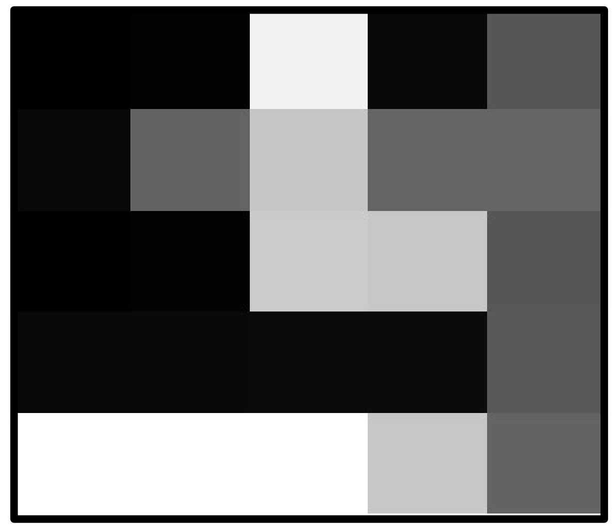
Image processing

Intensity tranformations

Pixel intensities

0	3	243	9	88
9	100	199	101	102
0	3	204	200	88
9	10	11	12	90
255	255	255	200	100



From 0 black to 255 white

*** We only consider grayscale images ***

Intensity Transformations

- Digital image f: $f(x,y)$

  where: $x = 0,\ldots,M-1$
  and: $y = 0,\ldots,N-1$

- We want new image g: $g(x,y) = T[f(x,y)]$

- T: Operator on intensities of f

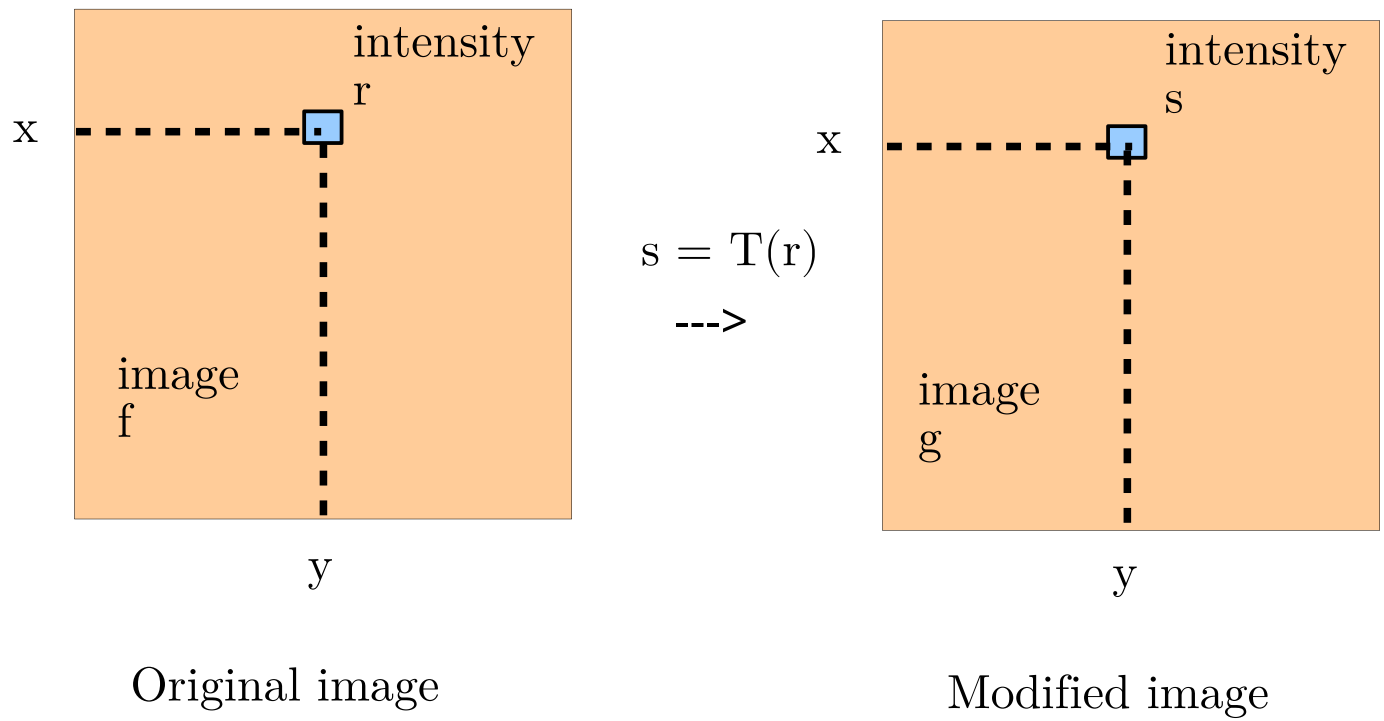
Operator acts on

- Single pixel at (x,y) ---> point-processing
- Neighborhood around (x,y) --> convolution (we will see that later on)

Intensity Transformation are Point-Processing

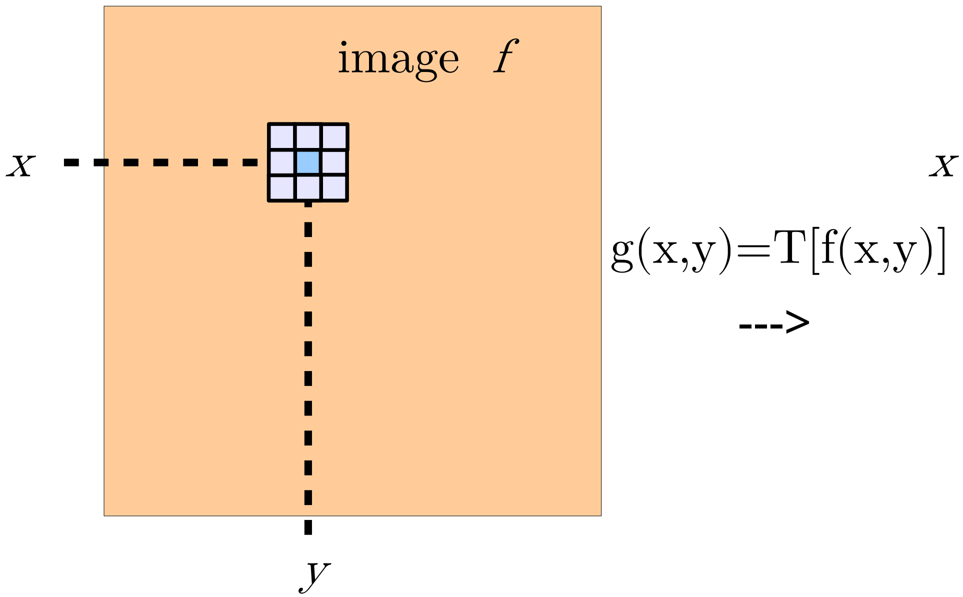
- 2 images f (original image) and g (modified image):
- r: Intensity of f at point (x,y)
- s: Intensity of g at point (x,y)
- Intensity transformation: $s = T(r)$

Operator on single point



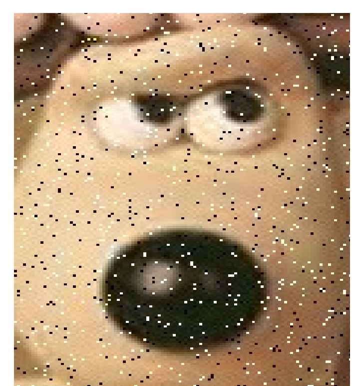
T does not change with position, T independent of (x,y)

Operator on neighborhood (convolution)



Example application: denoising
Information about the neighborhood is useful

We will see that in more details next week



Intensity Transformation Functions

1. Image negatives
2. Log transformations
3. Power-law (gamma) transformations
4. Contrast streching

Principle and some functions which modify the pixel intensity

whiter

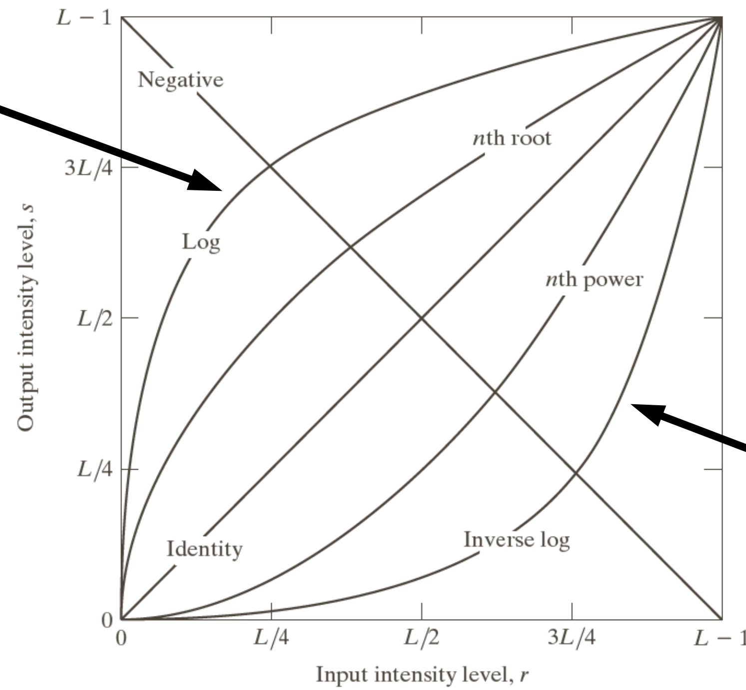


FIGURE 3.3 Some basic intensity transformation functions. All curves were scaled to fit in the range shown.

darker

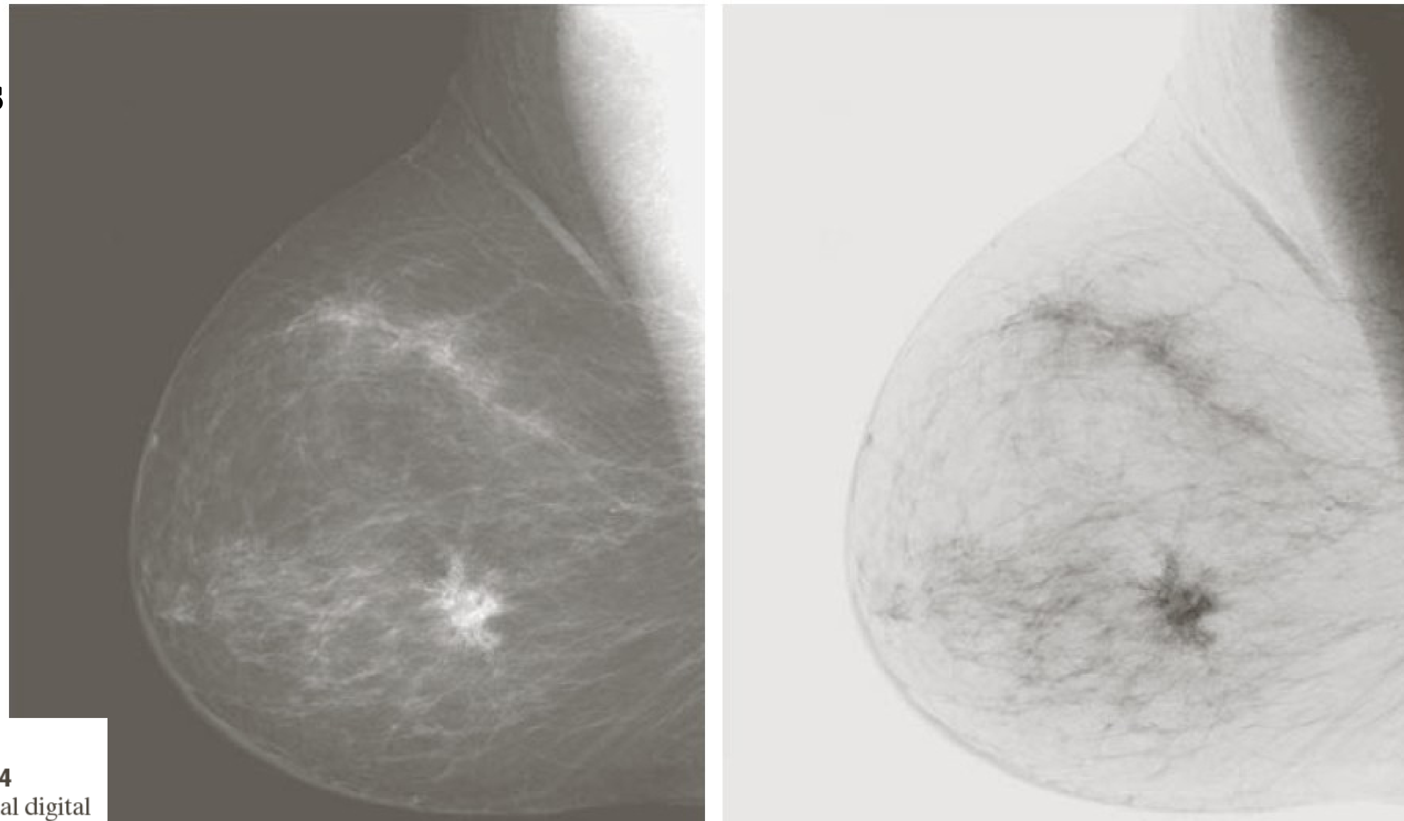
Usually:

The range of r should equal the range of s

1. Image Negatives

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

Enhancing white or gray detail in dark areas

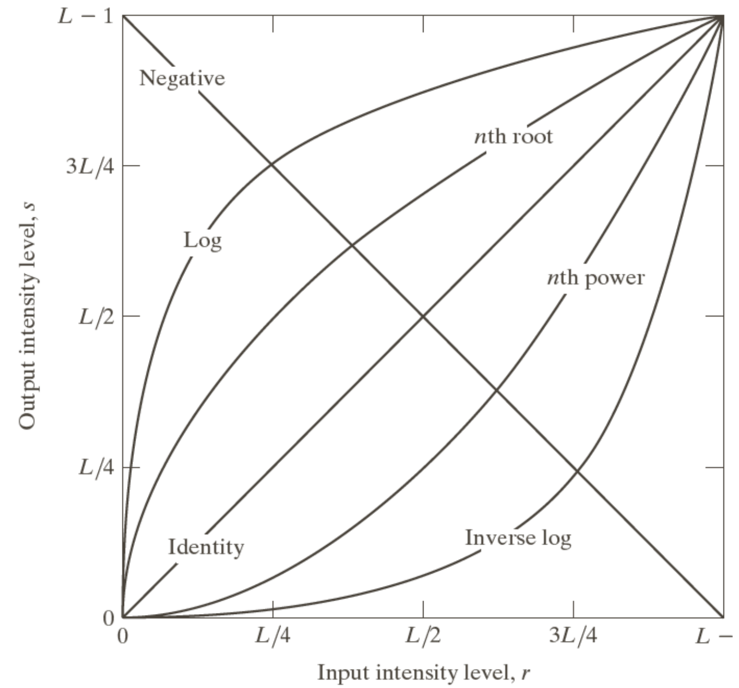


a b

FIGURE 3.4 (a) Original digital mammogram. (b) Negative image obtained using the negative transformation in Eq. (3.2-1). (Courtesy of G.E. Medical Systems.)

2. Log Transformation

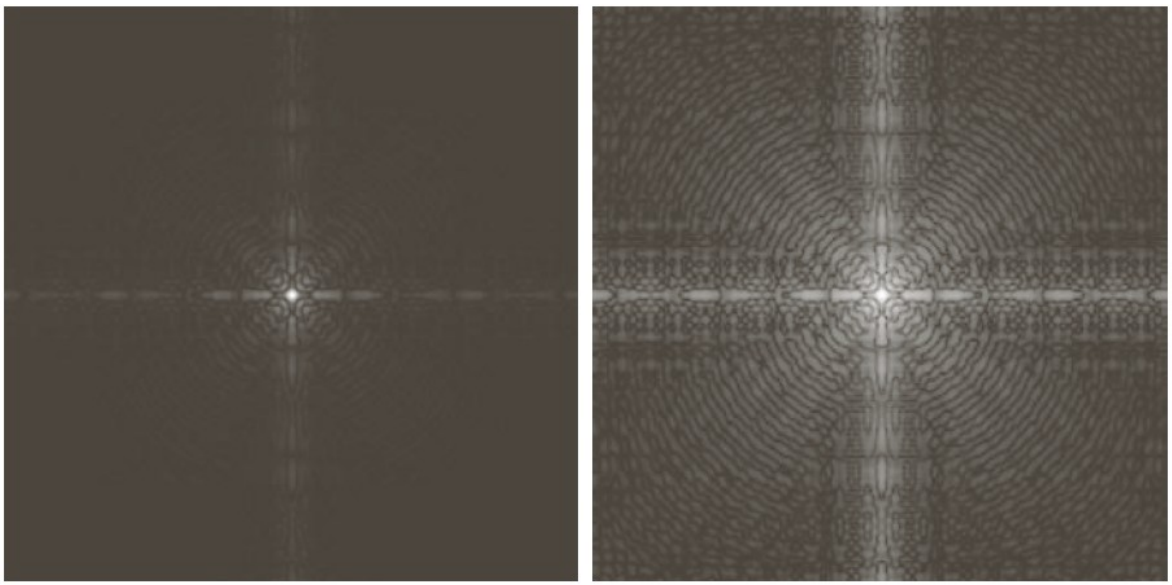
- $s = T(r) = c \log(1 + r)$



- Reduce the importance of high values
- Maps a narrow low intensity range into a broader range
- Maps a broad high intensity region into a narrower range

What is c here?

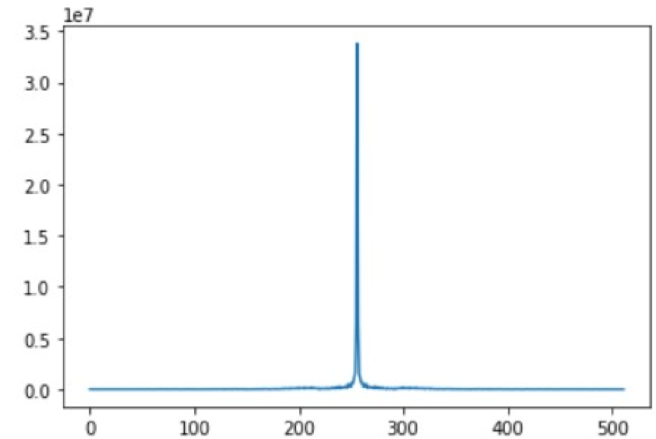
Fourier spectrum log-transformed:



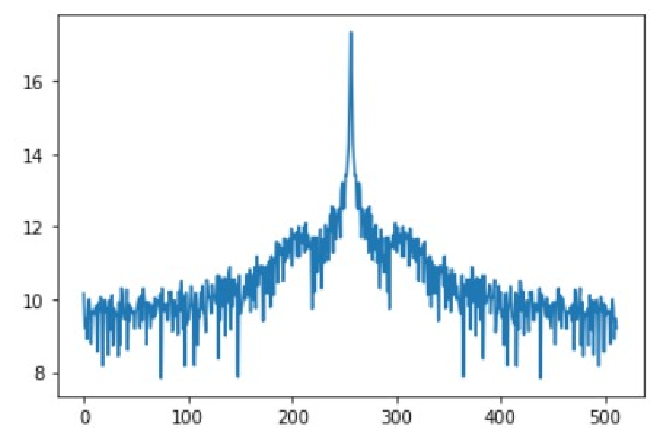
a b

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

```
plt.plot(signal)
plt.show()
```




```
plt.plot(np.log(signal))
plt.show()
```



Very high values can mask important lower values. (more details later on)

3. Power-Law (Gamma) Transformation

- $s = T(r) = c\, r^{\gamma}$

- More versatile function by varying gamma
- Useful in contrast manipulation
- Also an inherent component of display systems
  - Gamma correction

$$s = T(r) = c\,r^{\gamma}$$

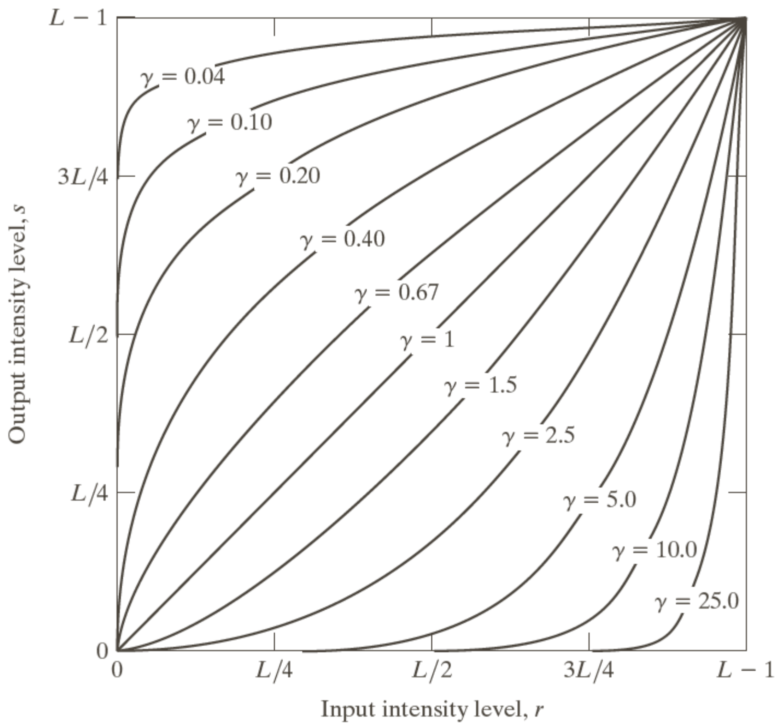
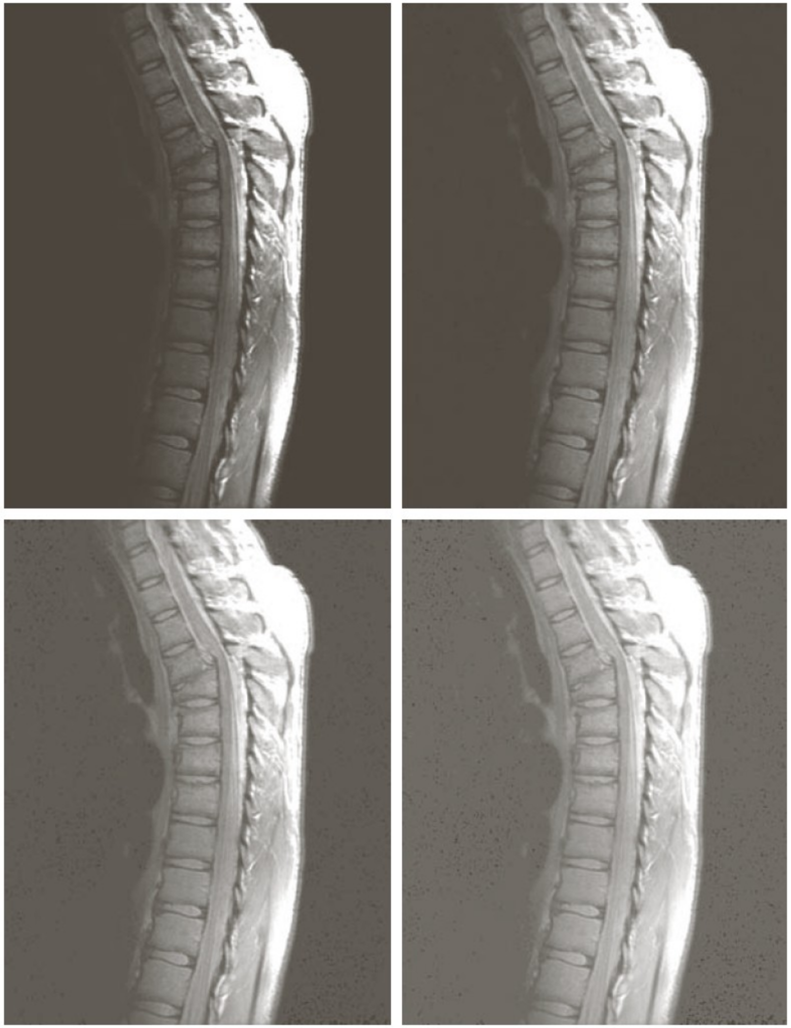


FIGURE 3.6 Plots of the equation $s = cr^{\gamma}$ for various values of γ ($c = 1$ in all cases). All curves were scaled to fit in the range shown.

Gamma < 1



a b
c d

FIGURE 3.8 (a) Magnetic resonance image (MRI) 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 courtesy of Dr. David R. Pickens, Department of Radiology and Radiological Sciences, Vanderbilt University Medical Center.)

Gamma > 1

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.)

Gamma Correction

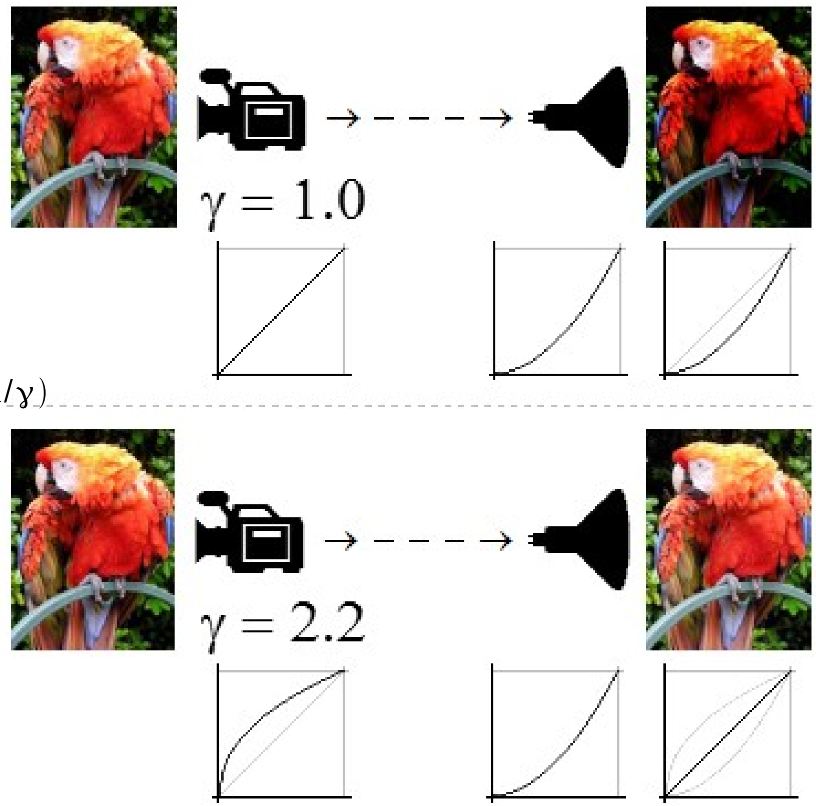
- Display systems voltage-to-intensity are not linear, they follow a power-law
  - Cathode ray tubes
  - LCDs
  - projectors
- Gamma > 1 ---> Images are displayed darker!
- Must be corrected at source

Gamma Correction

- Let

$$s = T(r) = (r^{\gamma})^{(1/\gamma)}$$

$$s = T(r) = r$$



Wikipedia

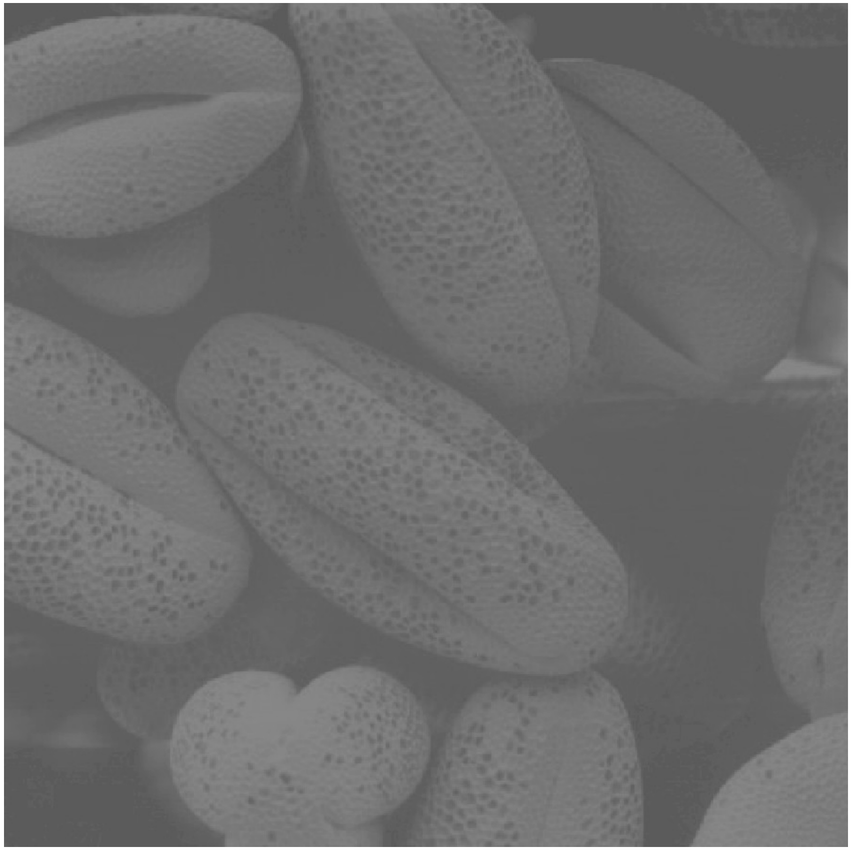
- Example of different gamma encoding
- On a computer screen, the second panel from the top corresponds to the inherent gamma of the display device

Wikipedia

4. Contrast Stretching

- Given an image with intensity levels mostly in the mid-gray region
- Low contrast...
  - Poor illumination
  - Wrong setting of lens, etc
- Washed out appearance

- What to do?



- Need to obtain something like this!
- More contrast in the image

- Need to try to «stretch» out the intensity levels

Mathematical stretching function: s=T(r)

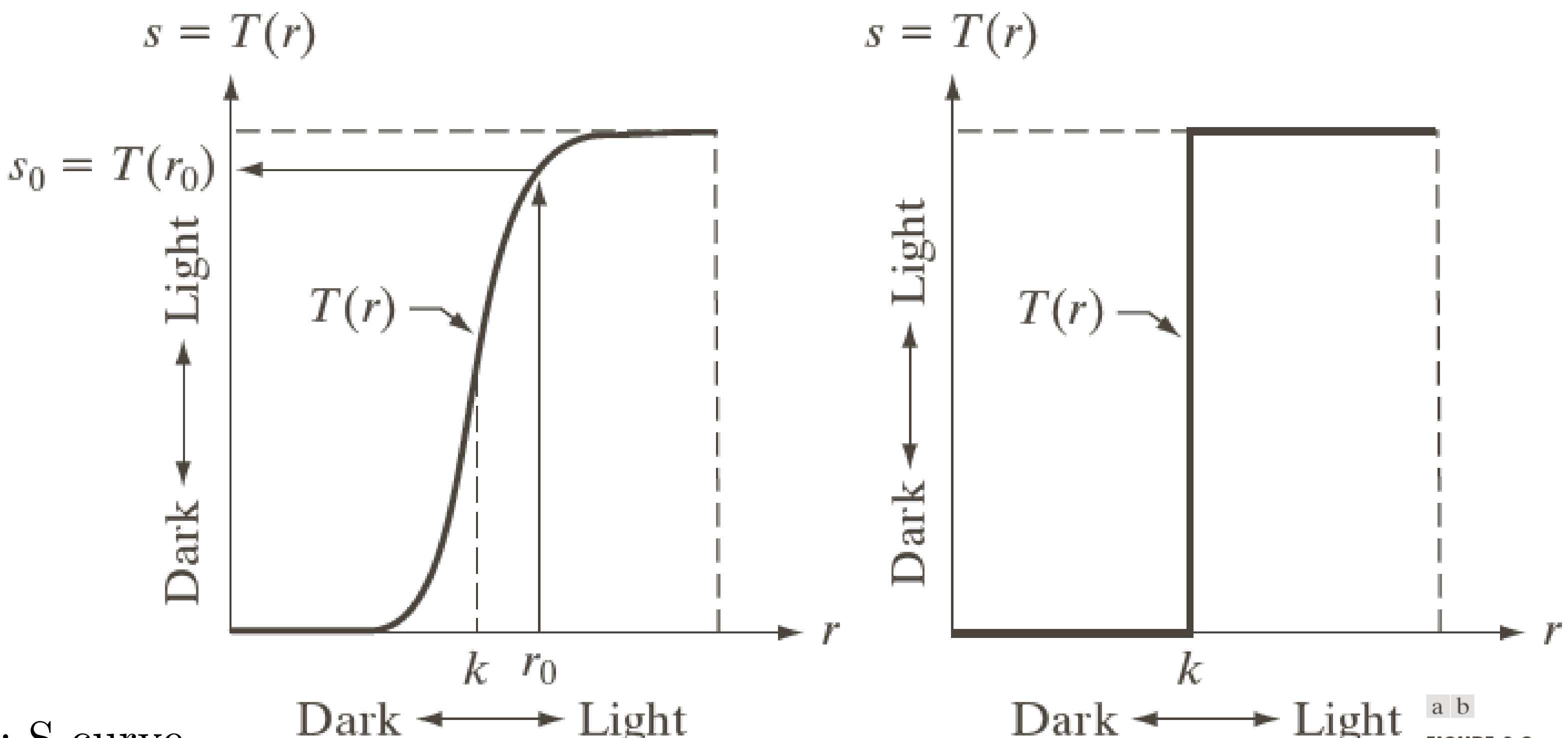


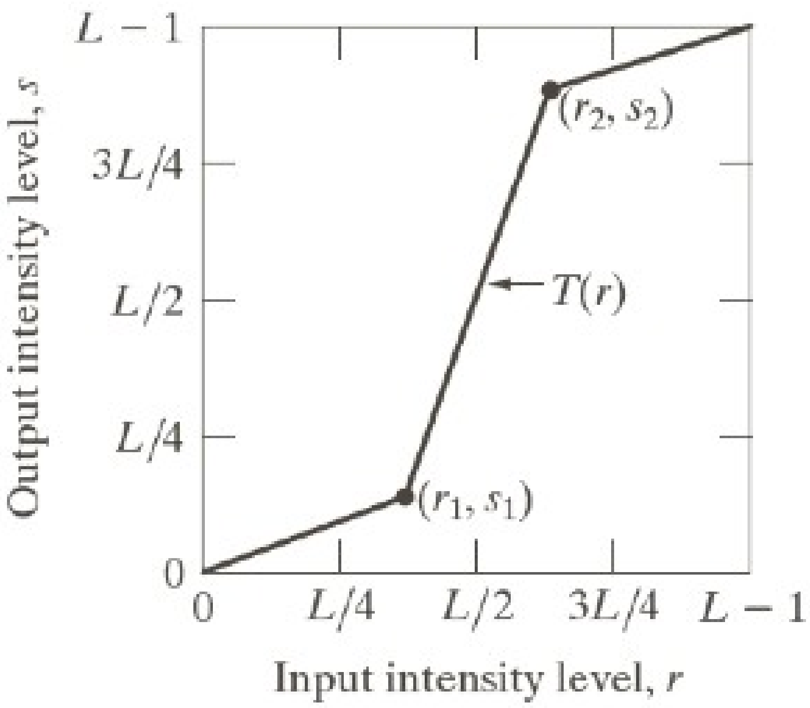
FIGURE 3.2 Intensity transformation functions. (a) Contrast-stretching function. (b) Thresholding function.

- Example: S-curve
- Darkening intensities below k
- Brightening intensities above k
- Limiting case: Thresholding – binary image (two intensities)

Piecewise-linear curve

- Other option: Piecewise-linear stretching curve: s=T(r)
- May be arbitrarily complex
- Requires more user input

- Here: Points (r_1,s_1) and (r_2,s_2) mark different linear functions



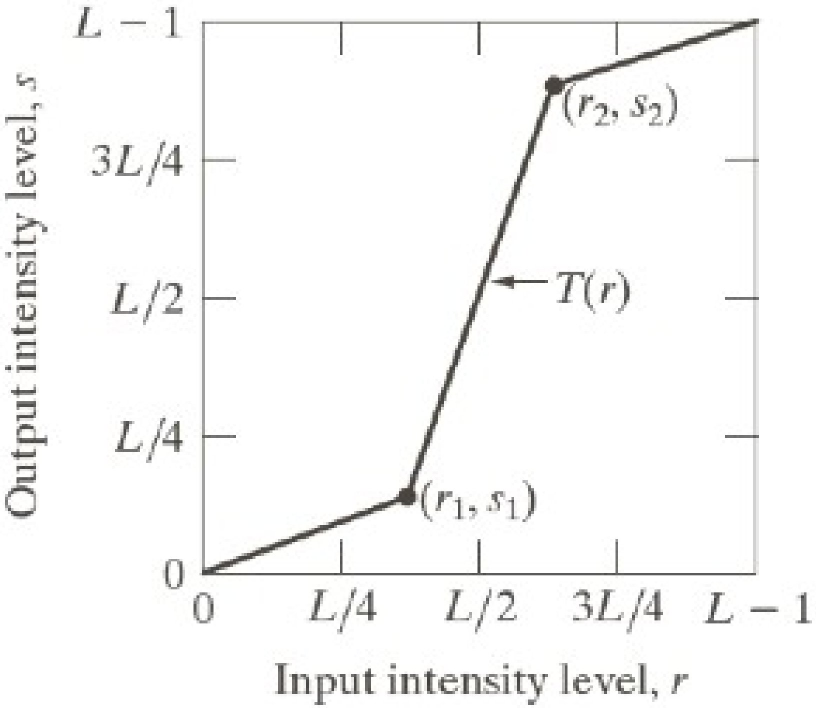
Special Case: Linear

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

Special Case: Thresholding

- $(r_1,s_1) = (m,0)$
- $(r_2,s_2) = (m,L\text{-}1)$

- Here: m ---> mean intensity



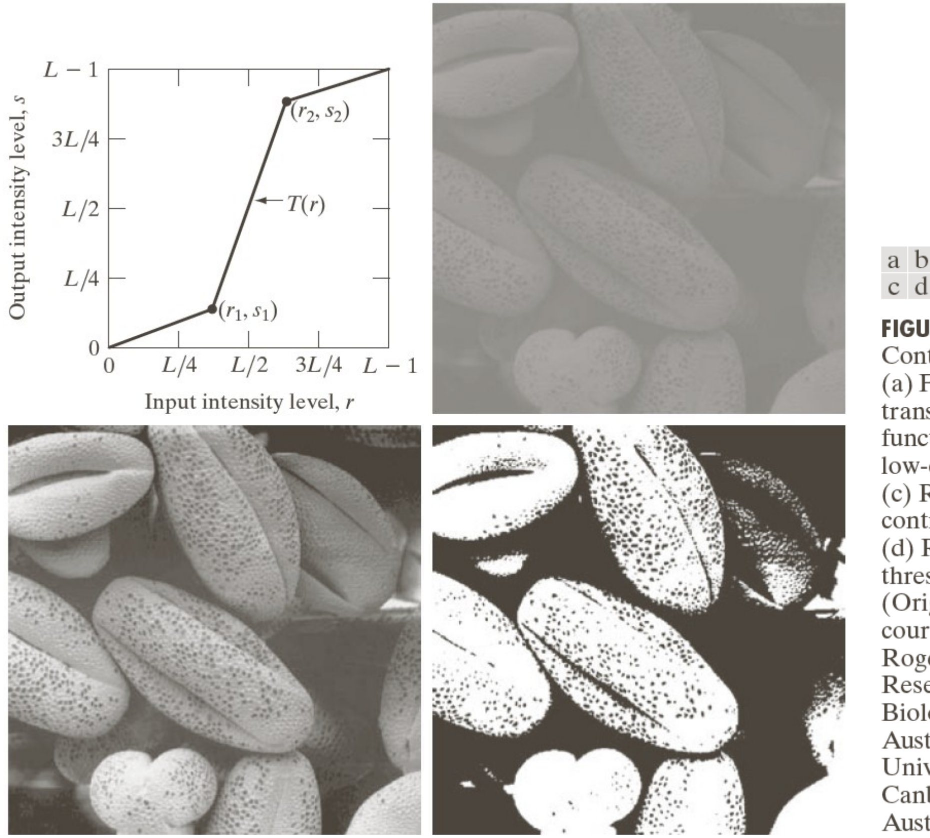
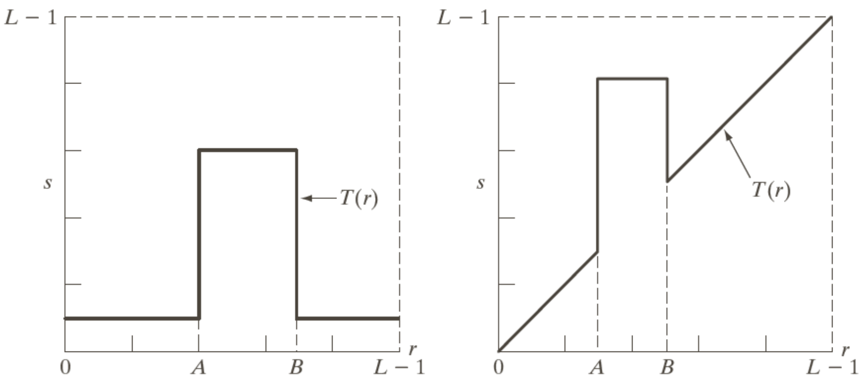


FIGURE 3.10 Contrast stretching. (a) Form of transformation function. (b) A low-contrast image. (c) Result of contrast stretching. (d) Result of thresholding. (Original image courtesy of Dr. Roger Heady, Research School of Biological Sciences, Australian National University, Canberra, Australia.)

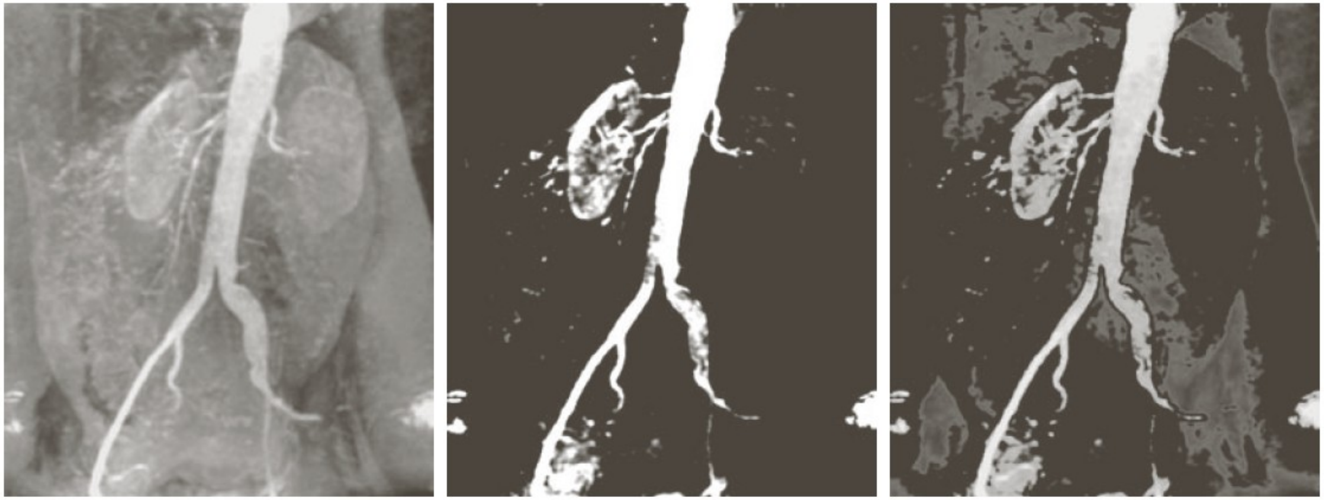
Other Piecewise-Linear Transformations

a b

FIGURE 3.11 (a) This transformation highlights intensity range $[A, B]$ and reduces all other intensities to a lower level. (b) This transformation highlights range $[A, B]$ and preserves all other intensity levels.



1) Intensity slicing: Highlights intensity range [A,B] and reduces all other intensities to a lower level
2) Highlights intensity range [A,B] and preserves all other intensities



a b c

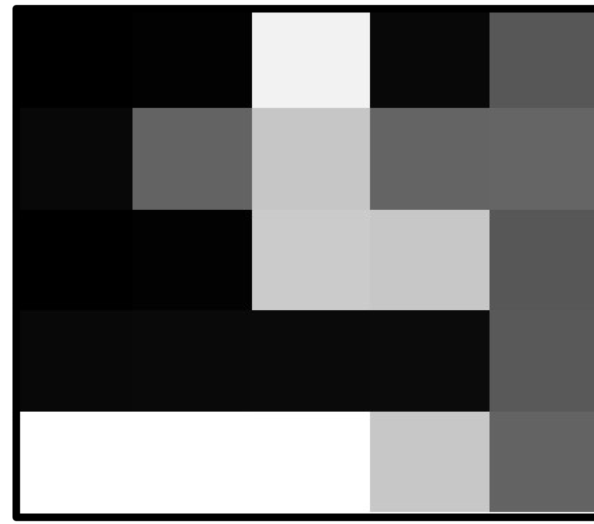
FIGURE 3.12 (a) Aortic angiogram. (b) Result of using a slicing transformation of the type illustrated in Fig. 3.11(a), with the range of intensities of interest selected in the upper end of the gray scale. (c) Result of using the transformation in Fig. 3.11(b), with the selected area set to black, so that grays in the area of the blood vessels and kidneys were preserved. (Original image courtesy of Dr. Thomas R. Gest, University of Michigan Medical School.)

Binary numbers and Intensity Transformation

- Binary numbers encoding for images
- Bit-plane slicing

Binary numbers

0	3	243	9	88
9	100	199	101	102
0	3	204	200	88
9	10	11	12	90
255	255	255	200	100



- Remember intensity range [0,L-1]
- $L = 2^k$ ---> k-bit images
- E.g. $L=2^8 = 256$ ---> intensities in [0,255]

- Why? Intensities are stored as *binary numbers* of k bits
- Bit: 0 or 1

Example: 8-bit binary numbers

- In general: 2 to the power of k-1,...,0

  8 bits:

** 2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0 **
128	64	32	16	8	4	2	1

- Yes or No? Represented by 1 or 0
- Intensity level given by sum of «yes»

- Example: 00000001 is 1 ---> only $2^0 = 1$ present
- 00000110 is 4+2 = 6, 00001010 is 8+2 = 10
- 10000010 ?
- Some bit position changes have an large impact on the value of the number!

More examples of 8-bit numbers

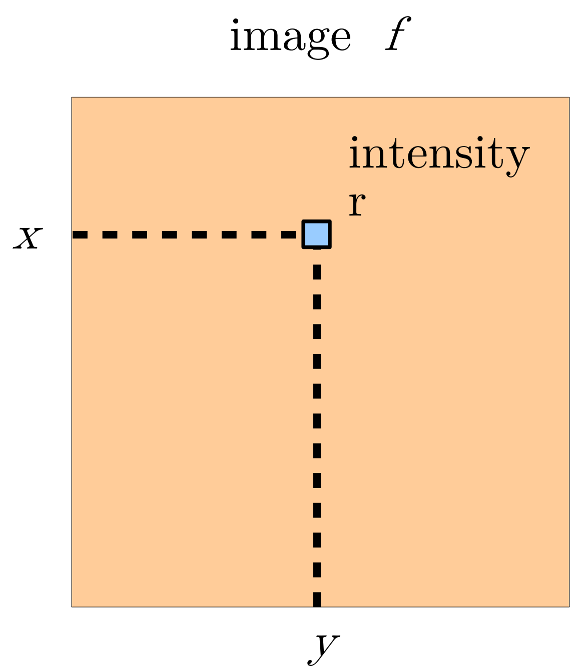
- $00000000 = 0$
- $00000001 = 1$
- $00000010 = 2$
- $00000011 = 3$
- $00000100 = 4$
- $00000101 = 5$

...

- $11111110 = 254$
- $11111111 = 255$

Bit plane slicing

Uses the binary value of the intensity



Example (8-bit images):
r = 129 = 1000 0001

Most significant bit

Least significant bit

Images and Bit-Planes

original image

binary images

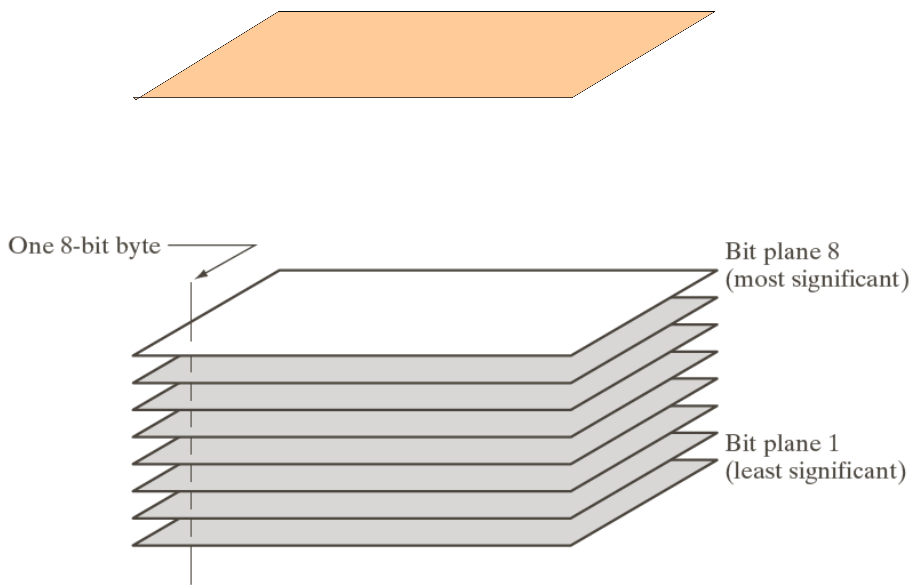


FIGURE 3.13 Bit-plane representation of an 8-bit image.

Should be black & white but grey because of interpolation (we will come back to that)

a	b	c
d	e	f
g	h	i

FIGURE 3.14 (a) An 8-bit gray-scale image of size 500 × 1192 pixels. (b) through (i) Bit planes 1 through 8, with bit plane 1 corresponding to the least significant bit. Each bit plane is a binary image.

- In this example, the most significant bit planes carry most of the visual information
- May reconstruct the image based on these
  - Multiply bit-plane n by 2^{n-1} to get an integer
  - Add the bit-planes used

- Example: Planes 8 and 7
  - Plane 8: 1 ---> 2^7
  - Plane 7: 1 ---> 2^6
  - Add together
  - How many intensities?

a b c

FIGURE 3.15 Images reconstructed using (a) bit planes 8 and 7; (b) bit planes 8, 7, and 6; and (c) bit planes 8, 7, 6, and 5. Compare (c) with Fig. 3.14(a).

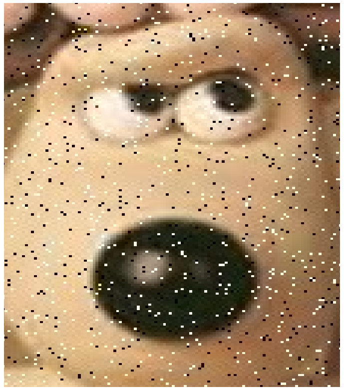
Most significant bits -> larger jumps in values

Least significant bits -> small fluctuations of the intensity (can be ignored)

- Bit-plane slicing is useful for *image compression*
  - Using for example the four most significant bits for storing images saves *50%* space!

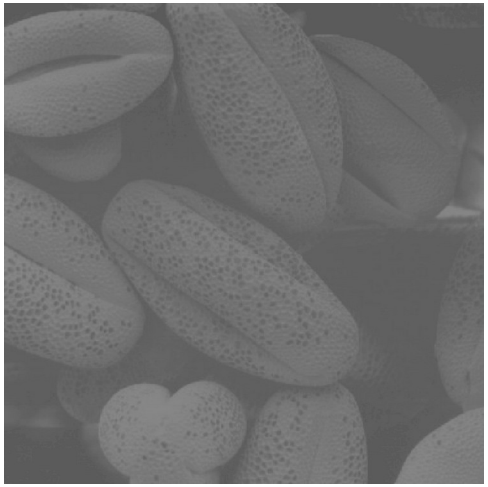
- Encoding pixel intensity can be done in different ways
- Binary encoding gives another point of view on images.

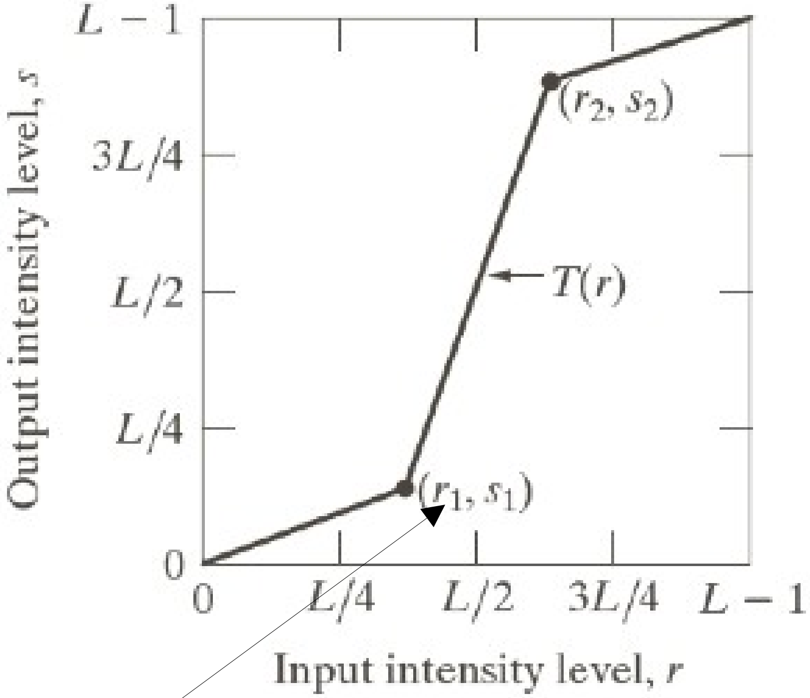
- Intensity transformations are simple and interesting but may be limited
- In the case of noisy images, we need more powerful methods



Next time: Histogram equalization

- Compute the best intensity transformation automatically, from the pixel values
- By Computing distribution of intensities in the image





May be a bit cumbersome to choose these parameters!

Let us find the function automatically!