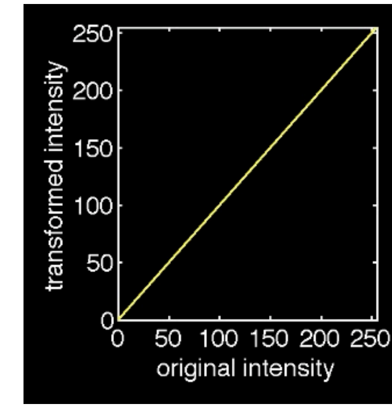
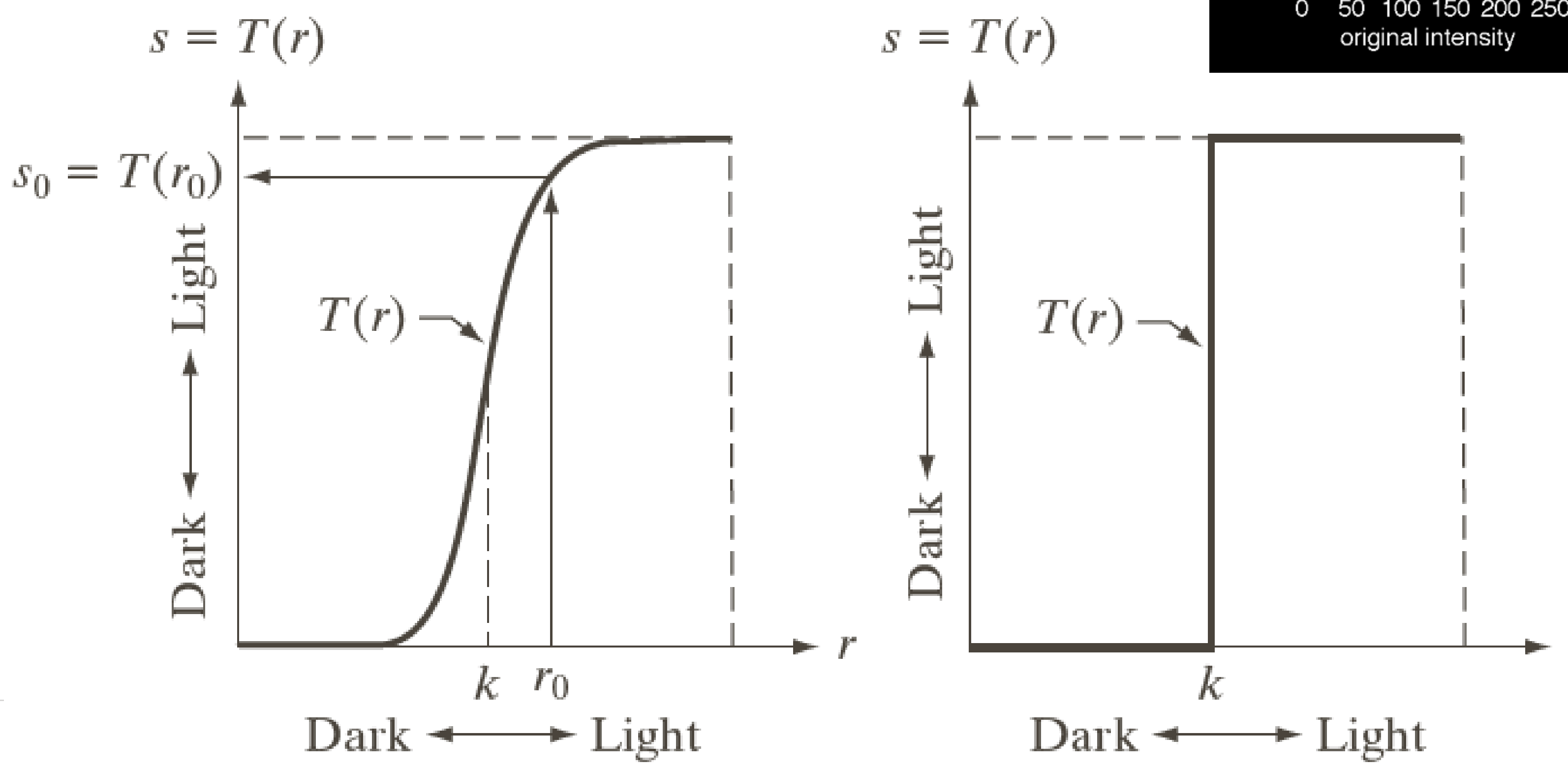
Intensity transformations

Point processing (vs. neighborhood processing)

Gray Level Transformations - Examples

- Map an input value *r* to an output value *s* $$s = T(r)$$
- Examples:
  - Contrast stretching
  - Thresholding





a b

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

Also clips intensities

- Most image editing software (e.g., Adobe Photoshop) allows you to do transformations like this

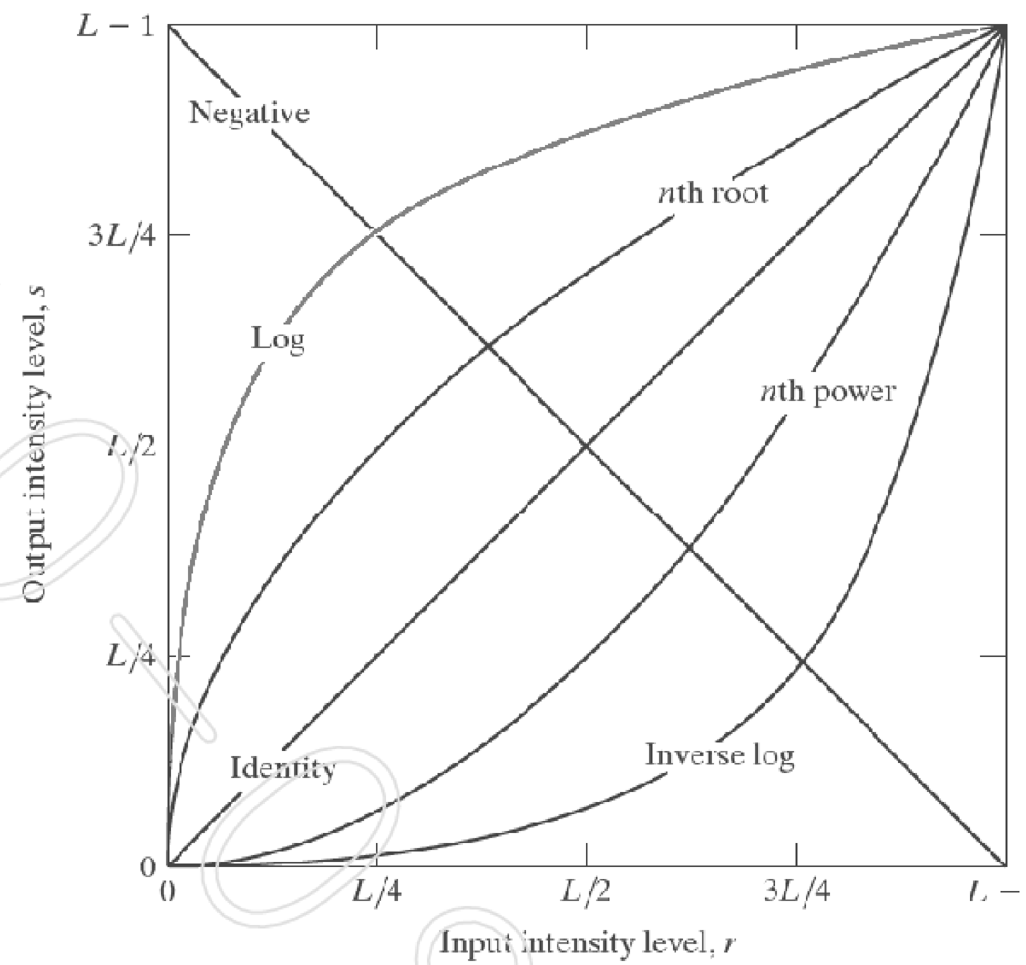
Logarithmic Transform

General form: $$s = T(r) = c \log(1 + r)$$

c = const., r >= 0

- Very low intensities widened
- Bright intensities squeezed

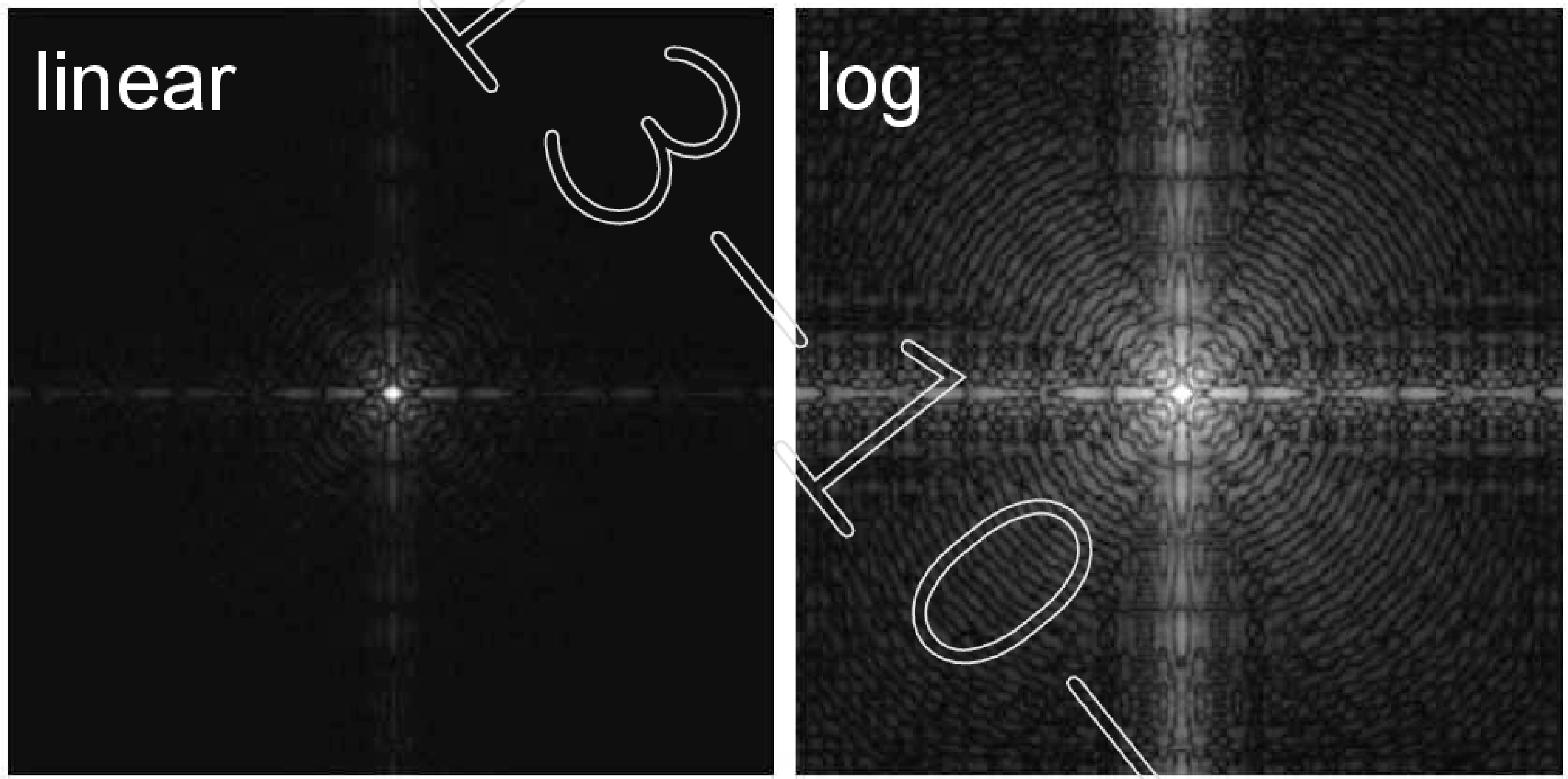
➡ Compresses the dynamic range of “images” with large variations in pixel values.



Note: The **inverse log** widens the bright intensities and squeezes the darker intensities.

Logarithmic Transform

Example: Absolute values of a Fourier-transform range easily between 0 and more than 10^6.



- With a linear scaling to [0,255] details of the spectrum are lost
- With a logarithmic transform the characteristic of the spectrum becomes more visible.

Example – gamma transform (exponential)

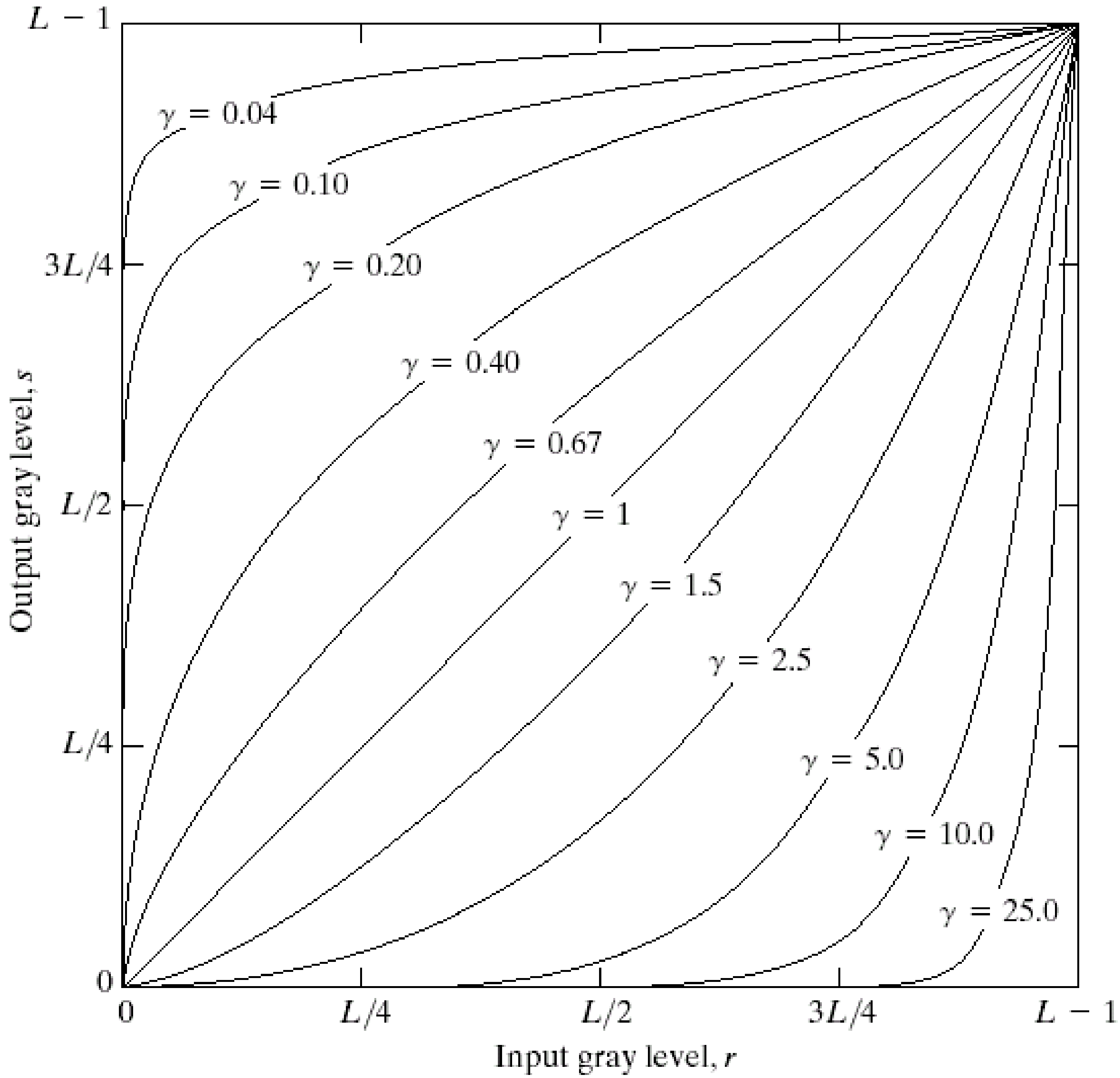


FIGURE 3.6 Plots of the equation $s = cr^{\gamma}$ for various values of γ ($c = 1$ in all cases).

- Each input value is raised to the power of gamma
- Gamma transforms were often used to correct the intensities in CRT displays

$$s = cr^{\gamma}$$

Matlab examples

- Image negative
  - I = imread ('rice.png');
  - In = 255 - I;   % image negative

- Gamma transform
  - I = imread('Fig0308(a).tif');   % image of spine
  - I = double(I)/255;   % scale to 0..1 (could also use im2double)
  - Ig = power(I, 0.3);   % enhances dark or light?

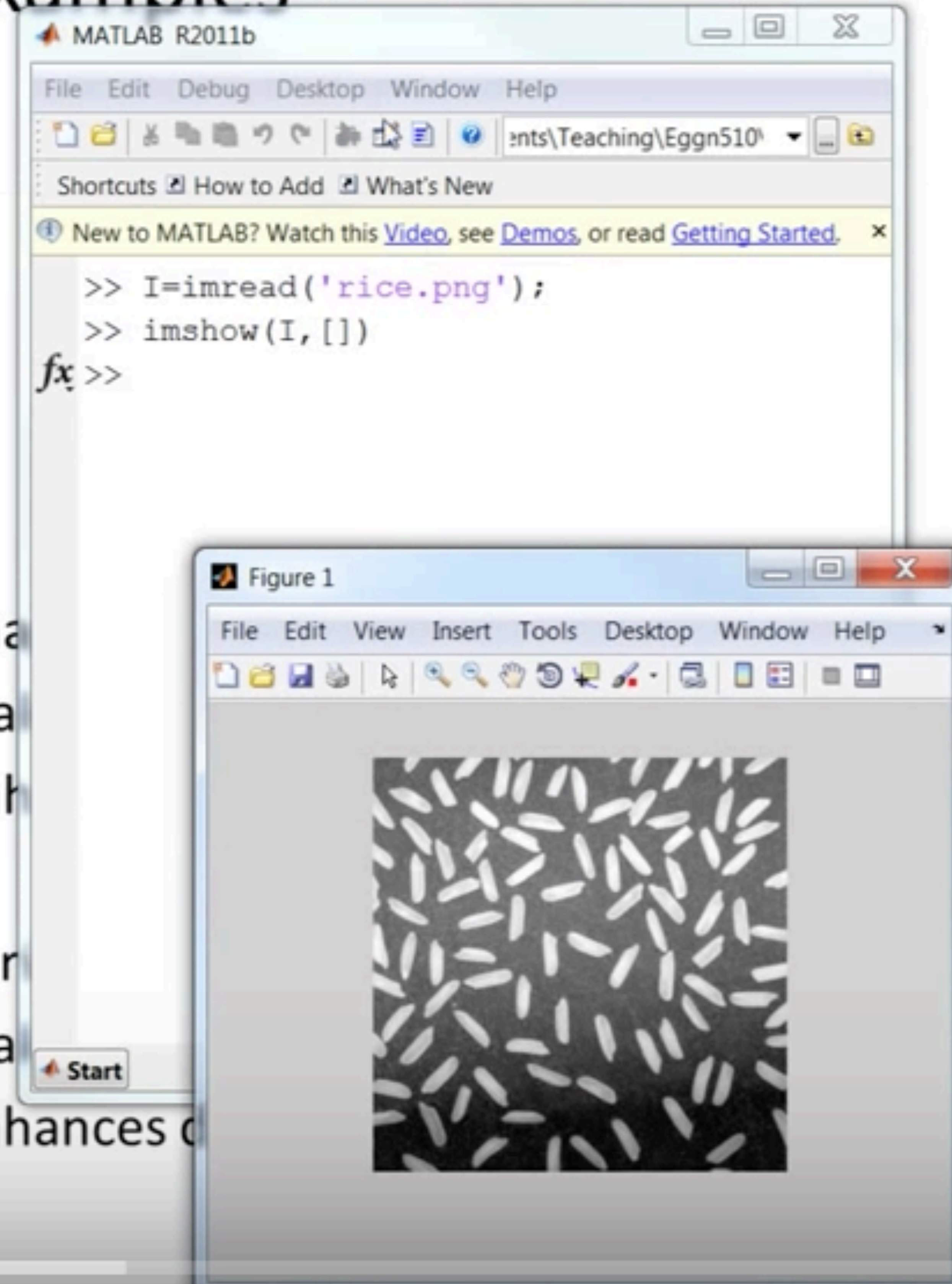
  - I = imread('Fig0309(a).tif');   % aerial image
  - I = double(I)/255;   % scale to 0..1
  - Ig = power(I, 4.0);   % enhances dark or light?

Matlab examples

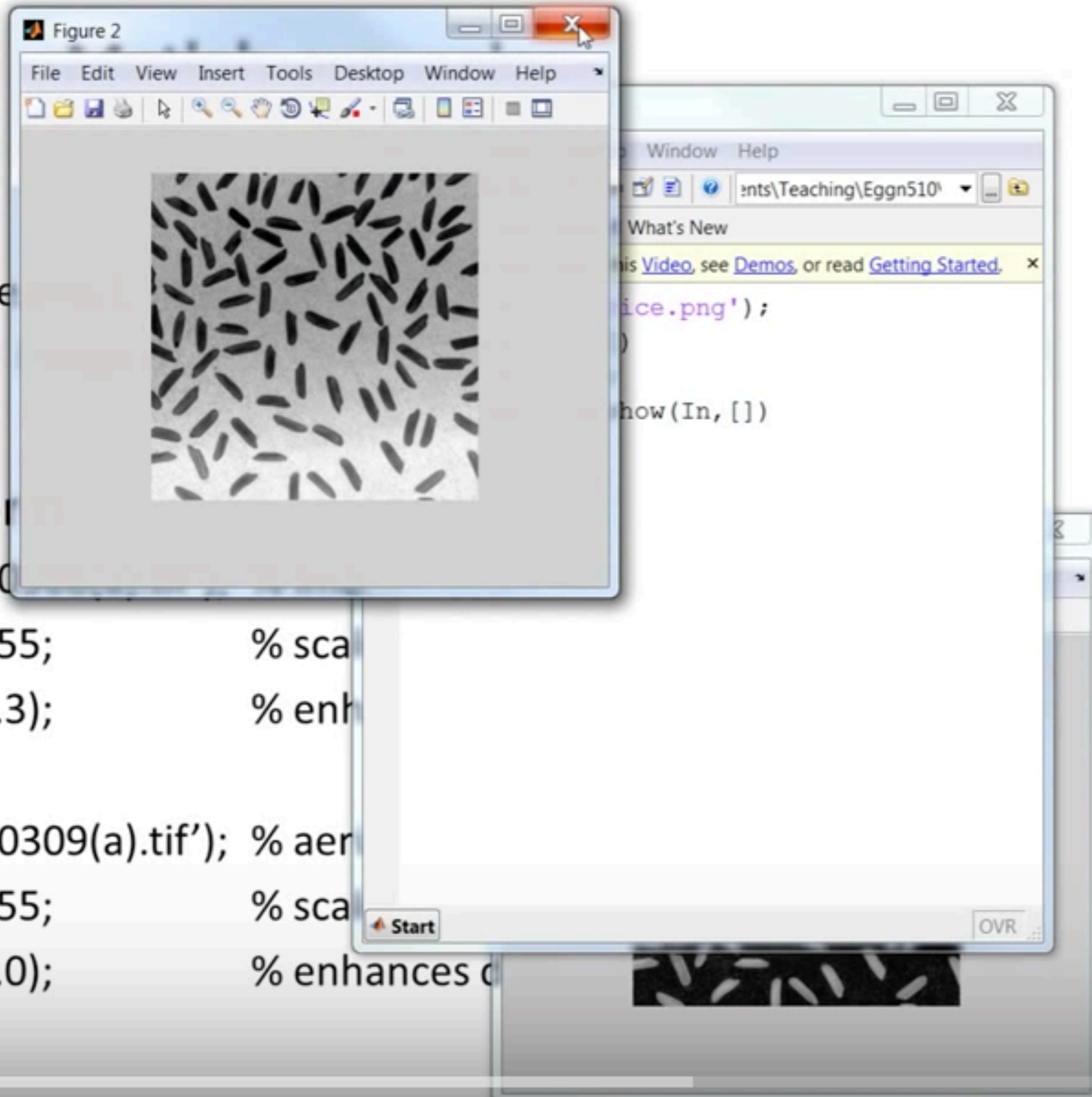
- Image negative
  - I = imread ('rice.png');
  - In = 255 - I;   % image negative

- Gamma transform
  - I = imread('Fig0308(a).tif'); % ima
  - I = double(I)/255; % sca
  - Ig = power(I, 0.3); % enh

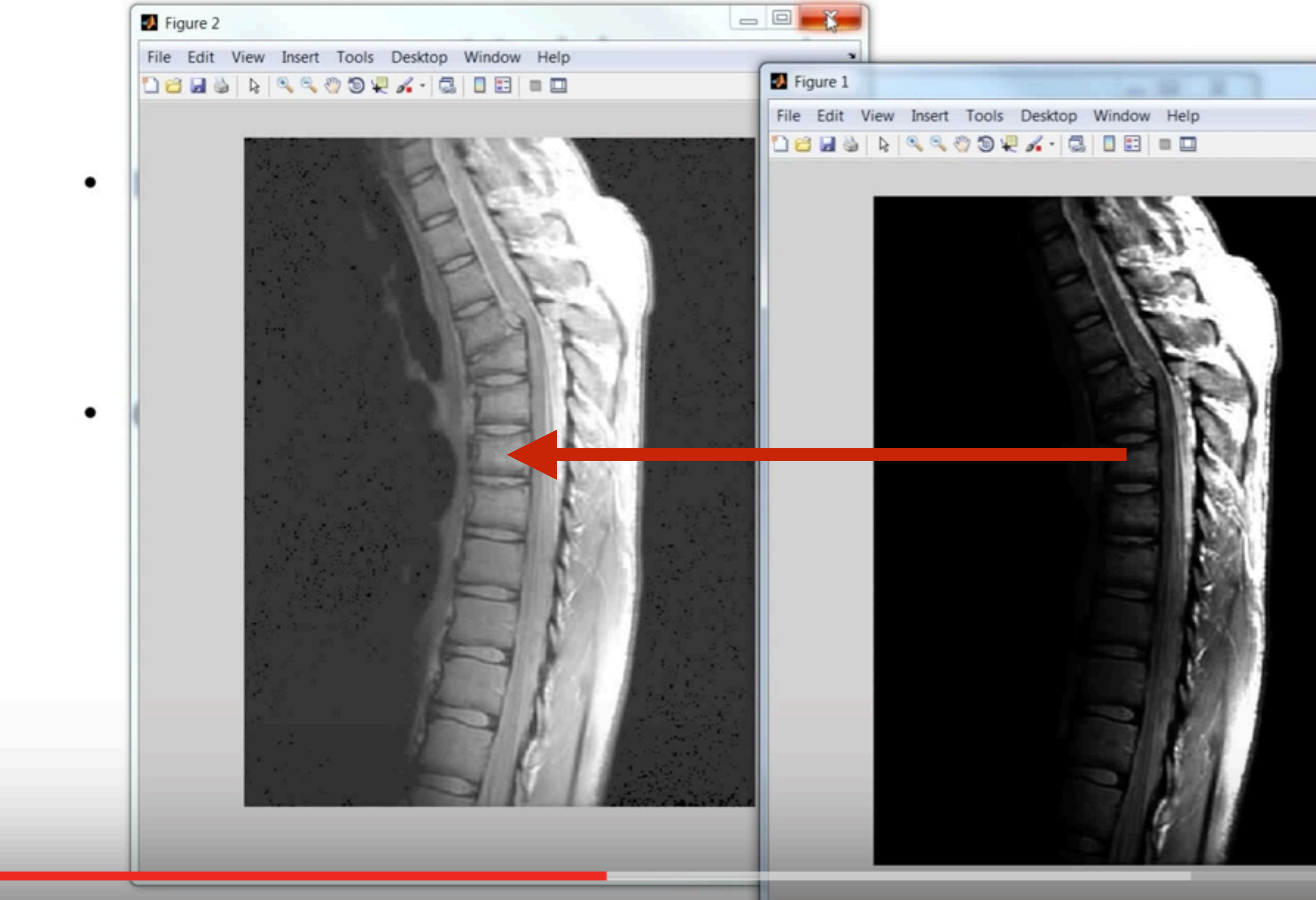
  - I = imread('Fig0309(a).tif'); % aer
  - I = double(I)/255; % sca
  - Ig = power(I, 4.0); % enhances c



- Image negative
  - I = imread('rice
  - In = 255 - I;   %
- Gamma transfor
  - I = imread('Fig0
  - I = double(I)/255;   % sca
  - Ig = power(I, 0.3);   % enh
  - I = imread('Fig0309(a).tif');   % aer
  - I = double(I)/255;   % sca
  - Ig = power(I, 4.0);   % enhances c



$s = cr^{\gamma}, \gamma\,?$



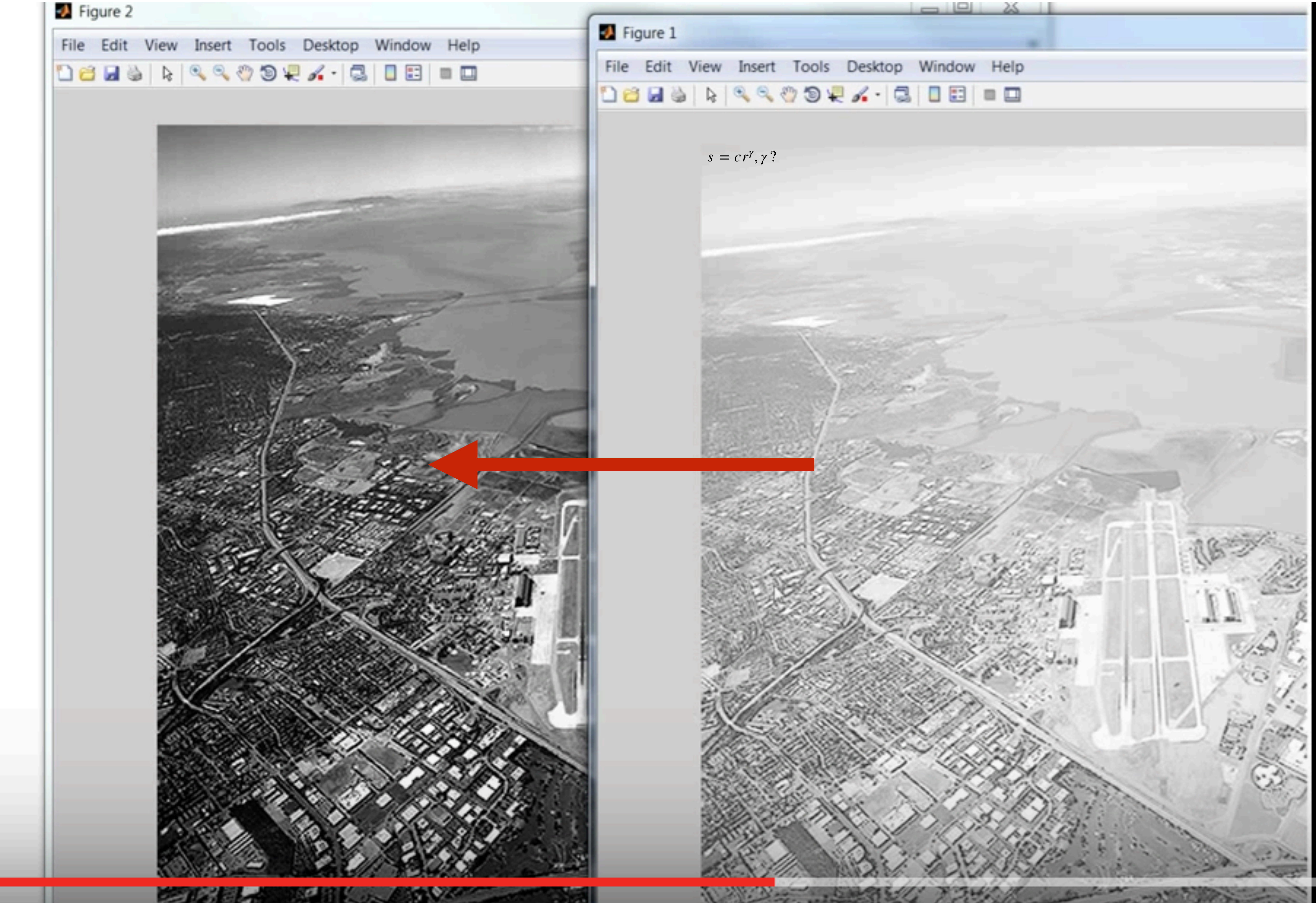


Figure 2
File Edit View Insert Tools Desktop Window Help
Figure 1
File Edit View Insert Tools Desktop Window Help
$s = cr^{\gamma}, \gamma ?$

Quick Review of Probability Concepts

- Probability
  - We do an experiment (e.g., flip a coin) N times
  - We count number of outcomes of a certain type (e.g. heads)
  - Probability of getting that outcome is the relative frequency as N grows large

$$P(heads) = n_H / N$$

  - Probability of a particular outcome is between 0 and 1
  - Probabilities of all outcomes sum to 1

- Random variable
  - Takes on values as a result of performing an experiment (i.e., maps experimental outcomes to real numbers)
  - Example
    - Random variable x represents coin toss
    - E.g., x=0 for heads and x=1 for tails
    - $P(x=0) = 0.5$, $P(x=1) = 0.5$

Probability, random variable (RV),
mean / expected value (RV), variance (RV), continuous (vs. discrete) (RV)

Quick Review of Probability Concepts (cont)

- Mean (or expected value) of a random variable

$$\bar{x} = \mu_x = E[x] = \underline{\frac{1}{N}\sum_{i=1}^{N} x_i}$$

  - where x_i is the value of the ith experiment
- Say x can take on the values $r_0, r_1, \ldots r_{L-1}$
- Then

$$p(x=r_k) = n_k / N$$

  - where n_k = number of times r_k occurs in N trials
- The mean is

$$\bar{x} = \frac{1}{N}\sum_{i=1}^{N} x_i = \frac{1}{N}\left[n_0 r_0 + n_1 r_1 + \ldots + n_{L-1} r_{L-1}\right]$$

$$= \left[\frac{n_0}{N} r_0 + \frac{n_1}{N} r_1 + \ldots + \frac{n_{L-1}}{N} r_{L-1}\right] = \underline{\sum_{k=0}^{L-1} p(r_k) r_k}$$

Quick Review of Probability Concepts (cont)

- Variance of a random variable is

$$\sigma^2 = E\left[\left(x-\mu_x\right)^2\right] = \underline{\frac{1}{N}\sum_{i=1}^{N}\left(x_i-\mu_x\right)^2}$$

- Also

$$\sigma^2 = \underline{\sum_{k=0}^{L-1} p(r)\left(r_k-\mu_x\right)^2}$$

- Note - sometimes it is easier to compute variance using

$$\sigma^2 = \frac{1}{N}\sum_{i=1}^{N}\left(x_i-\mu_x\right)^2 = \frac{1}{N}\sum_{i=1}^{N}\left(x_i^2-2x_i\mu_x+\mu_x^2\right)$$

$$= \frac{1}{N}\sum_{i=1}^{N}\left(x_i^2\right) - \frac{1}{N}\sum_{i=1}^{N}\left(2x_i\mu_x\right) + \frac{1}{N}\sum_{i=1}^{N}\left(\mu_x^2\right) = E[x^2]-\mu_x^2$$

Quick Review of Probability Concepts (cont)

- *Continuous* random variables (as opposed to *discrete*) can take on non-integer values (e.g., temperature)

- We can't talk about the probability of *x* taking a specific value, but we can give the probability of *x* having a value somewhere in a range

- The cumulative probability distribution function (CDF) is

$$F(a) = P(-\infty < x \le a) \qquad\qquad 0 \le F(x) \le 1$$

- The probability density function (pdf) is

$$p(x) = dF(x)/dx$$

- The probability that *x* is between *a* and *b* is

$$P(a < x \le b) = F(b) - F(a) = \int_a^b p(w)\,dw$$

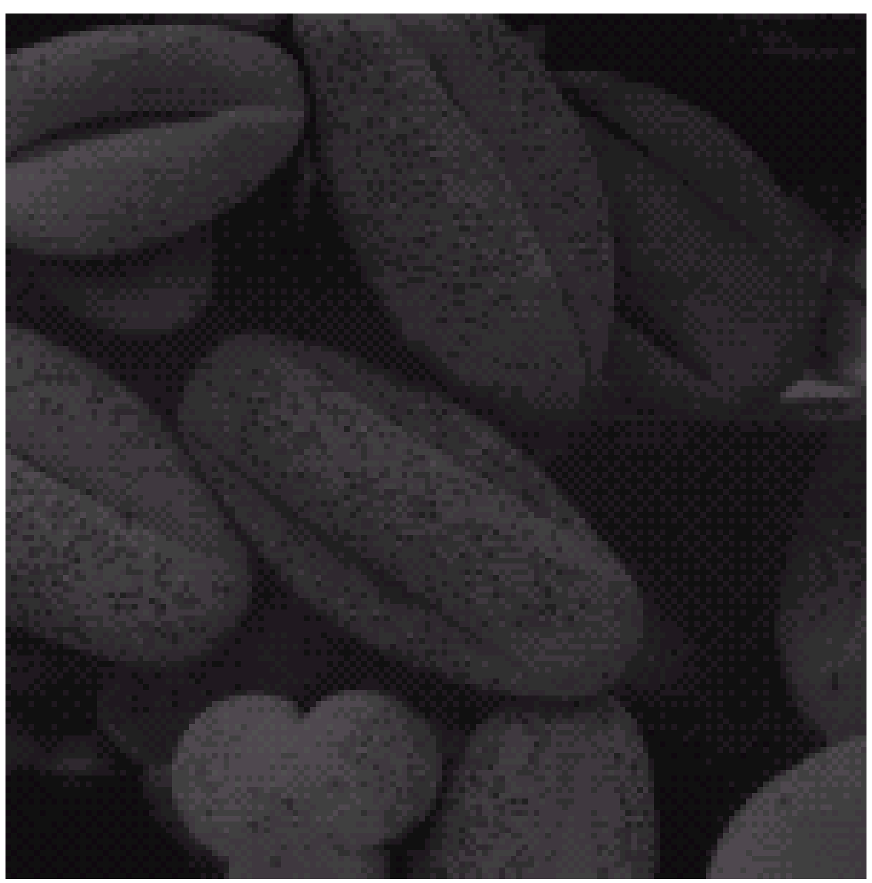
Image Histograms

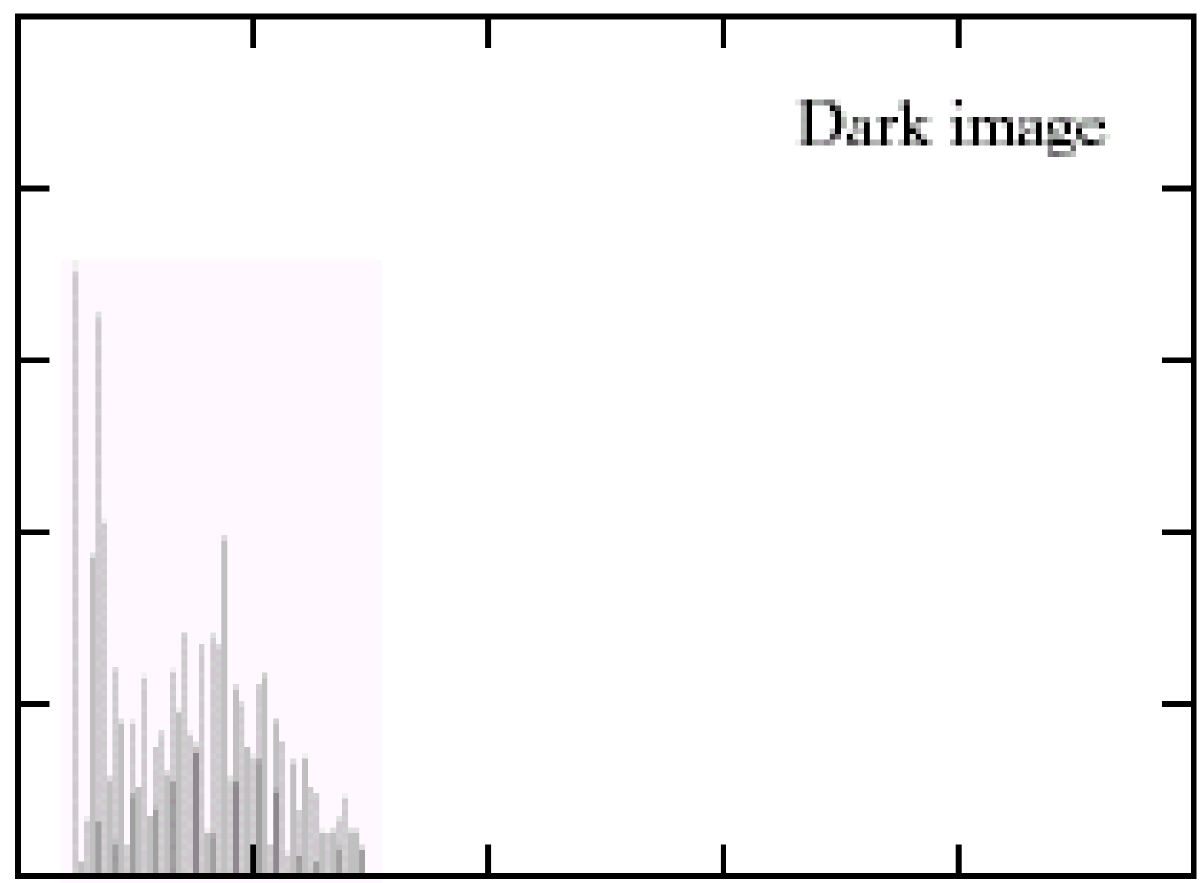
- Histogram is a count of the number of pixels n_k with each gray level r_k

  $n_k = h(r_k)$

- It is an approximation of the probability density function

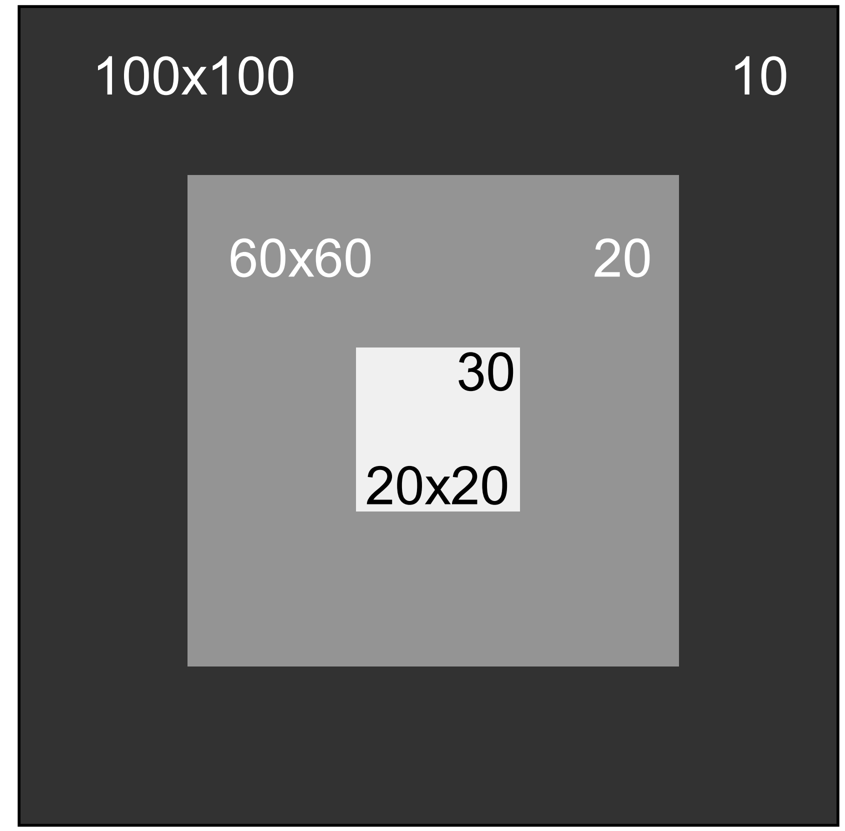
  $p(r_k) = n_k / N$





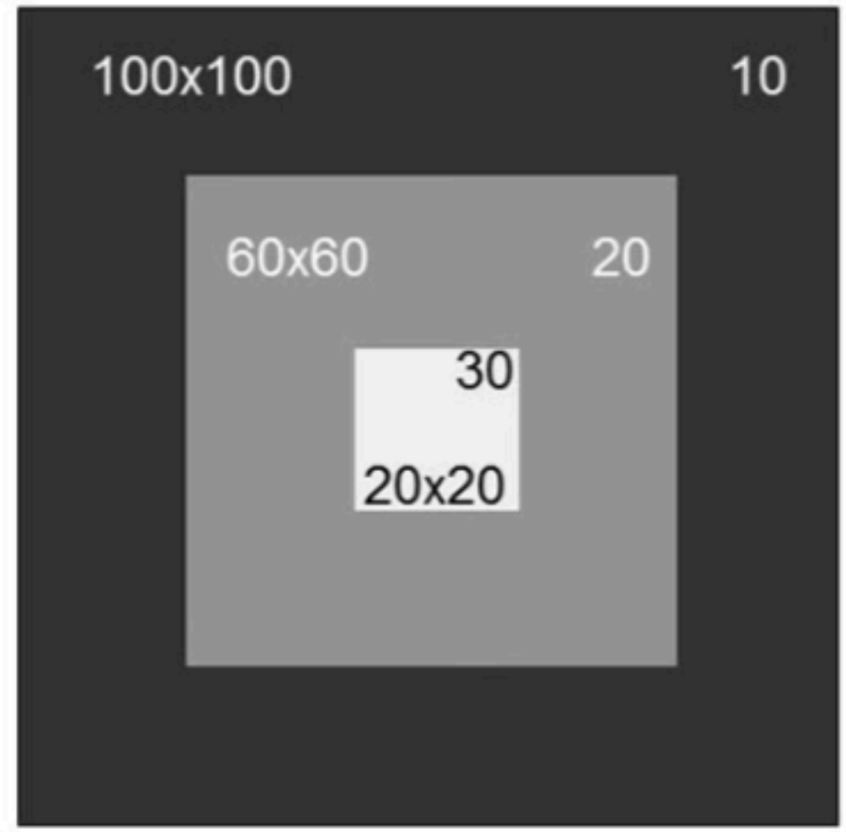
Example

- Find histogram
- Find pdf, CDF
- Find mean
- Find variance



Example

- Find histogram
- Find pdf, CDF
- Find mean
- Find variance

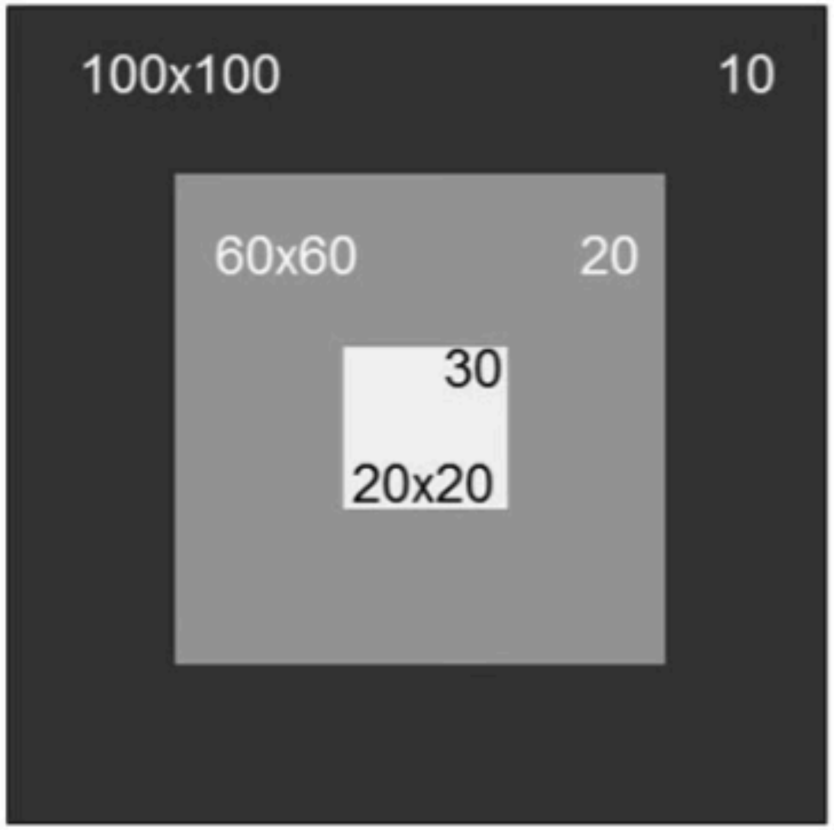


r	H(r)
0	0
⋮	⋮
10	6400
⋮	⋮
20	3200
⋮	⋮
30	400
⋮	⋮
255	0

Sum: 10,000 ✓

Example

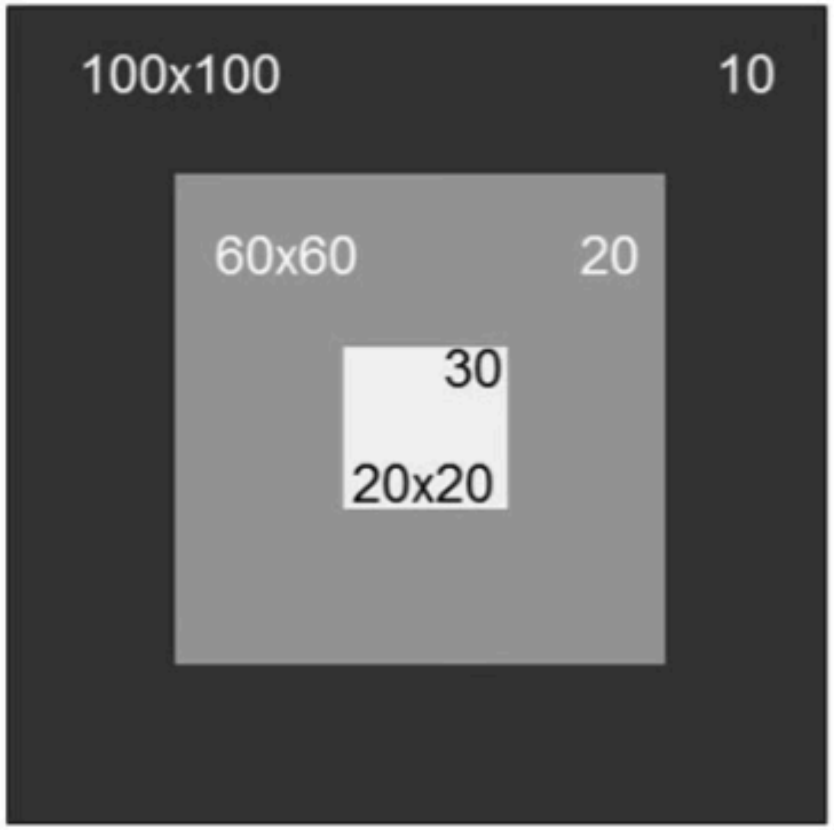
- Find histogram
- Find pdf, CDF
- Find mean
- Find variance



r	H(r)	p(r)
0	0	0
10	6400	0.64
20	3200	0.32
30	400	0.04
255	0	
	Sum: 10,000 ✓	÷N

Example

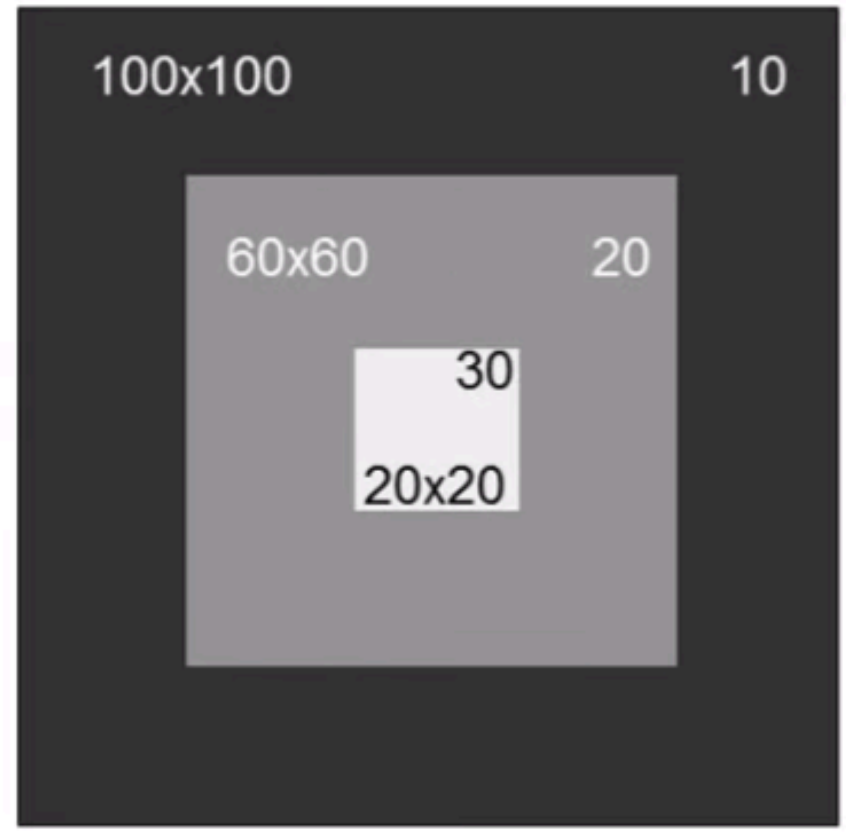
- Find histogram
- Find pdf, CDF
- Find mean
- Find variance



r	$H(r)$	$p(r)$	CDF $F(r)$
0	0	0	0
	⋮		⋮
10	6400	0.64	0.64
	⋮		0.64
20	3200	0.32	0.96
	⋮		0.96
30	400	0.04	1.0
	⋮		1.0
255	0		1.0
	Sum: 10,000 ✓	÷N	

Example

- Find histogram
- Find pdf, CDF
- Find mean
- Find variance



r	$H(r)$	$p(r)$	CDF $F(r)$
0	0	0	0
⋮	⋮		⋮
10	6400	0.64	0.64
			0.64
20	3200	0.32	0.96
			0.96
30	400	0.04	1.0
			1.0
⋮	⋮		⋮
255	0		1.0
SUM: 10,000 ✓		÷N	

$$\mu = \sum_{k=0}^{L-1} p_r(r_k)\, r_k$$

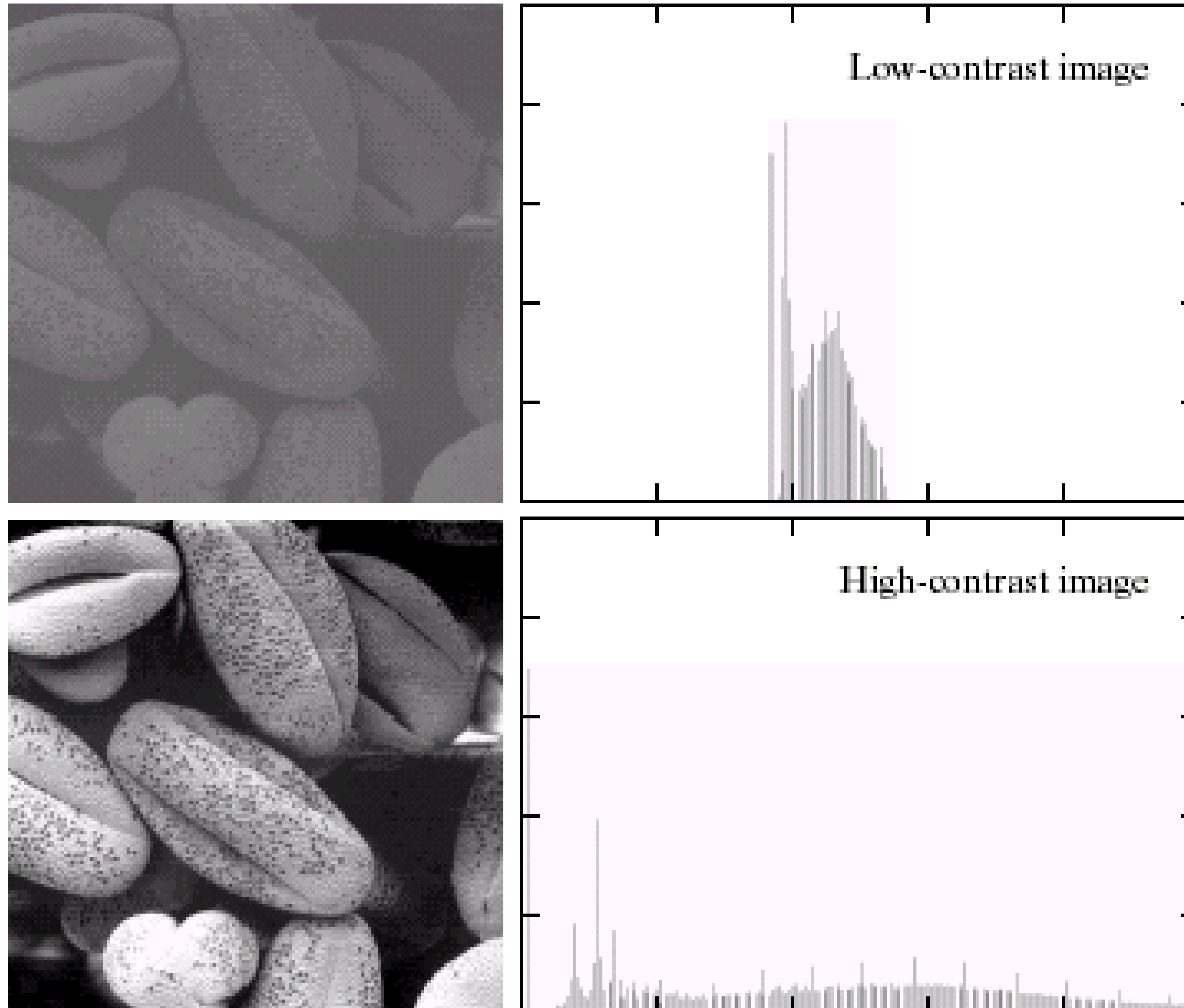
$$= (10)(0.64) + (20)(0.32) + (30)(.04)$$

$$= 14.0$$

$$\sigma^2 = \sum p(r_k)(r_k - \mu)^2 = 32.0$$

Histogram Equalization

- We can transform image values to improve the contrast
- Want histogram of the image to be flat
- This will make full use of the entire display range
- This is called histogram equalization



Histogram Equalization

- Let the histogram of the input image be $H(r)$

- The pdf of the input image is

  $p_r(r) = H(r)/N$

- We want a transformation $s = T(r)$ that will give an output image whose histogram is flat:

  $p_s(s) = \text{const}$

- The transformation should be a monotonically increasing function
  - this prevents artifacts created by reversals of intensity

Histogram Equalization

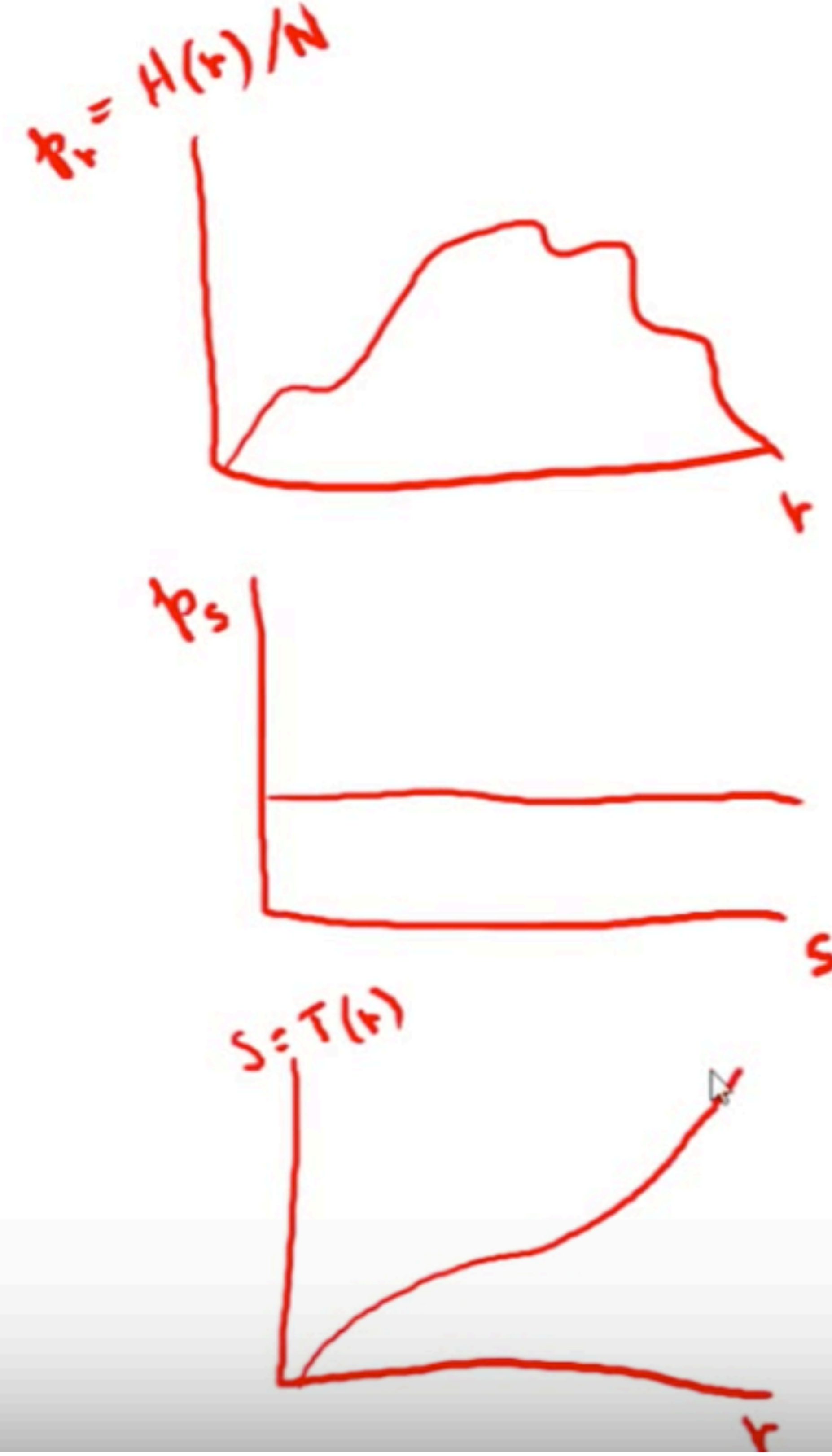
- Let the histogram of the input image be $H(r)$
- The pdf of the input image is

  $p_r(r) = H(r)/N$

- We want a transformation $s = T(r)$ that will give an output image whose histogram is flat:

  $p_s(s) = \text{const}$

- The transformation should be a monotonically increasing function
  - this prevents artifacts created by reversals of intensity



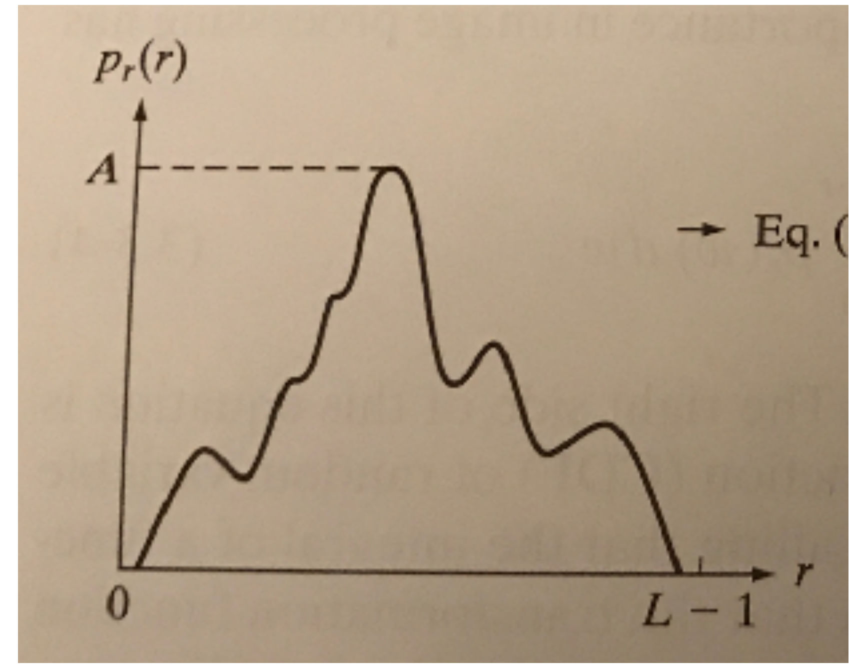
Histogram equalization (contd.)

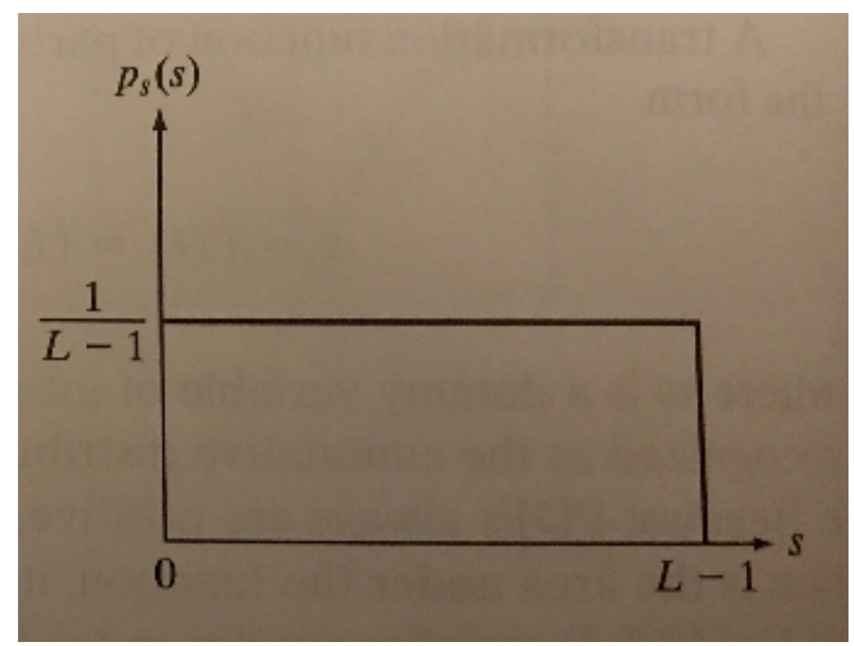
- Consider the cumulative probability distribution function of the input image

$$F(r) = \int_0^r p_r(w)\,dw$$

- If we use this as our transformation function (scaled by the maximum value *L*-1), the output image will have $p_s(s)$ = const

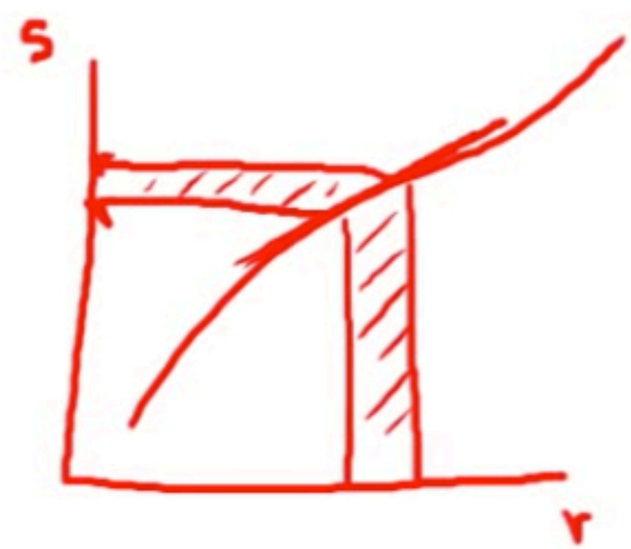
$$s = T(r) = (L-1)\int_0^r p_r(w)\,dw$$





Histogram Equalization

- (Show this from considerations of probability)



$$p_s = p_r \left| \frac{dr}{ds} \right|$$

A fundamental result from probability theory

LET

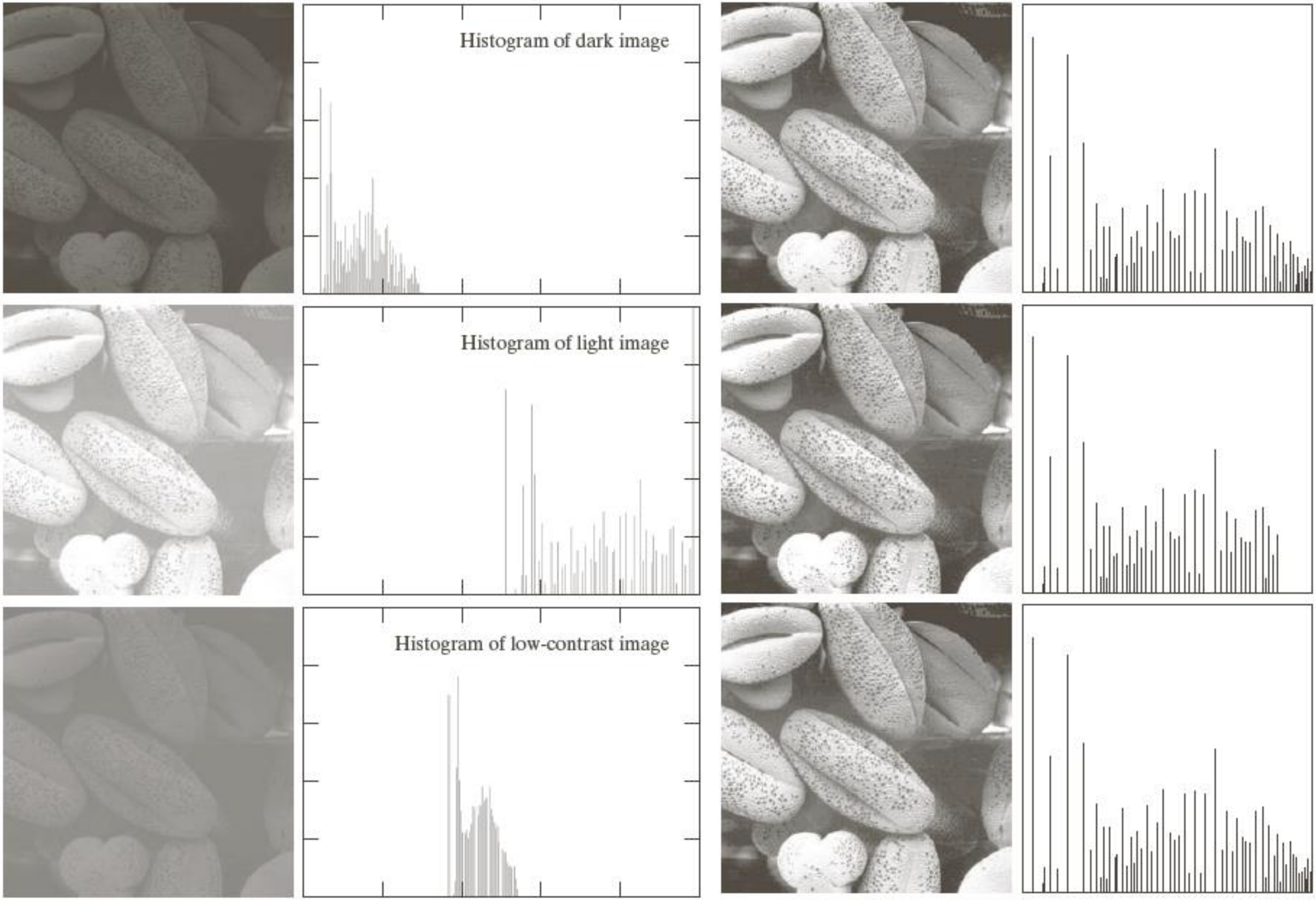
$$s = T(r) = (L-1)\int_0^r p_r(w)\,dw$$

Leibniz's rule in basic calculus: derivative of definite integral wrt upper limit = integrand evaluated at this limit

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

Continues case Approx. in discrete case

SINCE $$p_s = p_r \left| \frac{dr}{ds} \right| = p_r \left| \frac{1}{(L-1)\,p_r(r)} \right| = \frac{1}{L-1}$$


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

Doing histogram equalization by hand

- Get histogram of *MxN* input image $H_r(r) = n_r$. Gray levels range from 0..*L*-1.
- Determine probability density function (pdf)

$$p_r(r_k) = \frac{n_k}{MN}$$

- Determine cumulative probability distribution (CDF)

$$F_r(r_k) = \sum_{j=0}^{k} p_r(r_j)$$

- Scale *T(r)* to desired range of output gray levels

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

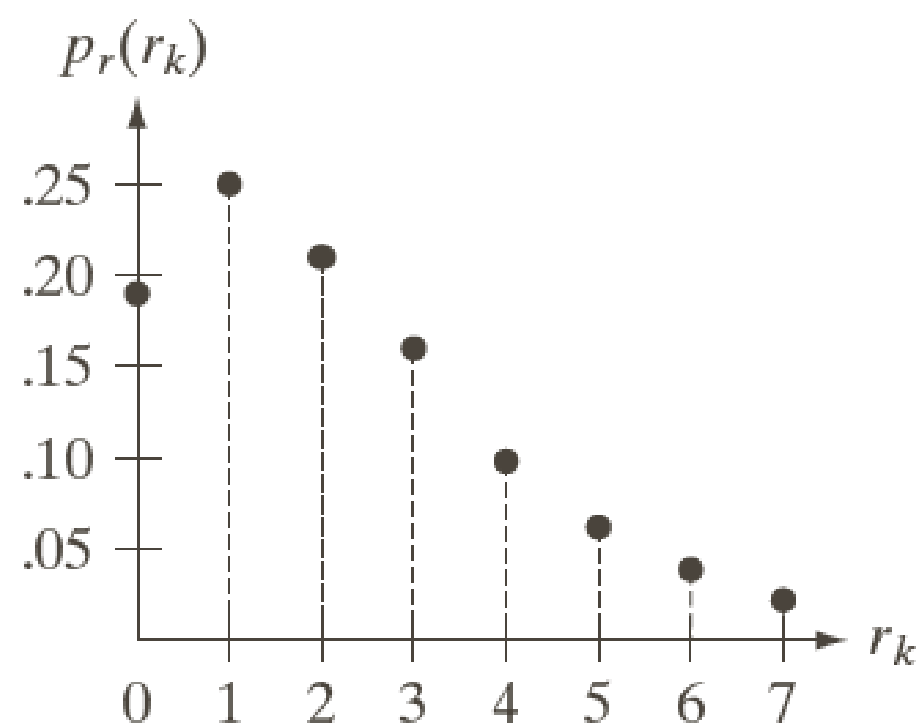
- round / floor
- Apply the transformation *s = T(r)* to compute the output values

Example

- 64x64 image
  - MxN = 64x64 = 4096
- 3 bits/pixel
  - Gray levels range from 0 to L-1
  - L = 2^3 = 8

Example 3.5 in book

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



Example (continued)

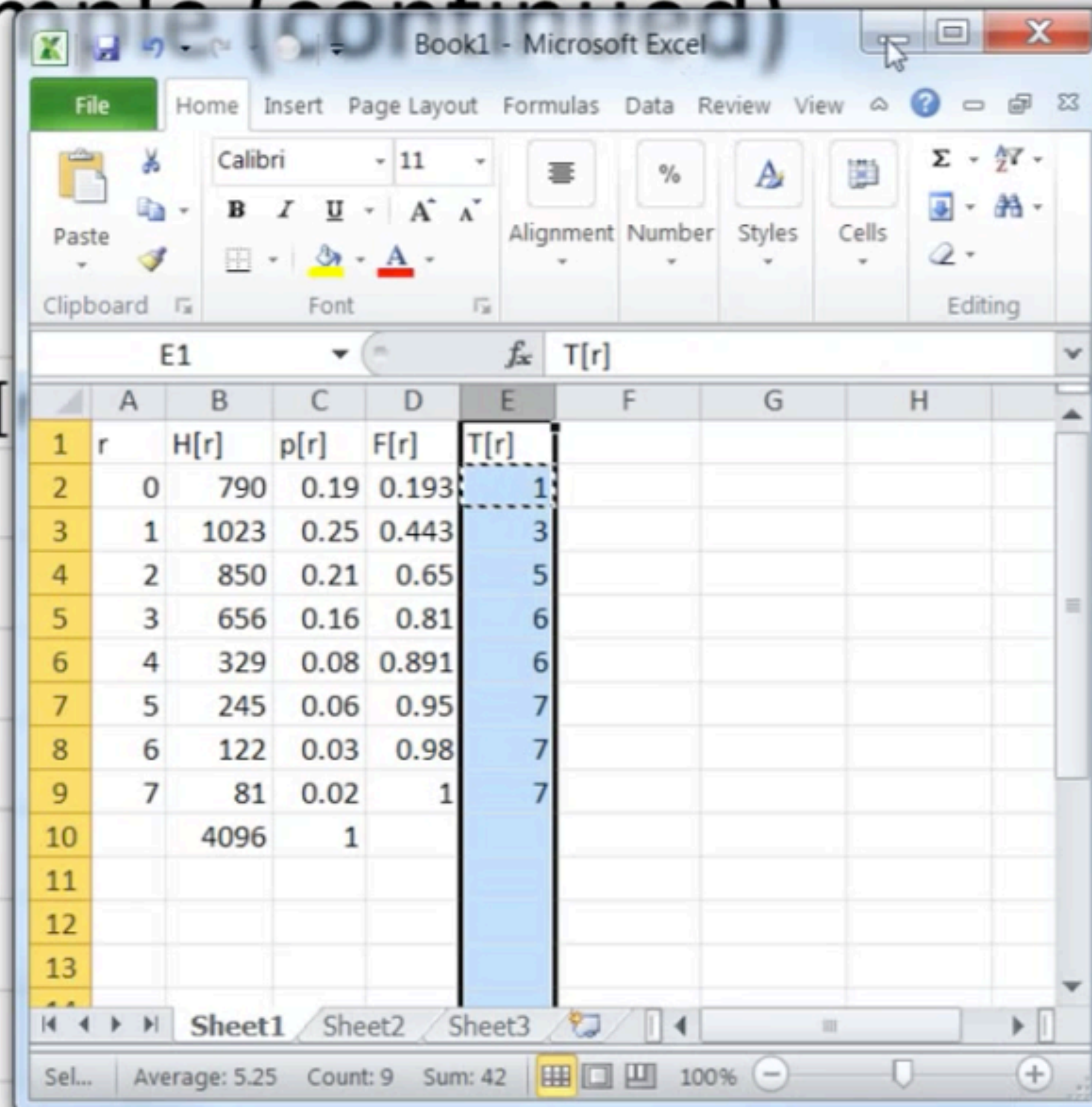
- Excel spreadsheet

r	H[r]	p[r]
0	790	
1	1023	
2	850	
3	656	
4	329	
5	245	
6	122	
7	81	
	4096	

Example (continued)

- Excel spreadsheet

r	H[r]	p[
0	790	
1	1023	
2	850	
3	656	
4	329	
5	245	
6	122	
7	81	
	4096	



r	H[r]	p[r]	F[r]	T[r]
0	790	0.19	0.193	1
1	1023	0.25	0.443	3
2	850	0.21	0.65	5
3	656	0.16	0.81	6
4	329	0.08	0.891	6
5	245	0.06	0.95	7
6	122	0.03	0.98	7
7	81	0.02	1	7
	4096	1		

Example (continued)

- To calculate histogram of transformed image $H_s(s)$:
  - For each value of s
    - Find values of r where $s = T(r)$
    - Sum $H_r(r)$ for those values

- Excel spreadsheet

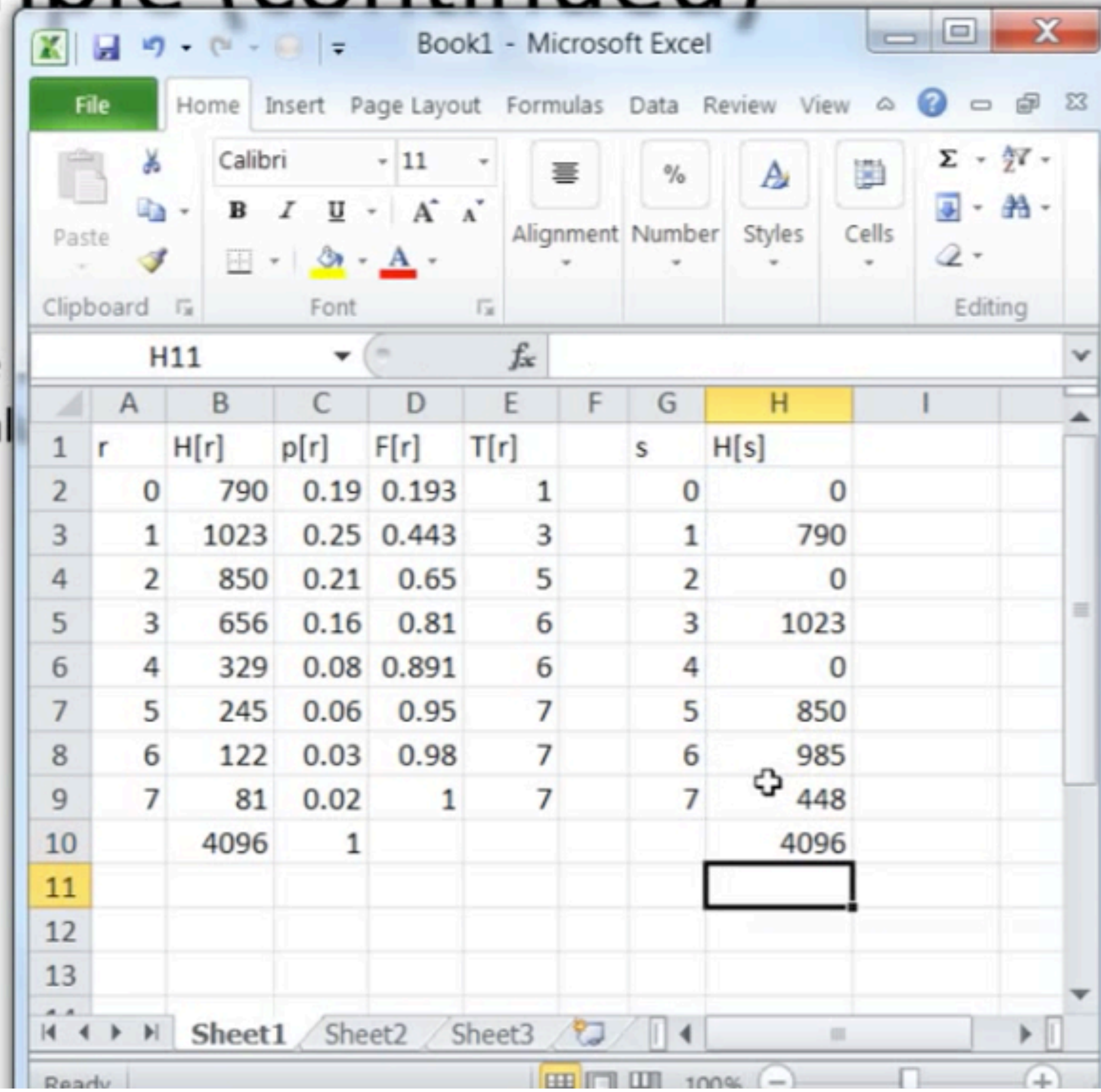
- Example:
  - Take s=6
  - $T(r)$ = 6 for r=3,4
  - $H_s(6) = H_r(3) + H_r(4)$
    $= 656 + 329 = 985$

NB: both input 3 and 4 maps to output 6,
i.e. # output 6 = # input 3 + 4

Example (continued)

- To calculate histogram of transformed image $H_s(s)$:
  - For each value of s
    - Find values of r where
    - Sum $H_r(r)$ for those val

- Example:
  - Take s=6
  - $T(r)$ = 6 for r=3,4
  - $H_s(6) = H_r(3) + H_r(4)$
    $= 656 + 329 = 985$



	A	B	C	D	E	F	G	H
1	r	H[r]	p[r]	F[r]	T[r]		s	H[s]
2	0	790	0.19	0.193	1		0	0
3	1	1023	0.25	0.443	3		1	790
4	2	850	0.21	0.65	5		2	0
5	3	656	0.16	0.81	6		3	1023
6	4	329	0.08	0.891	6		4	0
7	5	245	0.06	0.95	7		5	850
8	6	122	0.03	0.98	7		6	985
9	7	81	0.02	1	7		7	448
10		4096	1					4096

Manual Histogram Equalization - Example

r	H(r)
0	0
1	0
2	0
3	10
4	20
5	40
6	20
7	10

Manual Histogram Equalization - Example

r	H(r)	$p_r(r)$
0	0	0
1	0	0
2	0	0
3	10	.1
4	20	.2
5	40	.4
6	20	.2
7	10	.1

$\Sigma = 100$

Manual Histogram Equalization - Example

r	H(r)	$p_r(r)$	$F(r)$
0	0	0	0
1	0	0	0
2	0	0	0
3	10	.1	.1
4	20	.2	.3
5	40	.4	.7
6	20	.2	.9
7	10	.1	1.0

$\Sigma = 100$

Manual Histogram Equalization - Example

r	H(r)	$p_r(r)$	$F(r)$	$s = T(r)$
0	0	0	0	0
1	0	0	0	0
2	0	0	0	0
3	10	.1	.1	1
4	20	.2	.3	2
5	40	.4	.7	5
6	20	.2	.9	6
7	10	.1	1.0	7

$\Sigma = 100$

Manual Histogram Equalization - Example

r	H(r)	$p_r(r)$	F(r)	s=T(r)
0	0	0	0	0
1	0	0	0	0
2	0	0	0	0
3	10	.1	.1	1
4	20	.2	.3	2
5	40	.4	.7	5
6	20	.2	.9	6
7	10	.1	1.0	7

$\Sigma = 100$

s	H(s)
0	0
1	10
2	20
3	0
4	0
5	40
6	20
7	10

Adaptive (Local) Histogram Equalization

- Divide image into rectangular subregions (or “tiles”), do histogram equalization on each

- To avoid “blocky” appearance:
  - Make tiles overlapping
  - Or, interpolate across tiles

- Matlab’s `adapthisteq`
  - Optional parameters
    - ‘NumTiles’ *default is [8 8]*
    - ‘ClipLimit’ (0..1; limits # pixels in a bin; higher numbers => more contrast) *default is 0.01*

- Try “liftingbody.png”

Matlab example

- Read image

```
I=imread('liftingbody.png');
```

- Do regular histogram equalization and adapative histogram equalization

```
Ieq = histeq(I);
Iadapteq = adapthisteq(I);
```

- Display results
  - "subplot" allows you to put multiple images in a single figure

```
subplot(1,3,1), imshow(I,[]);   % row, cols, index
subplot(1,3,2), imshow(Ieq,[]);   % row, cols, index
subplot(1,3,3), imshow(Iadapteq,[]);   % row, cols, index
```

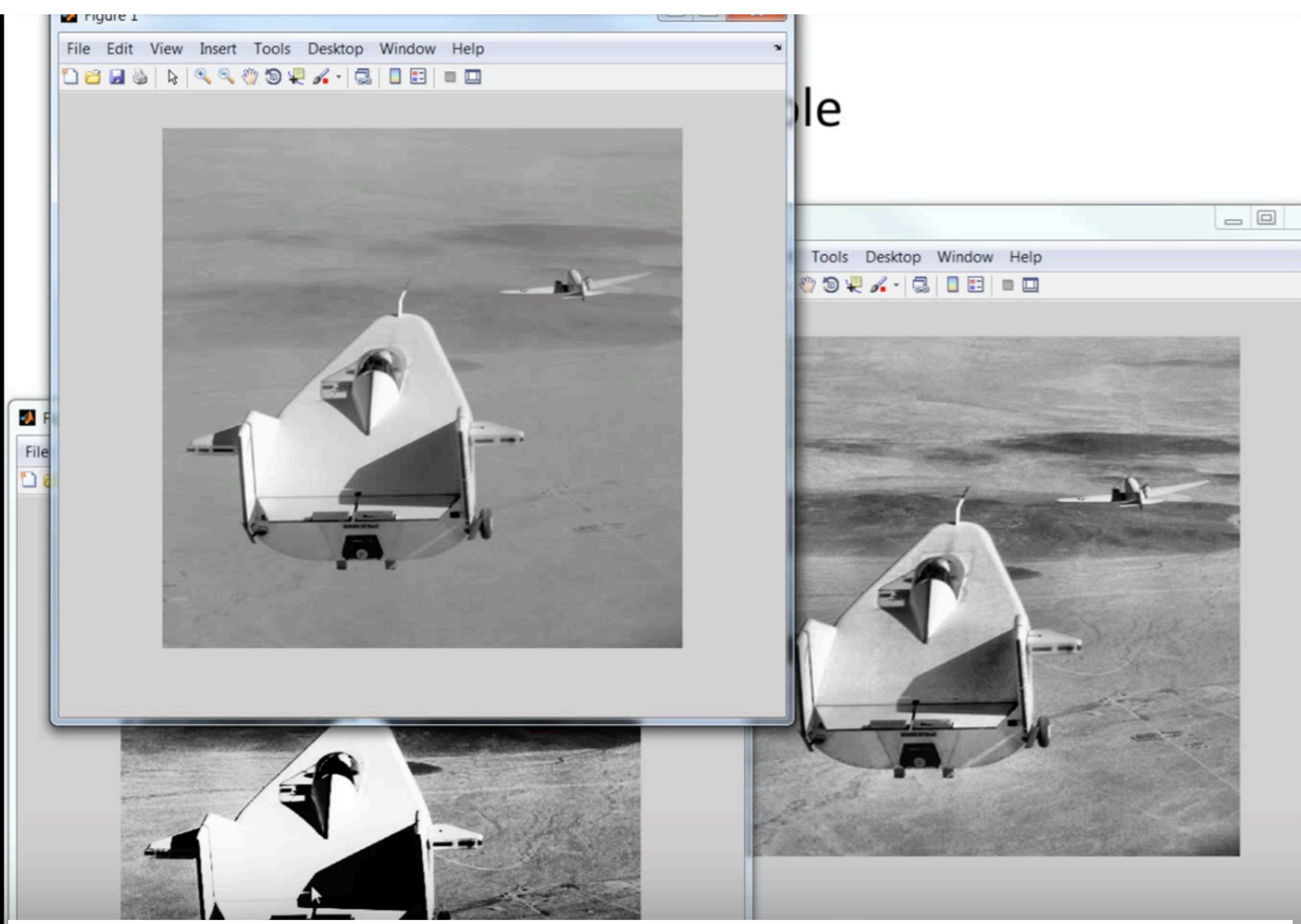


Figure 1
File Edit View Insert Tools Desktop Window Help
le
Tools Desktop Window Help
File

Summary / Questions

- Gray level transformations map each input intensity value to an output intensity value.
- We can use these transformations to improve the contrast in an image.
- Why is it desirable to have a flat histogram in the output image?