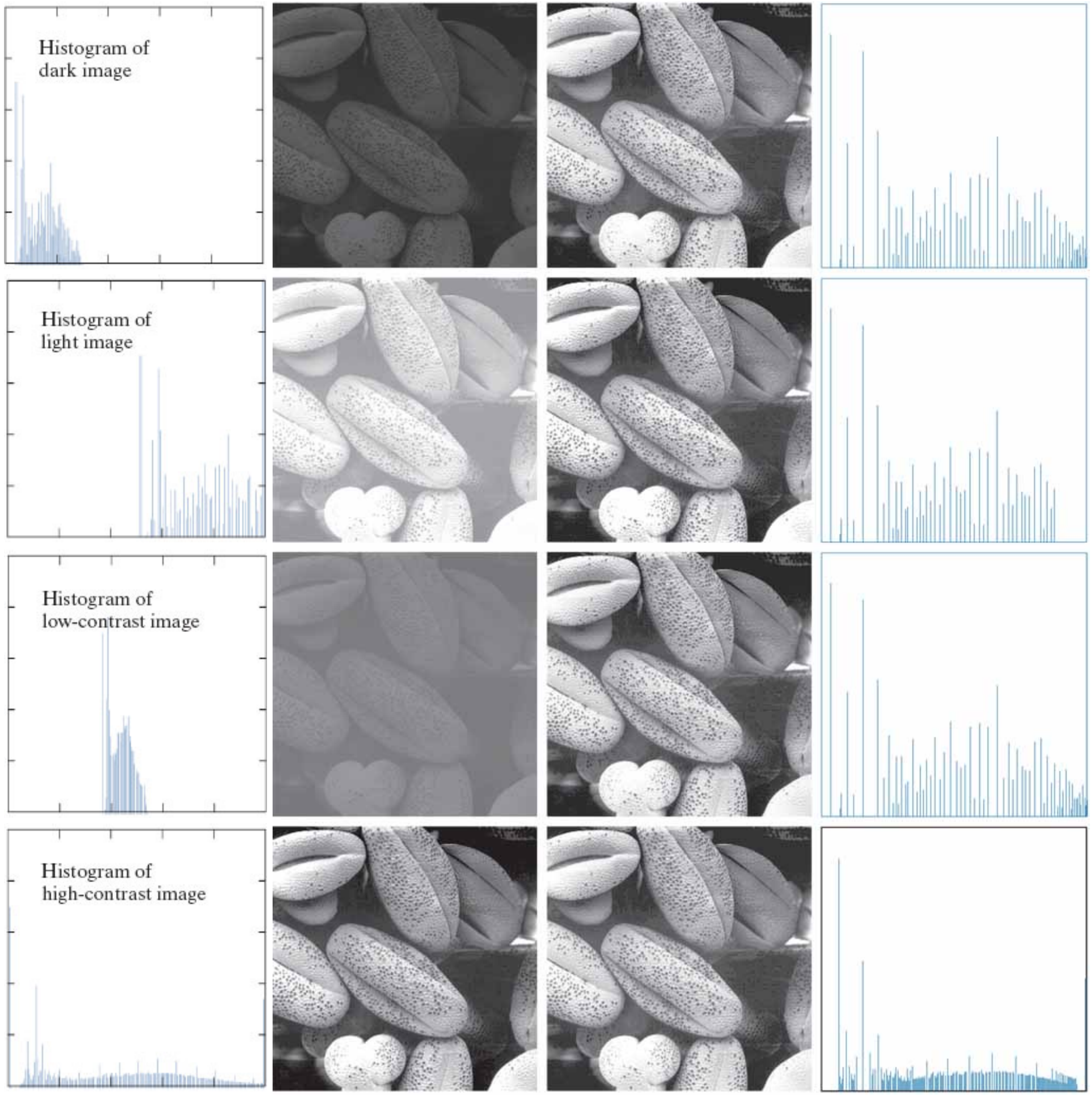
Intensity distribution and histogram values for a 3-bit 64x64 digital image





Examples

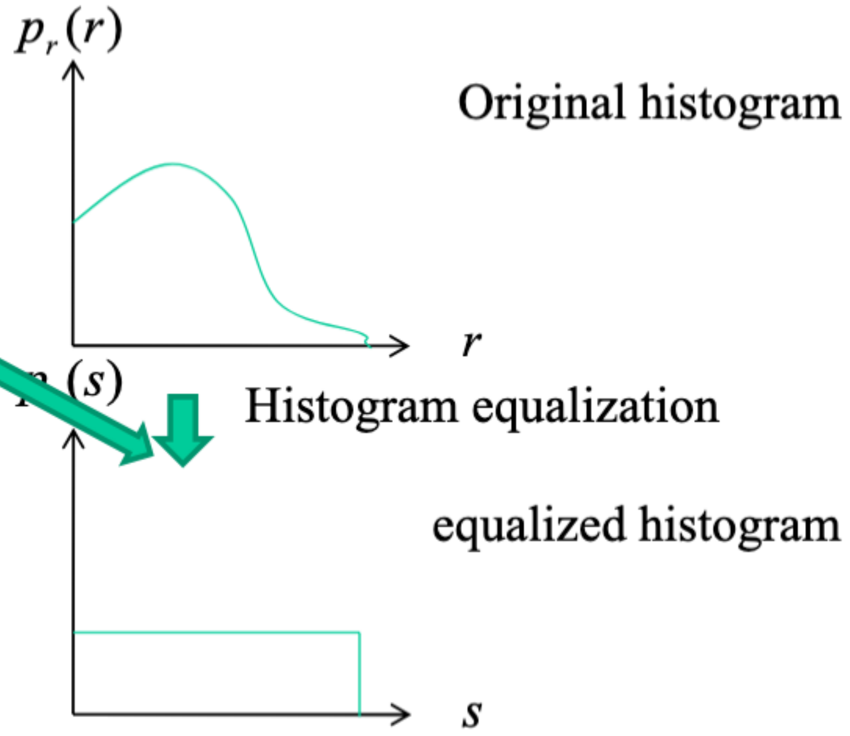

255
192
128
64
0
s
Intensity values of histogram-equalized images
s_k
(1)
(2)
(3)
(4)
0
r_k
64
128
192
255
r
Intensity values of original images


Histogram of dark image
Histogram of light image
Histogram of low-contrast image
Histogram of high-contrast image

Histogram matching (specification)

Estimated pdf in the input image

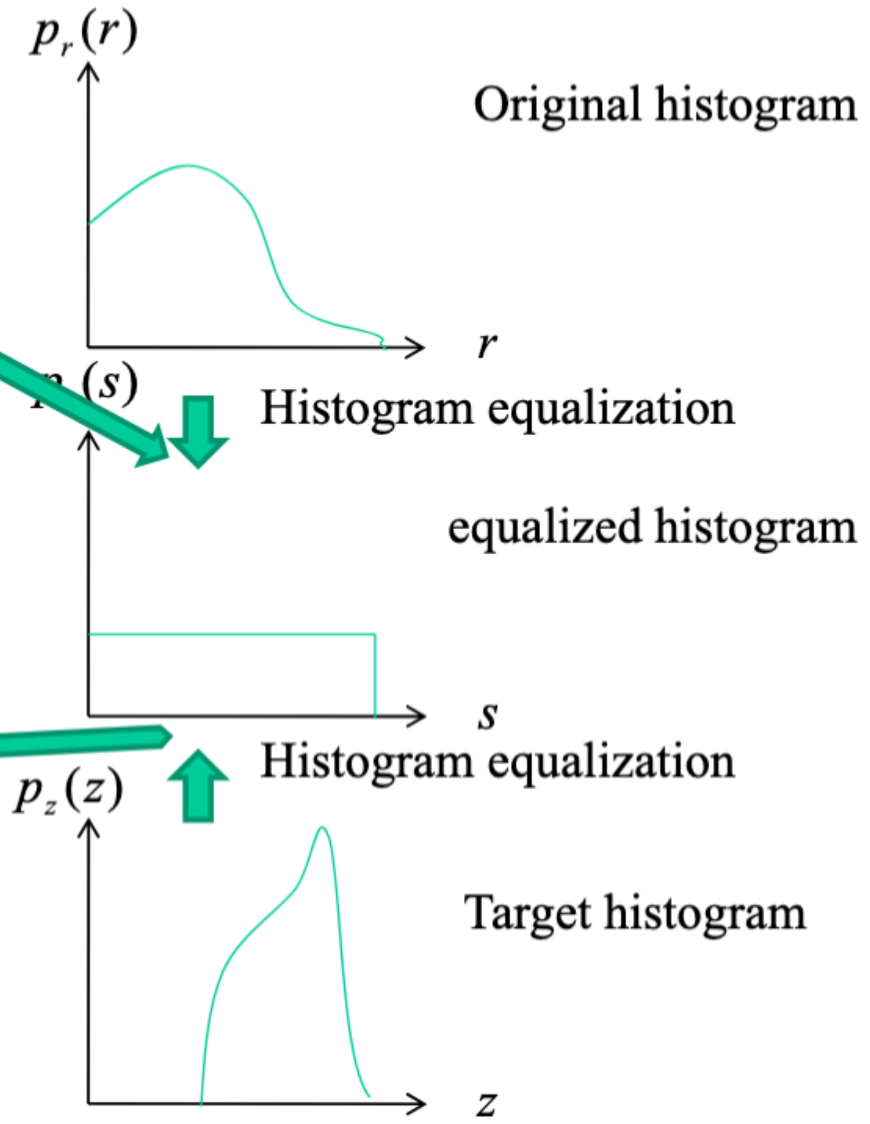
$$s = T(r) = (L-1)\int_0^r \underset{\uparrow}{p_r(w)}\,dw$$



Histogram matching (specification)

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

$$s = G(z) = (L-1)\int_0^z p_z(t)\,dt$$



Histogram matching for continuous data

- Compute the probability distribution function of input data $p_r(r)$
- Perform histogram equalization $\rightarrow s = T(r)$

- Compute $s = G(z)$, where G is the equalization function derived from a specified histogram
- Perform the inverse mapping $z = G^{-1}(s) = G^{-1}(T(r))$

- The output image with z values is then of the specified histogram

Histogram matching for discrete data

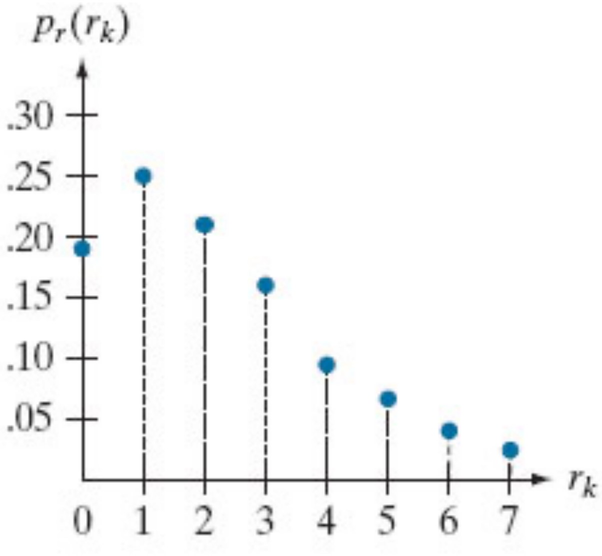
- Compute the probability distribution function of input data $p_r(r)$
- Perform histogram equalization $\rightarrow s = T(r)$

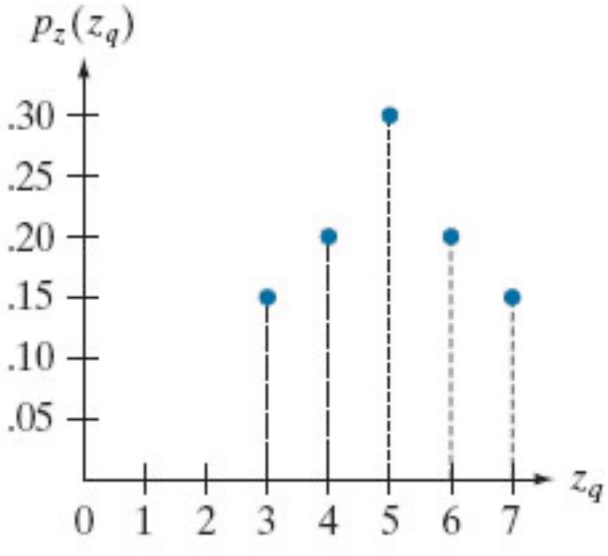
- Compute $s = G(z)$, where G is the equalization function derived from a specified histogram
- ~~Perform the inverse mapping~~ $\cancel{z = G^{-1}(s) = G^{-1}(T(r))}$

Given the s_k value, find the value of z_q
so that $G(z_q)$ is closest to s_k

- The output image with z values is then of the specified histogram

A discrete example



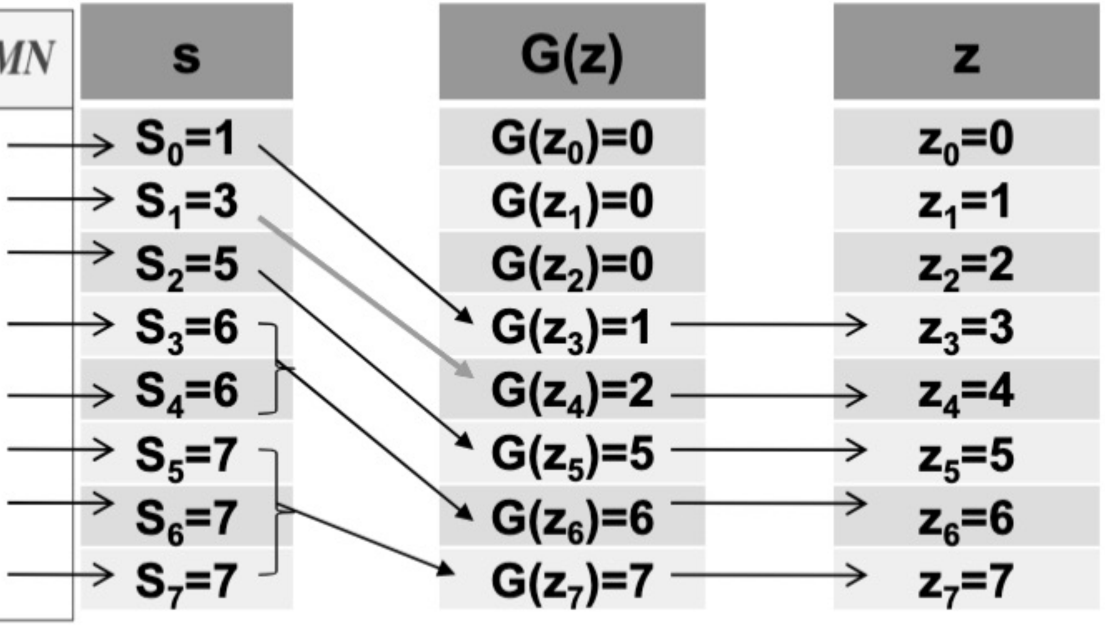


r_k	n_k	$p_r(r_k) = n_k/MN$
$r_0 = 0$	790	0.19
$r_1 = 1$	1023	0.25
$r_2 = 2$	850	0.21
$r_3 = 3$	656	0.16
$r_4 = 4$	329	0.08
$r_5 = 5$	245	0.06
$r_6 = 6$	122	0.03
$r_7 = 7$	81	0.02

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

A discrete example

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



$$r_0 \rightarrow z_3$$

$$r_1 \rightarrow z_4$$

$$r_2 \rightarrow z_5$$

$$r_3, r_4 \rightarrow z_6$$

$$r_5, r_6, r_7 \rightarrow z_7$$

z_q	Specified $p_z(z_q)$	Actual $p_z(z_k)$
$z_0 = 0$	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

Two types of digital image processing

- Pixel-level processing (intensity transformation)
  - Same function applied to every pixel
    - Thresholding – Logistic function
    - Log transformation
    - Power-law (Gamma correction)
    - Piecewise-linear transformation
    - Histogram processing

- Patch-level processing (filtering)
  - Same filter applied to sub-regions/patches