Intensity Transformations

Image Enhancement

The techniques in this chapter are mostly used for image enhancement.

Purpose: To make the image more suitable for the target application. The choice of method is very application-specific.

- For human viewing: very subjective
- For automatic processing (e.g., for recognition tasks): The method can be selected based on its effect on the performance of the overall system, but this still involves lots of trial-and-error.

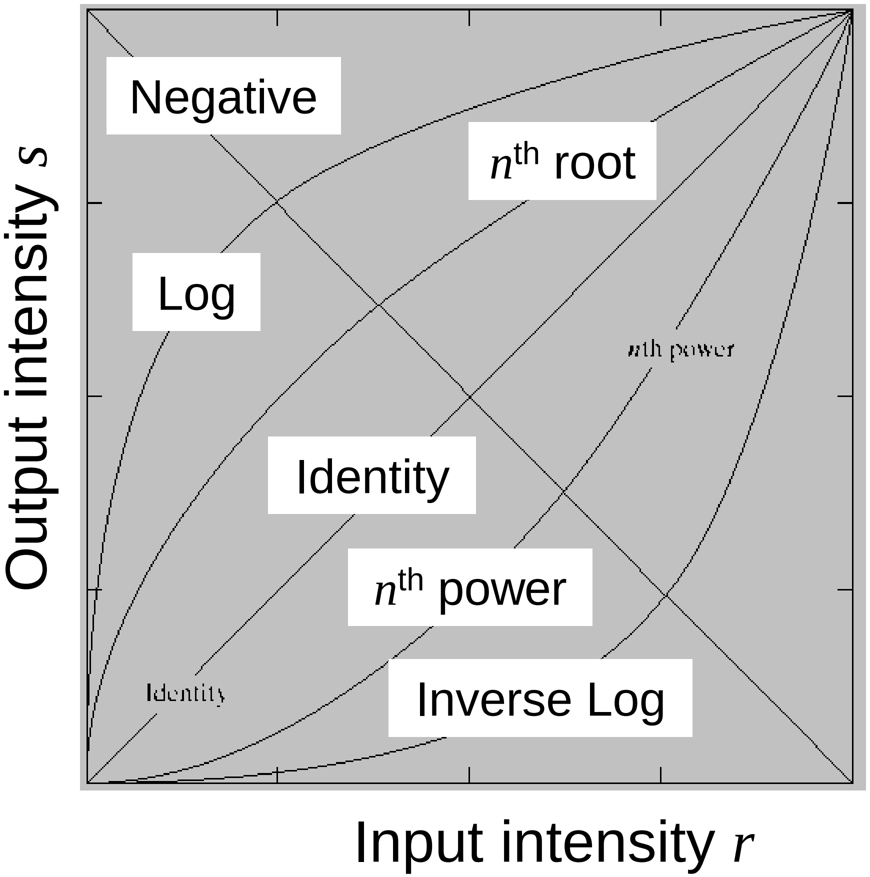
The general approaches:

- Spatial domain: Work on the pixel values directly
- Frequency domain: Work on the Fourier transform of the images

Intensity Transformations

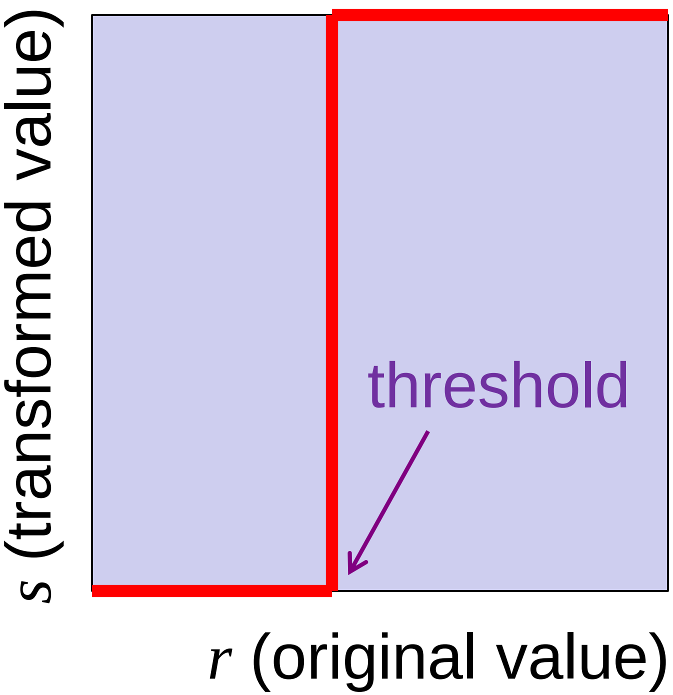
- Transform function ***s=T(r)***, where ***s*** and ***r*** are the original and transformed intensity values, respectively.
- Per-pixel operation; values of neighboring pixels are not considered.

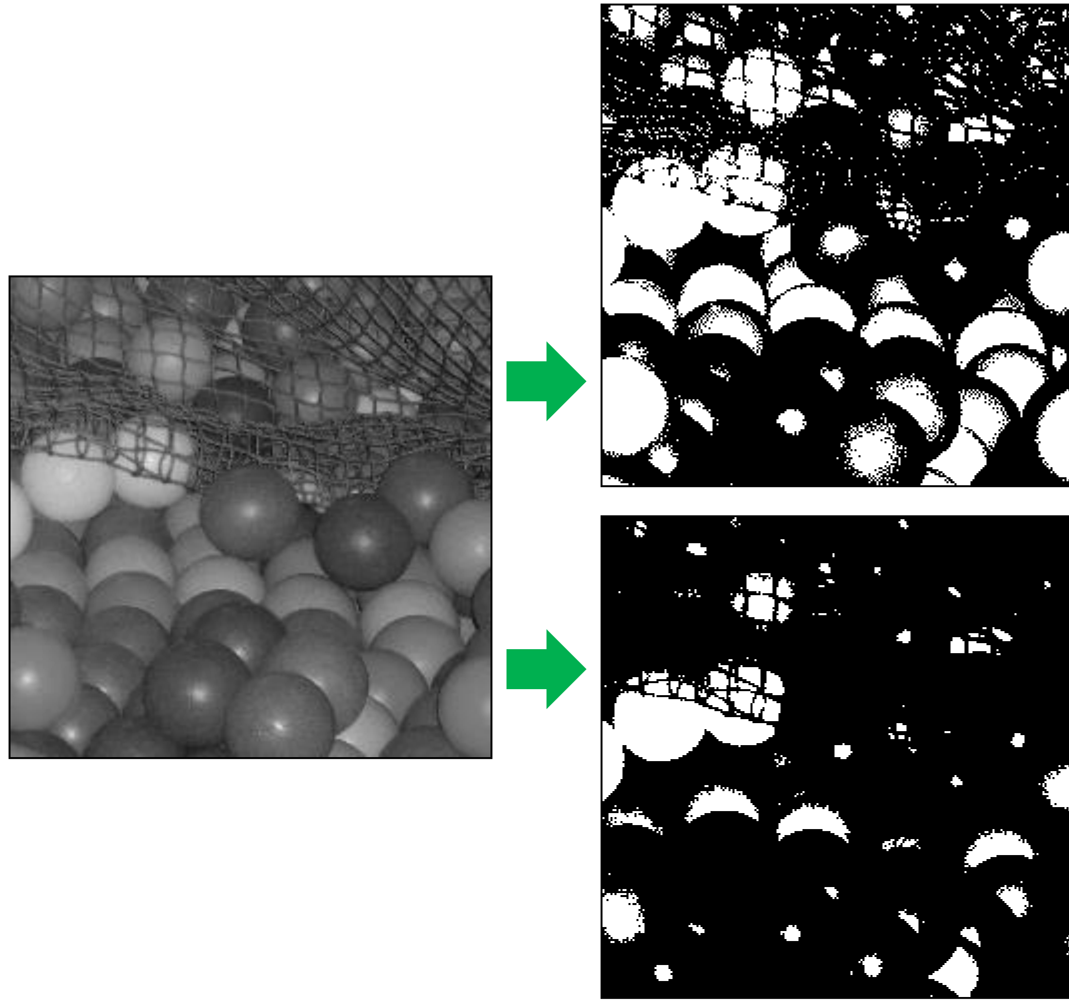
Some example intensity transformations:



Intensity Transformations

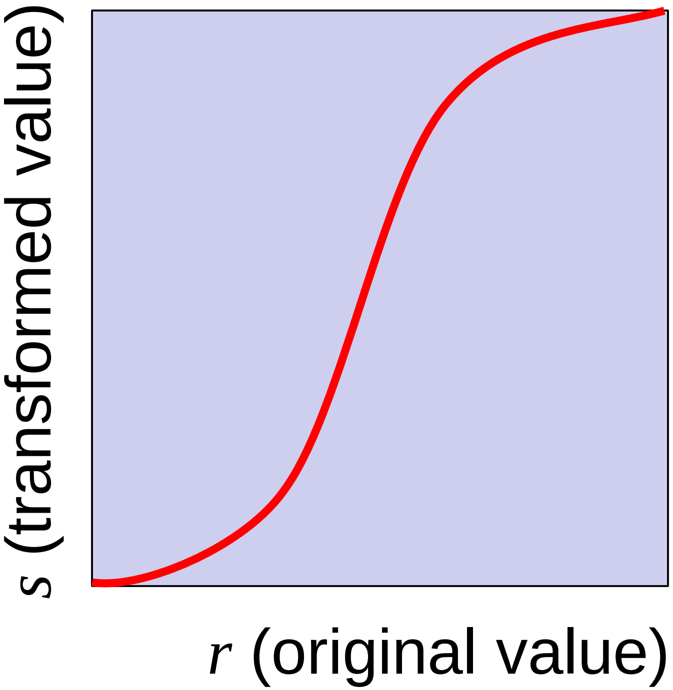
Example: **Thresholding**

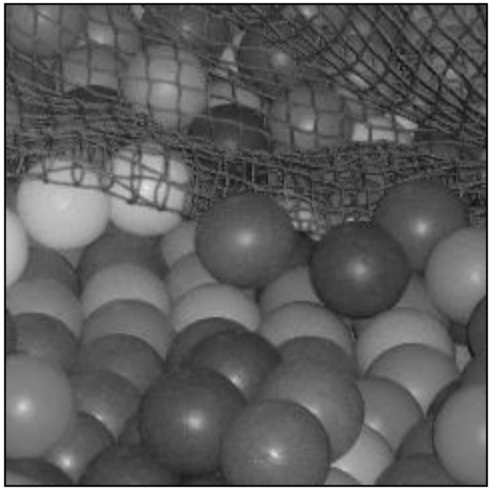


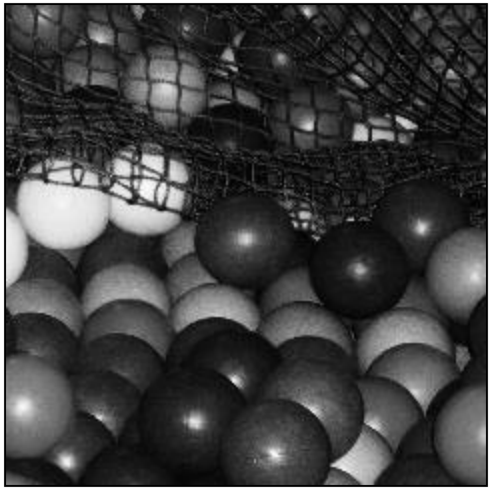


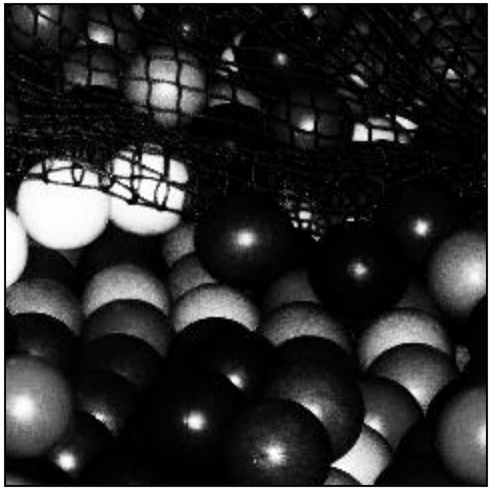
Intensity Transformations

Example: Contrast Stretching



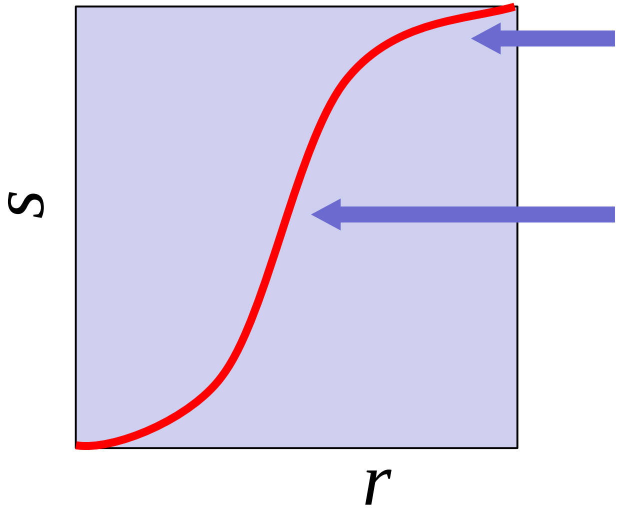






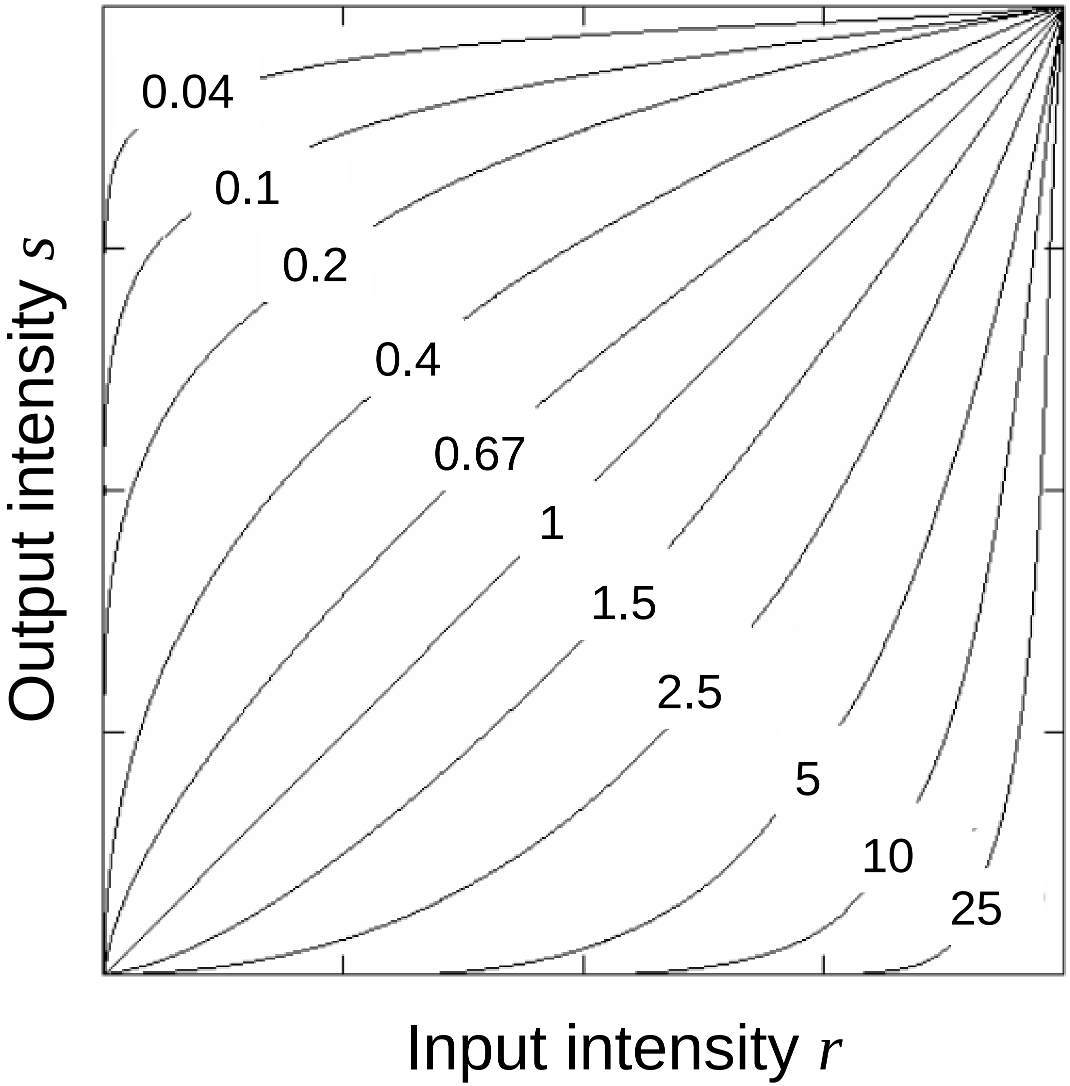
Intensity Transformations

- A major usage of intensity transformations is for contrast enhancement.
- The key to understand the behavior of a transform function is the slope (ds/dr).
  - Contrasts at r with $ds/dr > 1$ are enhanced.
  - Contrasts at r with $ds/dr < 1$ are reduced.



Power-Law Transformations

Form: $s = c\, r^{\gamma}$

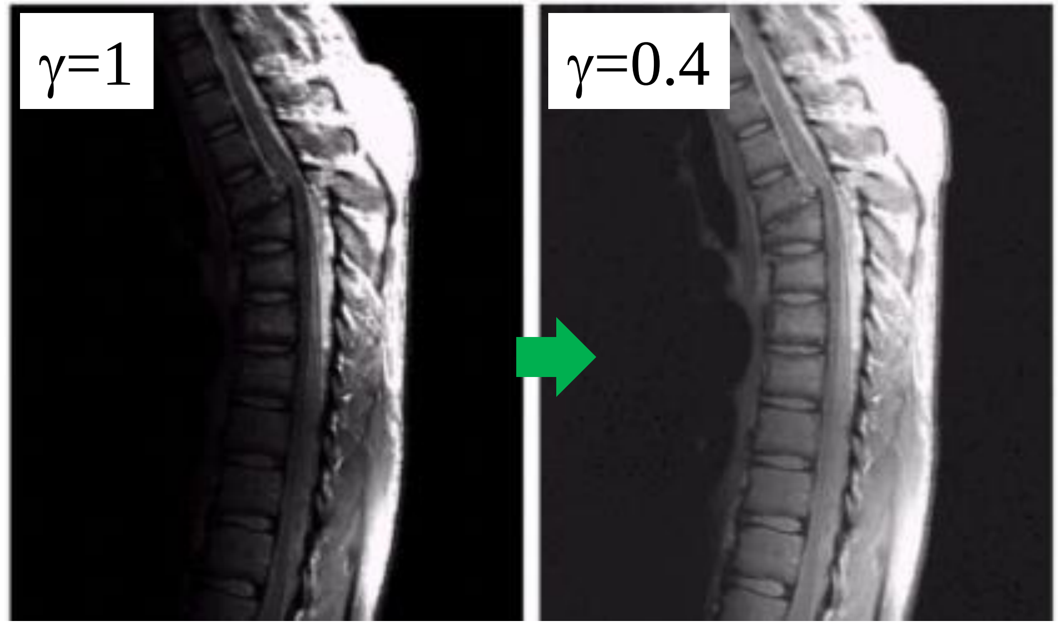


This is commonly called **gamma correction** and normally c = 1 (with both s and r scaled to between 0 and 1.)

Power-Law Transformations

Examples:

Under-exposure:

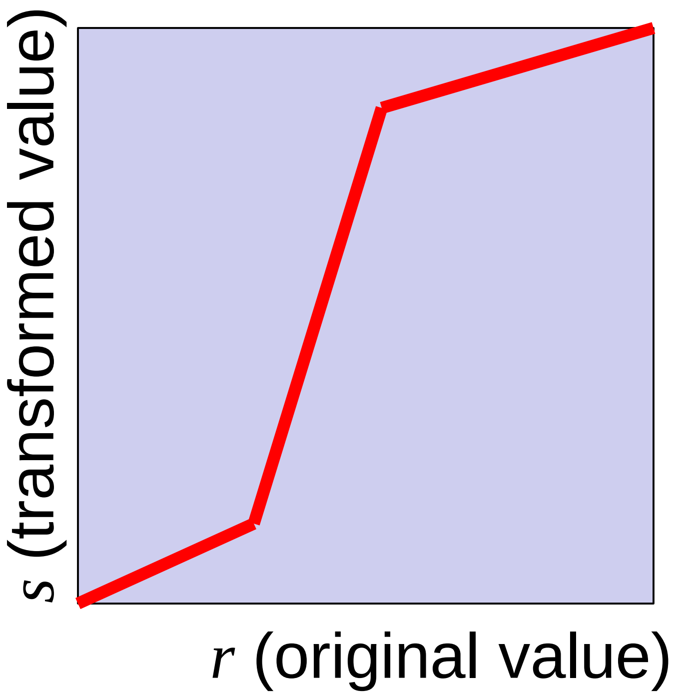


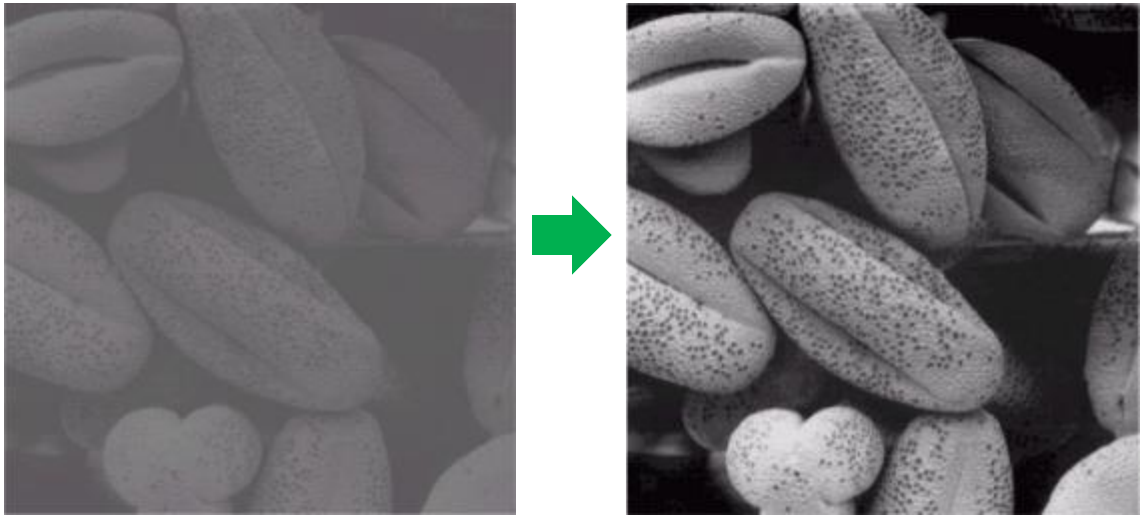
Over-exposure:

Piecewise-Linear Transformations

Example: Piecewise-linear contrast stretching

This allows good flexibility on how to adjust contrasts.





Histogram

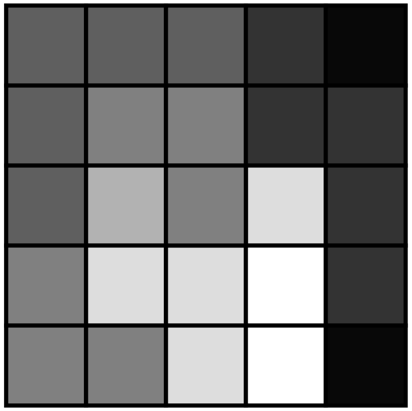
Definition of a histogram: $h(r_k) = n_k$

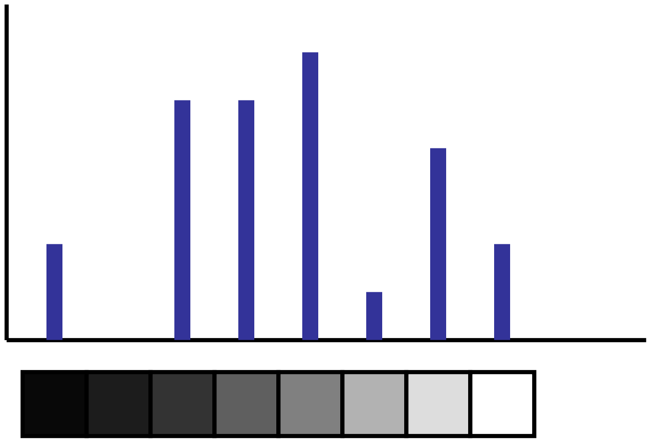
Histogram of a gray-level image:

r_k: the k^{th} gray level

n_k: the number of pixels with gray-level value r_k

Normalized histogram: $p(r_k) = n_k / n$



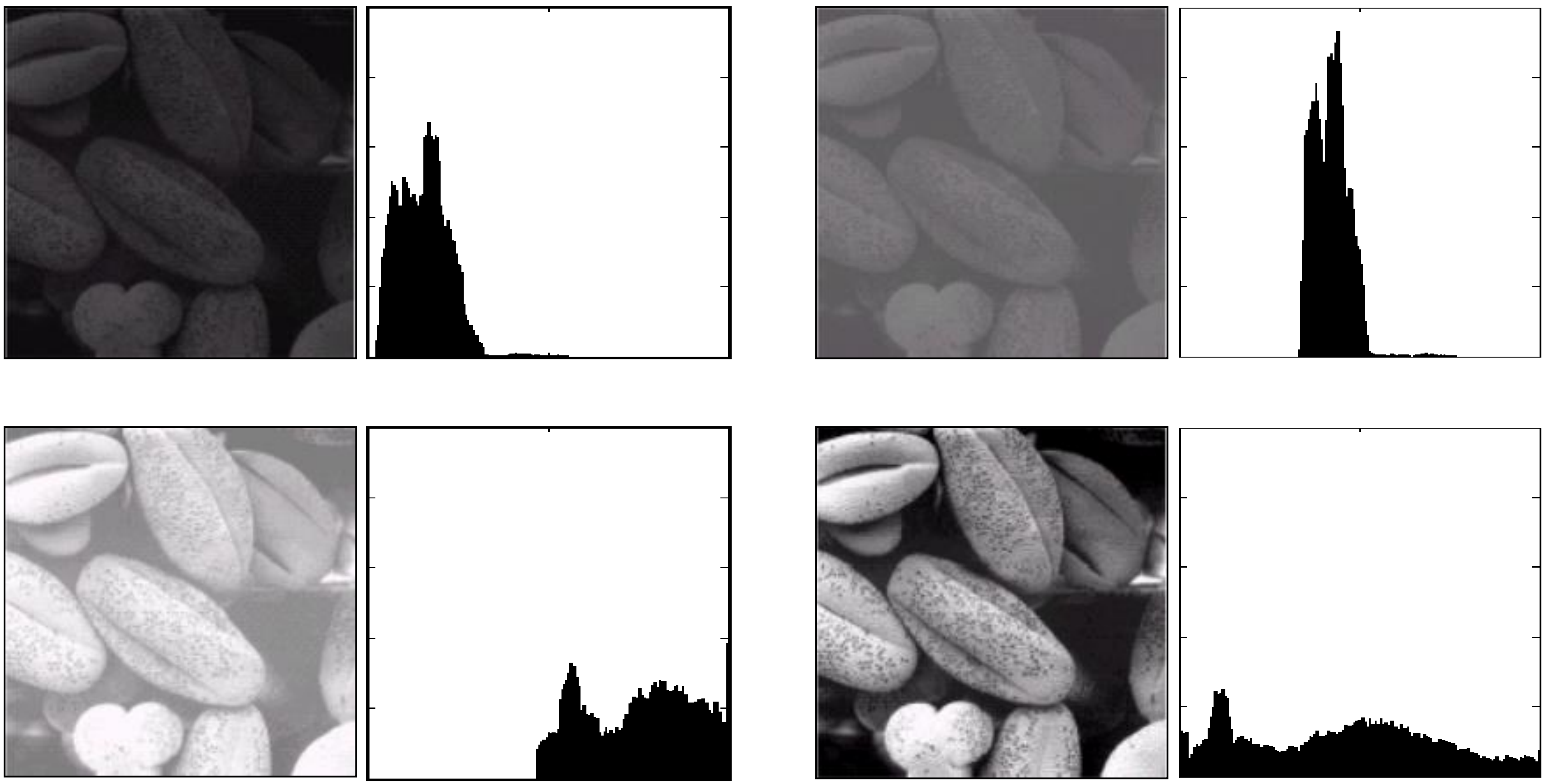


Histogram

- Applications of image histograms:
  - Image enhancement (discussed here)
  - Image segmentation
  - Image compression
  - Image retrieval
  - etc.
- Advantages of using histograms in processing:
  - Easy and efficient to compute
  - Simple hardware implementation
  - Invariant under translation, scaling, rotation, etc.

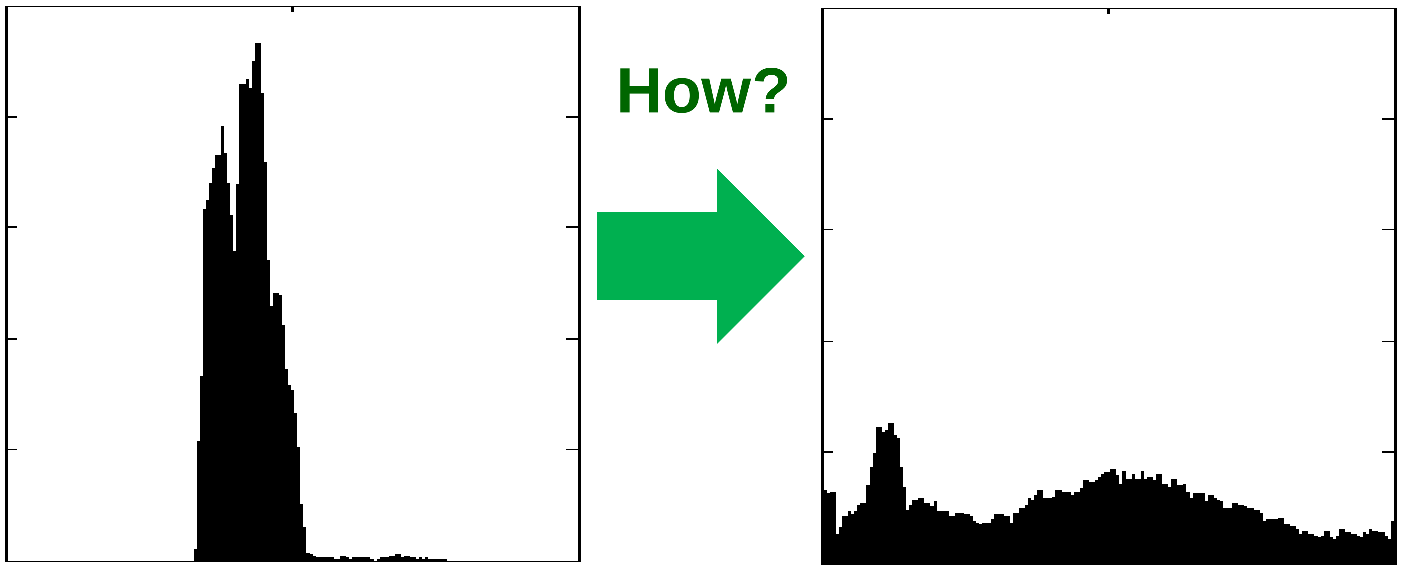
Histogram

Histogram is a very useful tool for understanding image properties regarding the distribution of intensity values.



Histogram Equalization

Goal: To derive an intensity transformation such that the resulting histogram is (almost) flat.



Histogram Equalization

Let's start with continuous r and s. (values of 0~1)

Now $p_r(r)$ and $p_s(s)$ are the normalized histograms (in the continuous sense) before and after the transformation, respectively.

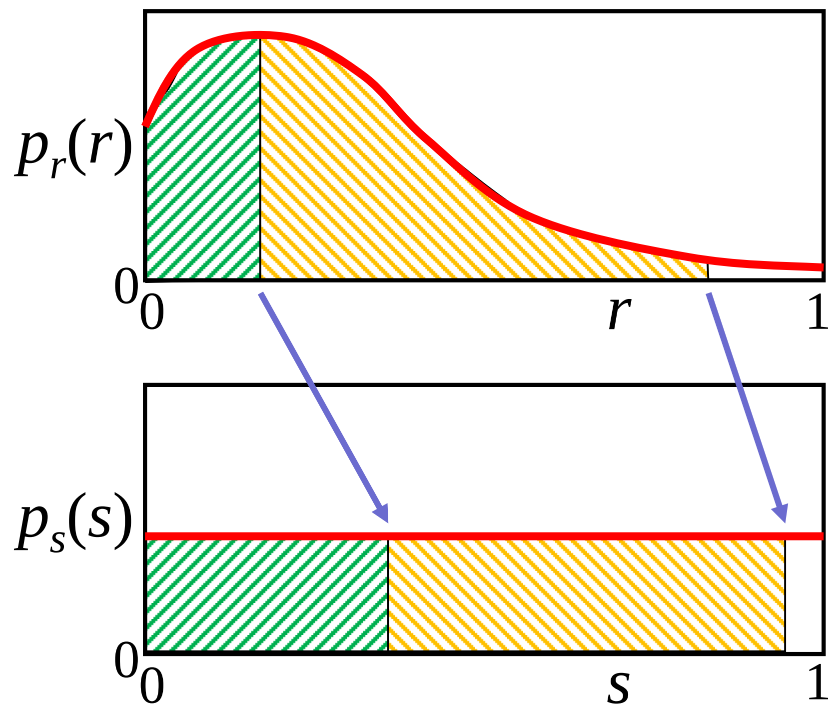
Transformation function:

$$s = T(r) = \int_0^r p_r(r')dr'$$

We get:

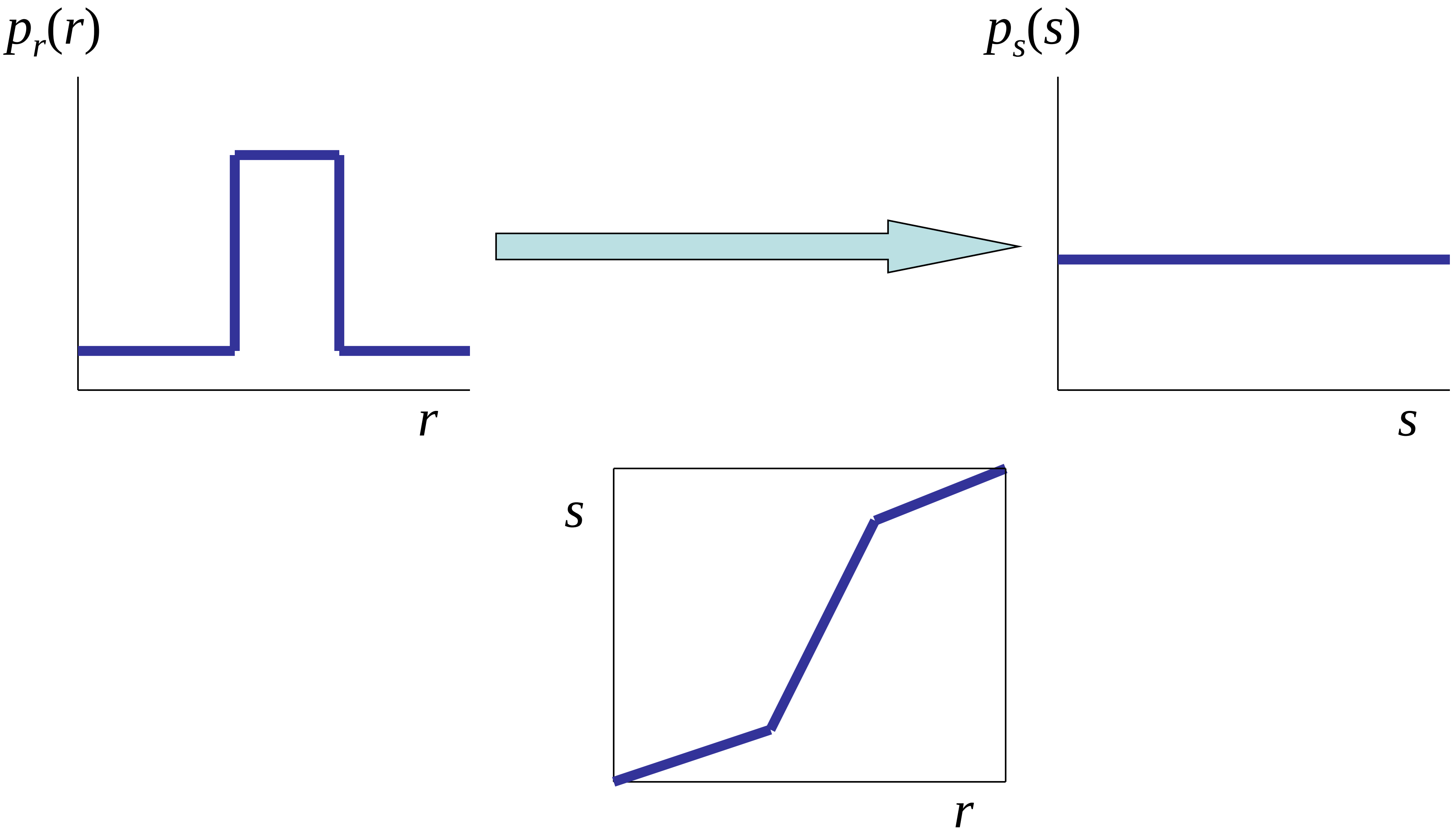
$$p_s(s) = 1$$

Illustration:



Histogram Equalization

Example:



Histogram Equalization

Now let us consider the discrete case (value range $0 \sim L-1$):

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

Transformation function:

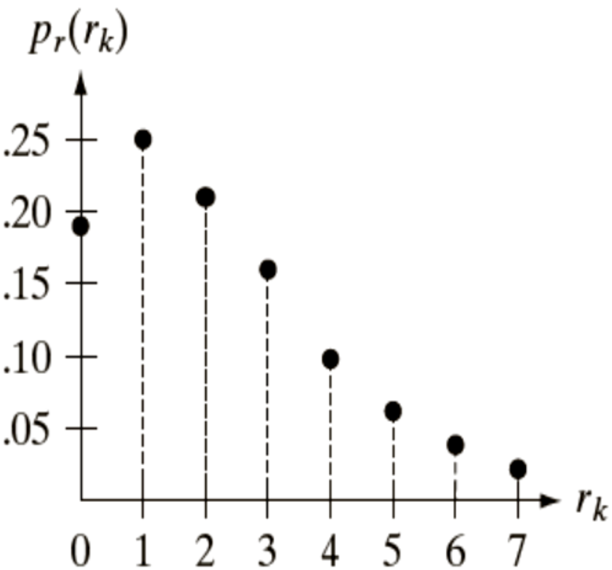
$$s_k = T(r_k) = \underbrace{(L-1)\sum_{j=0}^{k} p_r(r_j)}_{\text{We need to convert this to integers.}}$$

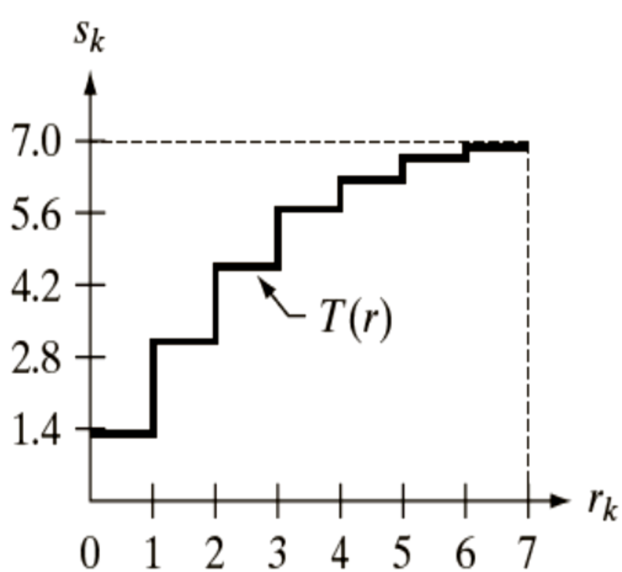
For efficiency, this transform is usually implemented as a look-up table (LUT) in practice. The same can be applied to other intensity transforms.

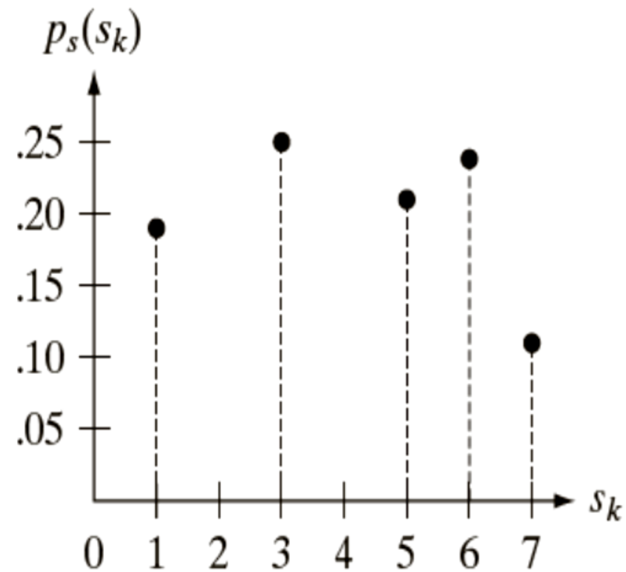
Histogram Equalization

A numerical example (L=8):

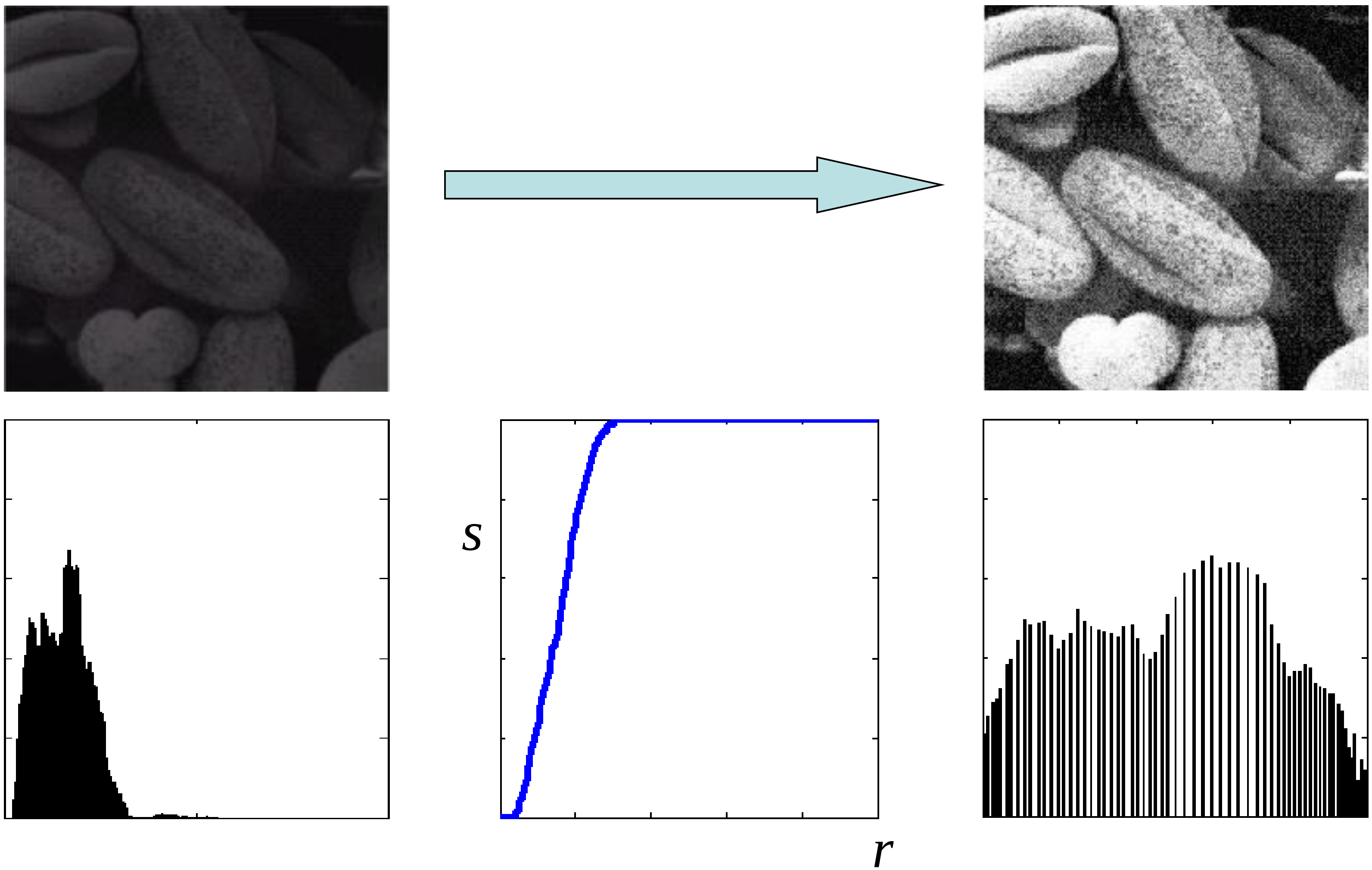
r_k	n_k	$p_r(r_k) = n_k/MN$
$r_0 = 0$	790	0.19
$r_1 = 1$	1023	0.25
$r_2 = 2$	850	0.21
$r_3 = 3$	656	0.16
$r_4 = 4$	329	0.08
$r_5 = 5$	245	0.06
$r_6 = 6$	122	0.03
$r_7 = 7$	81	0.02



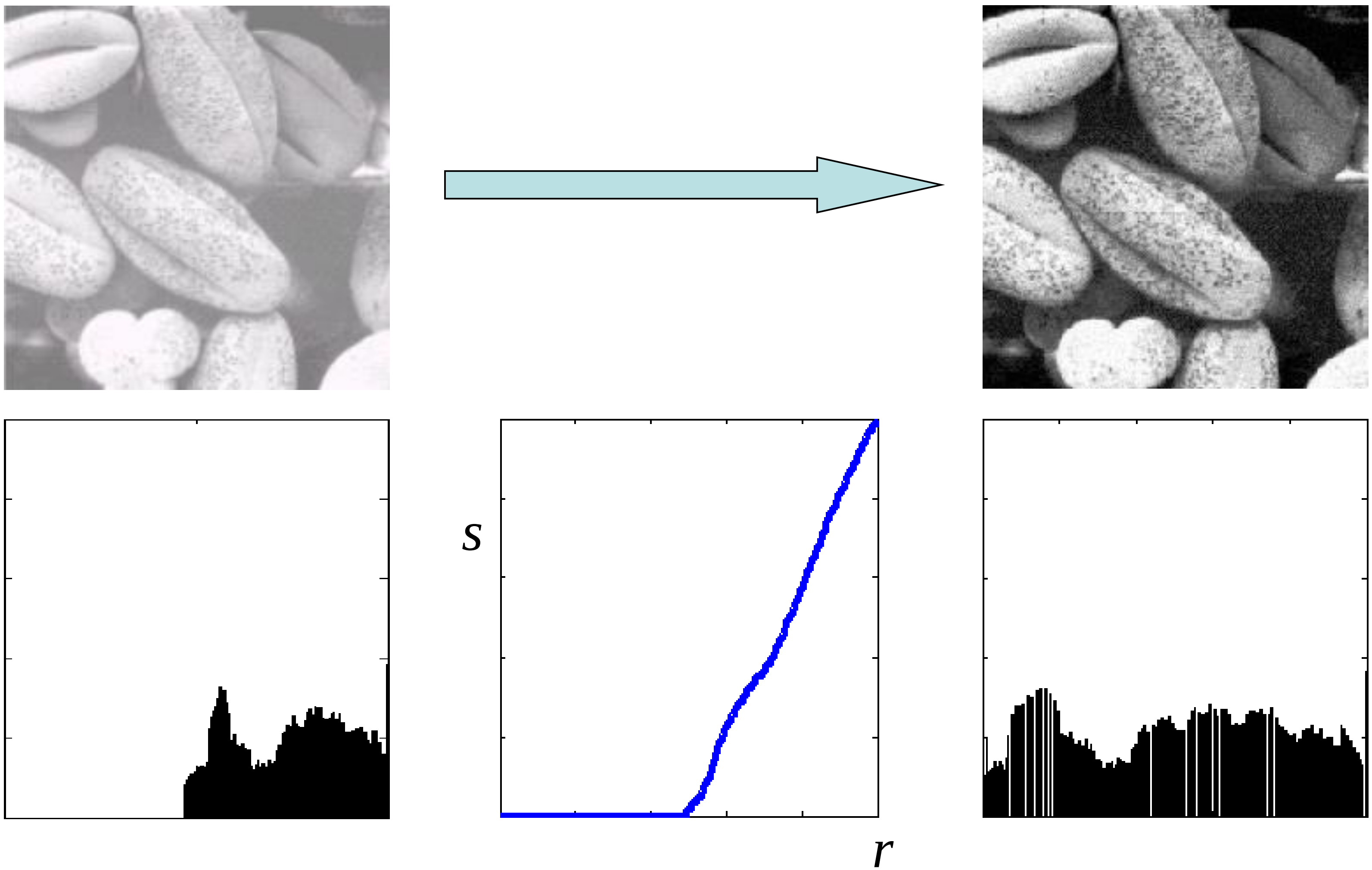




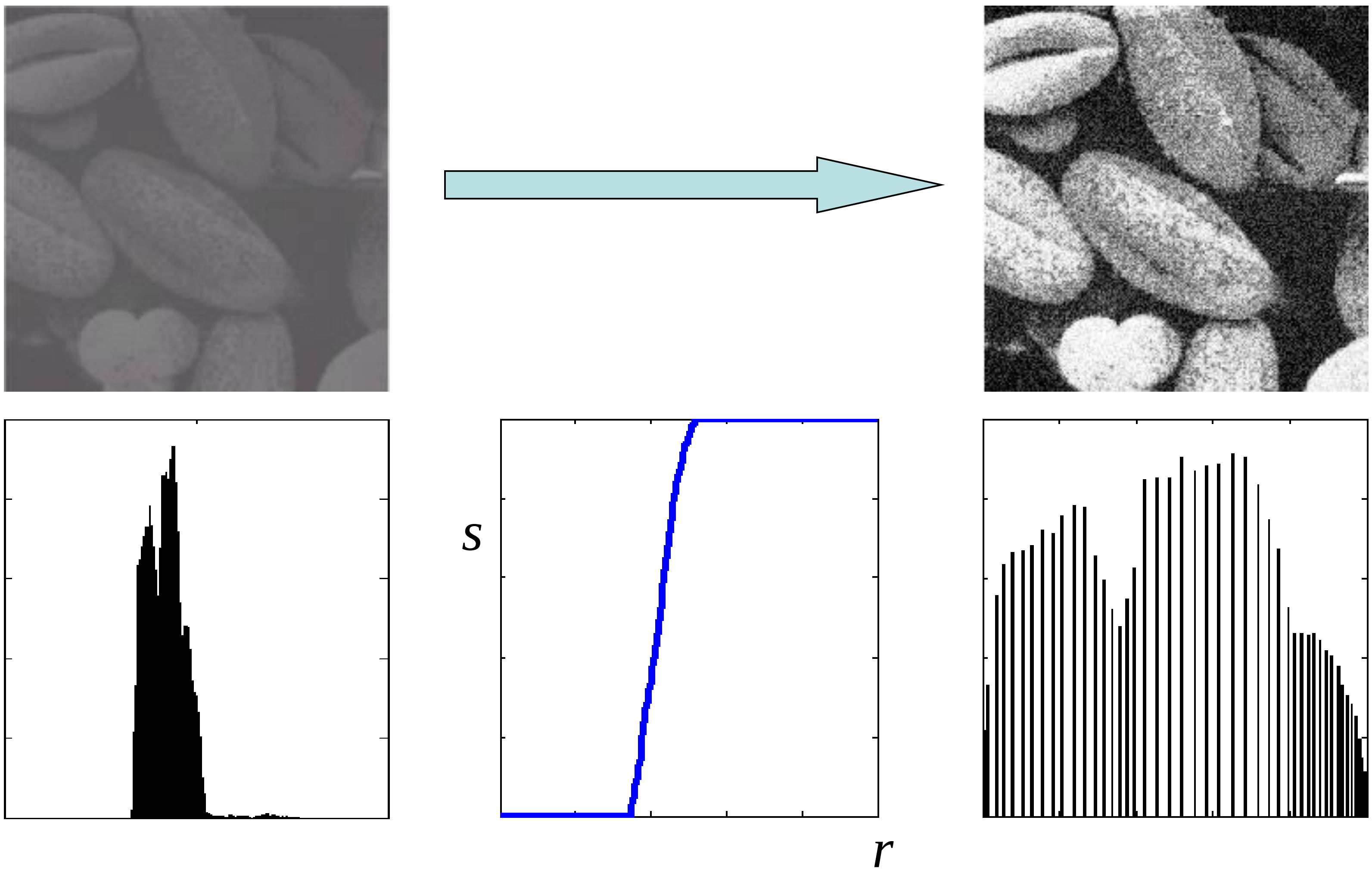
Histogram Equalization



Histogram Equalization



Histogram Equalization



Histogram Equalization