数字图像处理
Digital Image Processing

-- Statistics and Structures

钟 凡
zhongfan@sdu.edu.cn

描述一个像素

- 空间坐标 **(x, y)**
- 颜色值（**RGB, YUV, …**）

- **(x, y, R, G, B)**

多个像素?

- ……

多个像素

- 空间坐标 **(x, y)**
  - 像素与像素之间的联系（图像的结构）；
- 颜色值（**RGB, YUV, …**）
  - 像素颜色的分布（统计特征）；

直方图与图像统计

Histogram

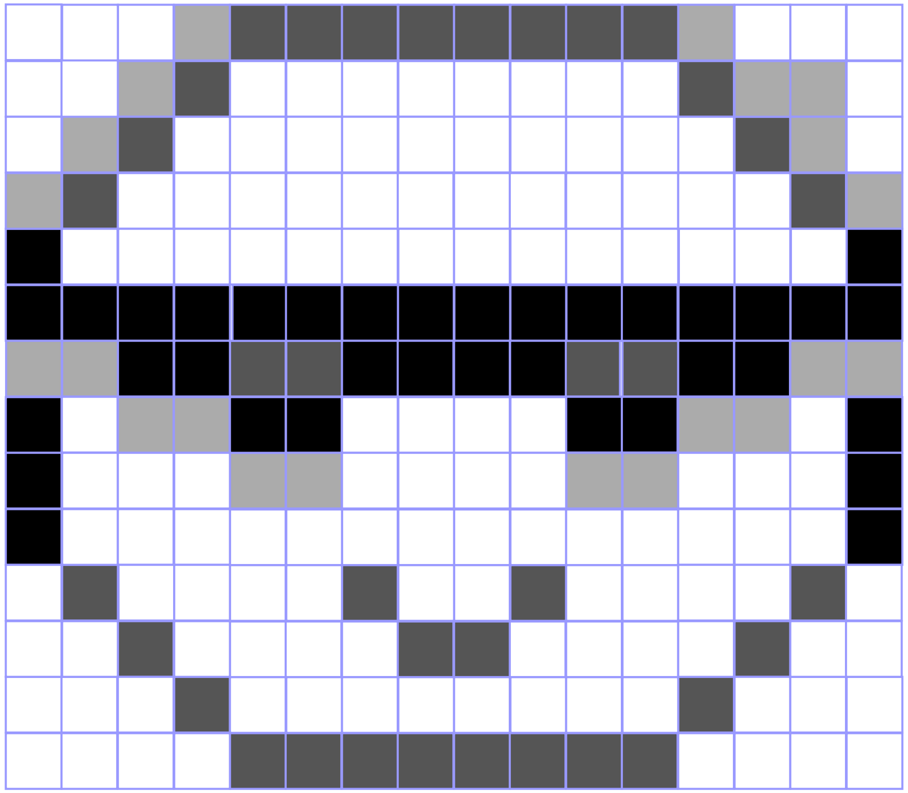
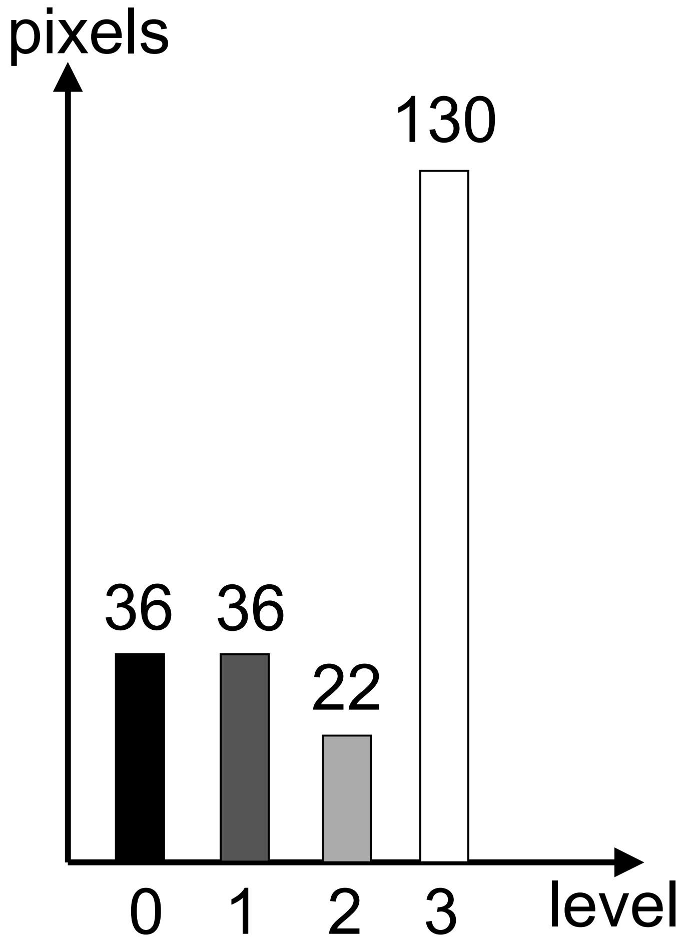


Image 16x14 = 224 pixels



Histogram Processing

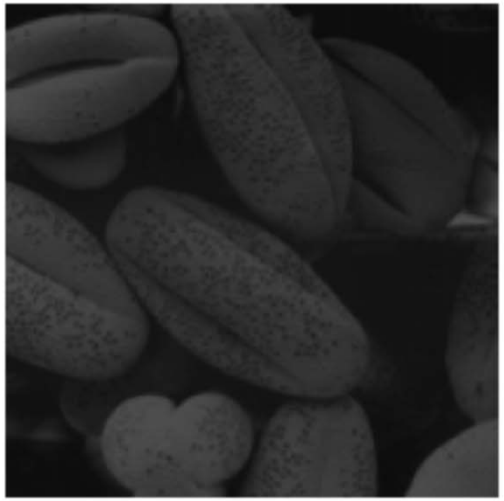
- The histogram of a digital image with gray levels from 0 to L-1 is a discrete function $h(r_k) = n_k$, where:
  - r_k is the kth gray level
  - n_k is the number of pixels in the image with that gray level
  - n is the total number of pixels in the image
  - $k = 0, 1, 2, \ldots, L-1$
- Normalized histogram: $p(r_k) = n_k / n$
  - sum of all components = 1
- The shape of the histogram of an image does provide useful information about the possibility for contrast enhancement

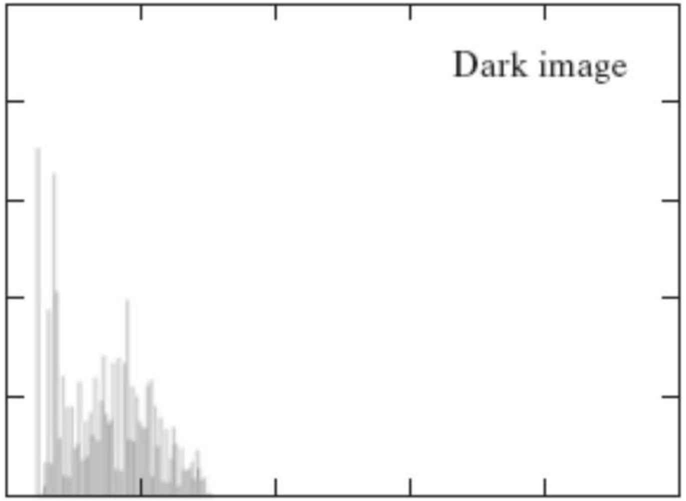
Calculate Histogram

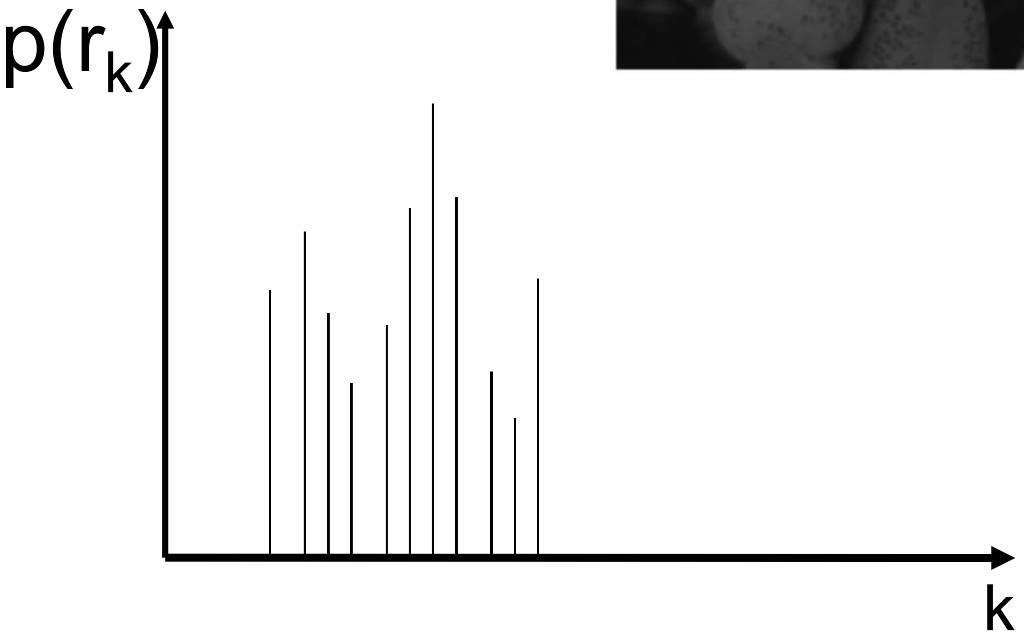
```
void calc_hist(uchar *data, int width, int height, int step, int H[256] )
{
    memset(H, 0, sizeof(H[0])*256);

    uchar *row=data;
    for(int yi=0; yi<height; ++yi, row+=step)
    {
      for(int xi=0; xi<width; ++xi)
       {
          H[ row[xi] ]++;
       }
    }
}
```

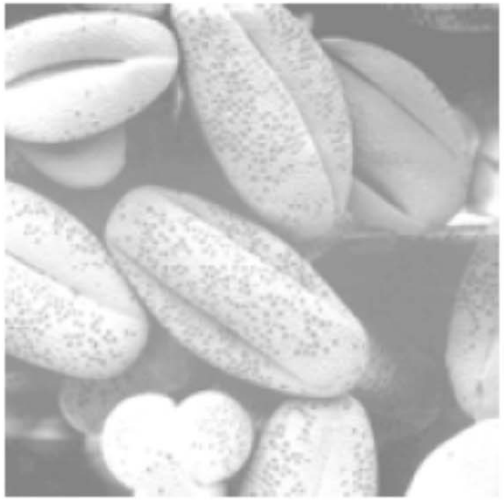
Dark Image

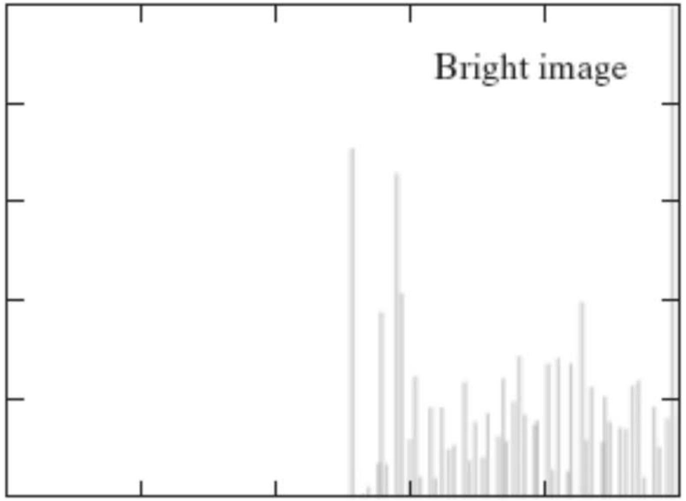


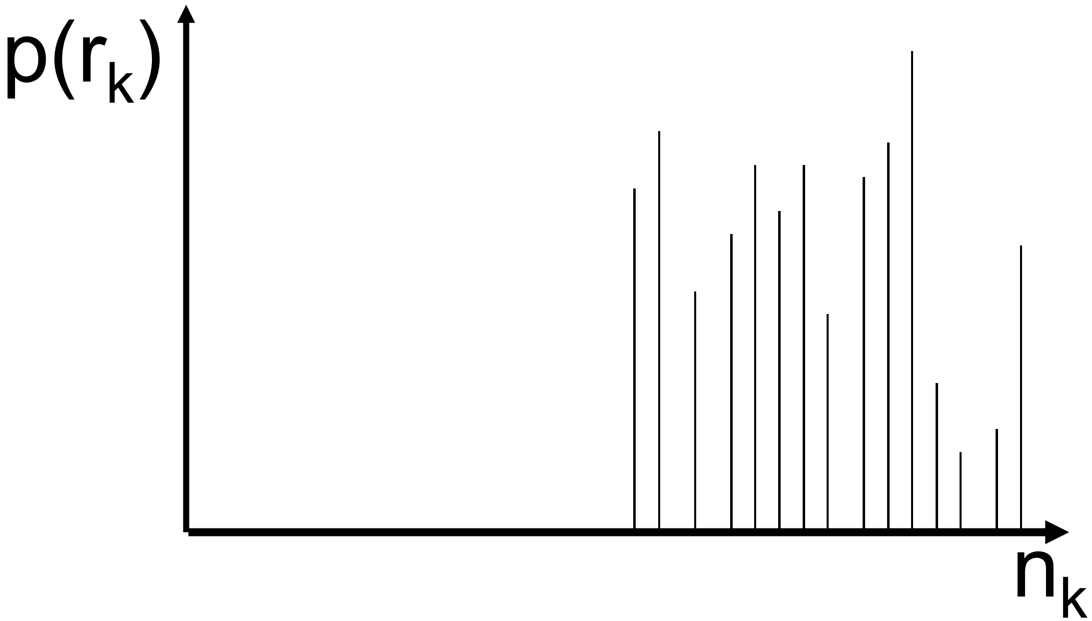




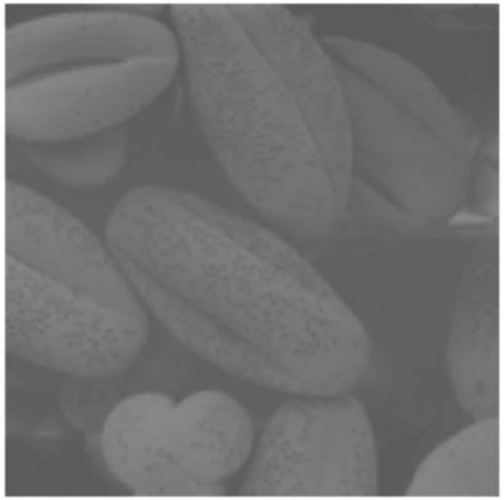
Bright Image

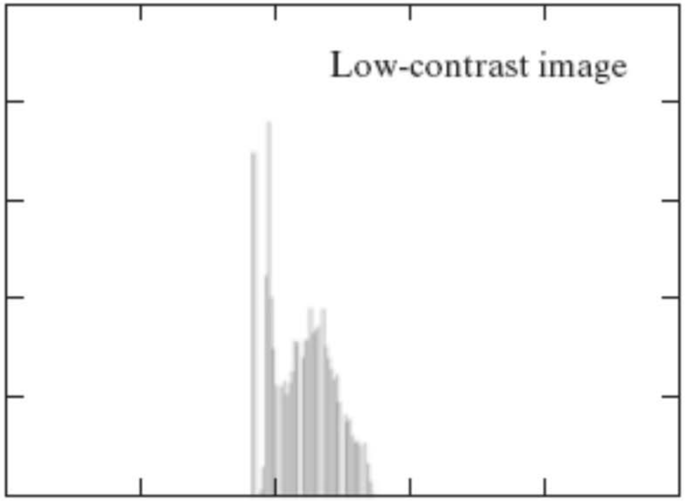


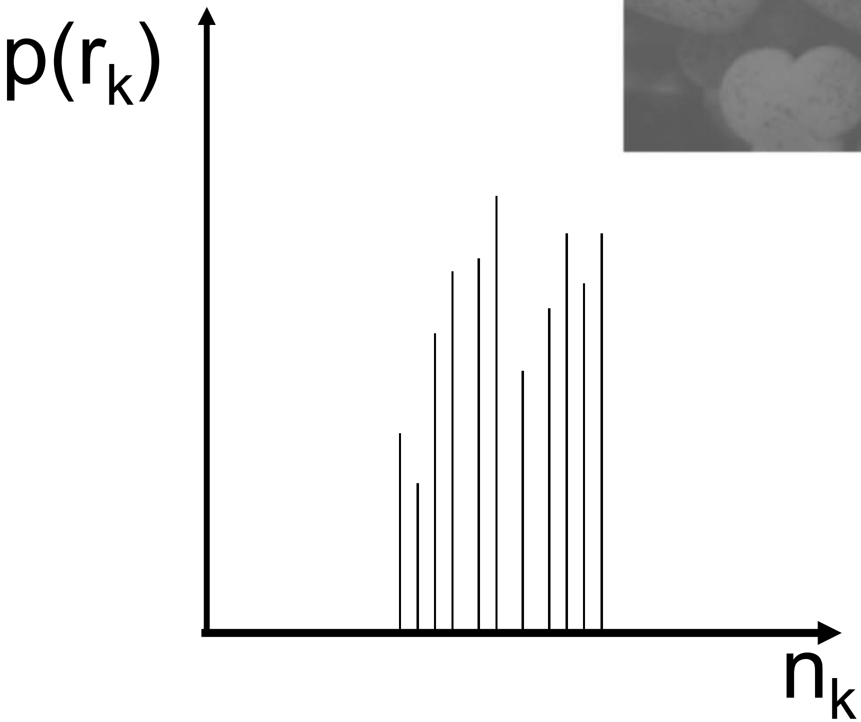




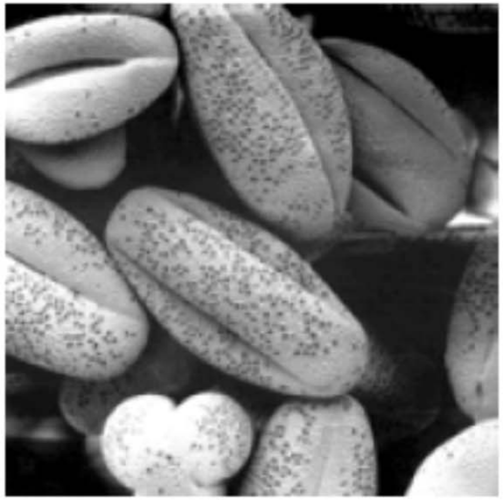
Low-Contrast Image

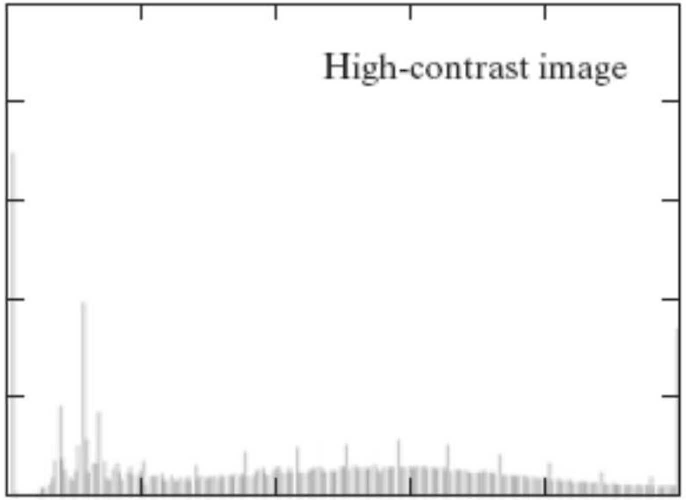


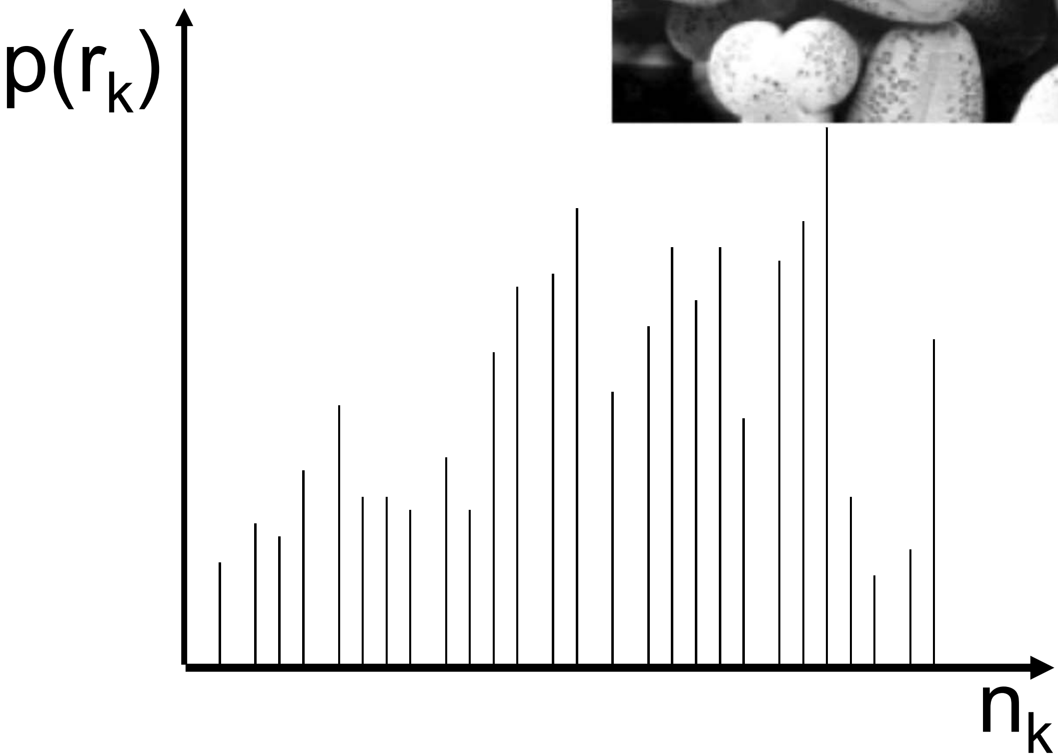




High-Contrast Image



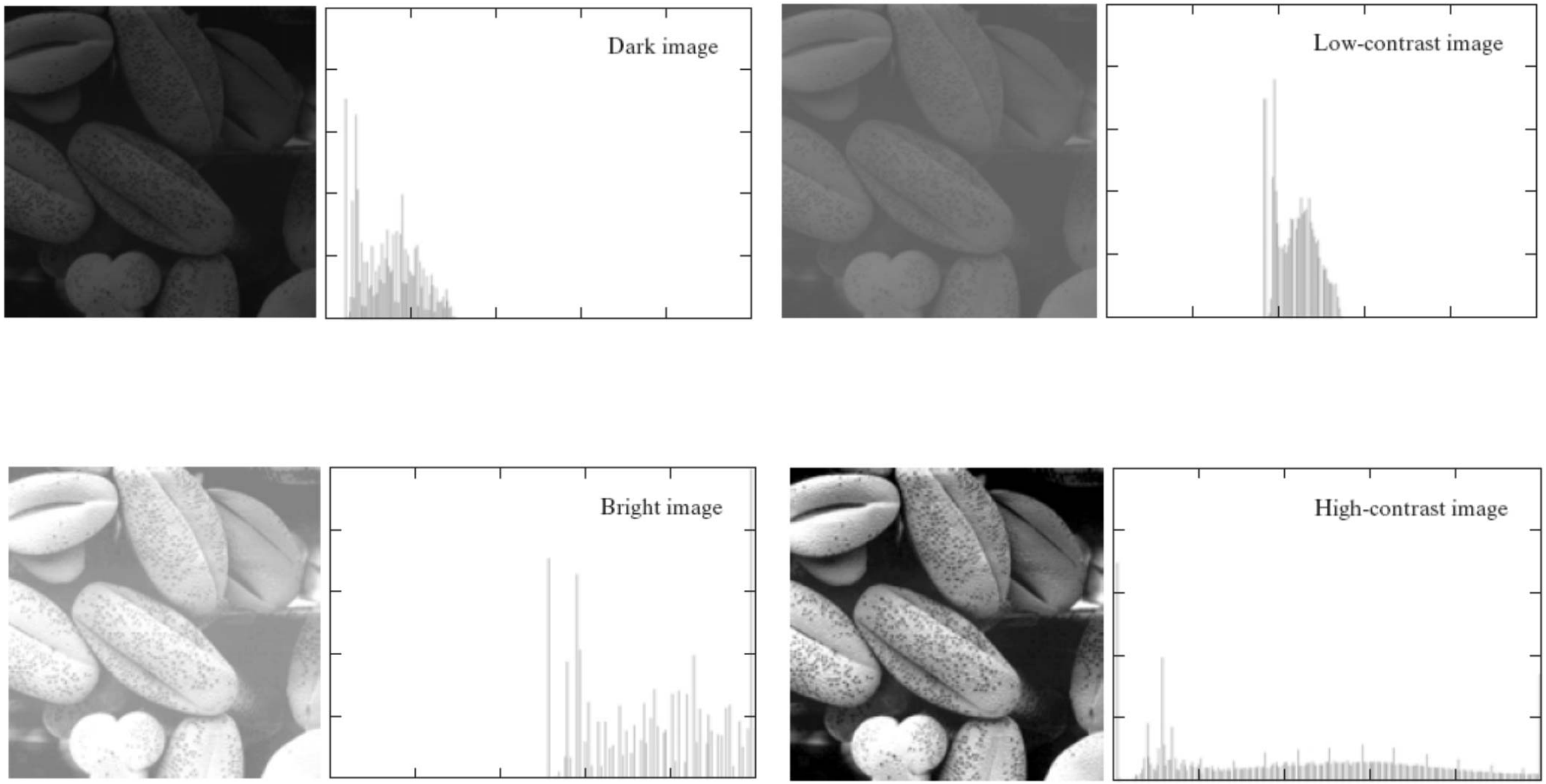




Histogram Processing

- **Histogram Equalization** 直方图均衡化
- **Histogram Matching**
- **Local Enhancement**
- **Enhancement Using Histogram Statistics**

Which looks better?

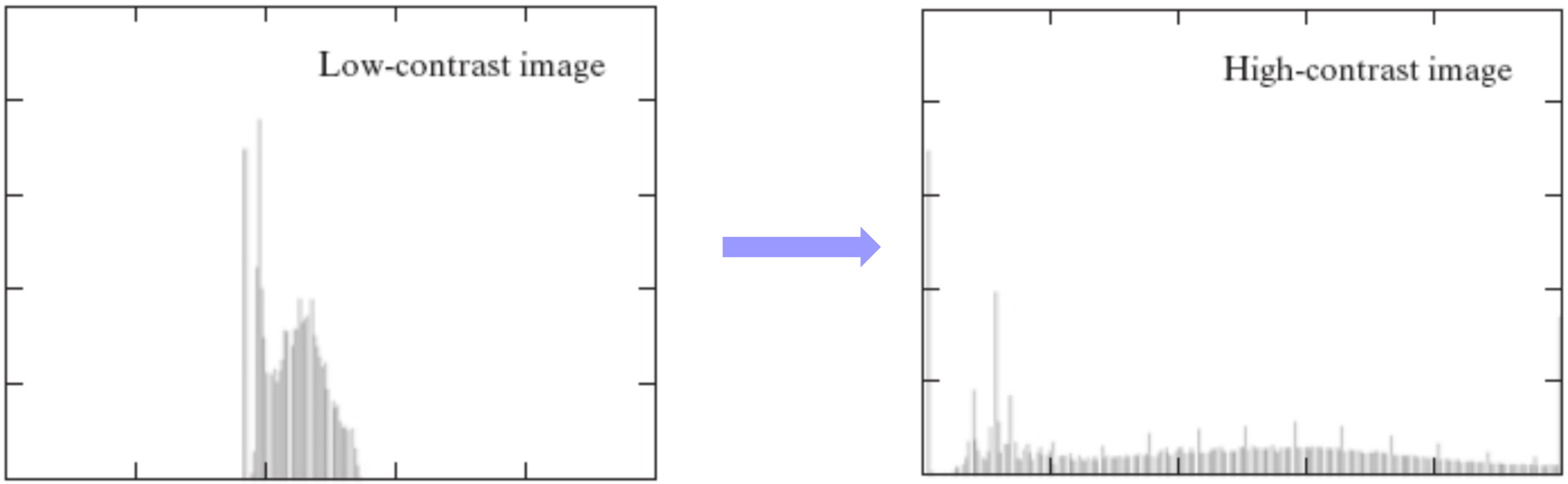


Which looks better?

- **The gray levels of pixels :**
  - **Occupy the entire range of possible gray levels**
  - **Distributed uniformly**

Histogram Equalization

- **For the above purpose…**



How to achieve?

- **Design a transformation function so that the result image occupy the entire range of gray level.**

HE Transformation Function

两个问题比较难懂，一是为什么要选用累积分布函数，二是为什么使用累积分布函数处理后像素值会均匀分布。

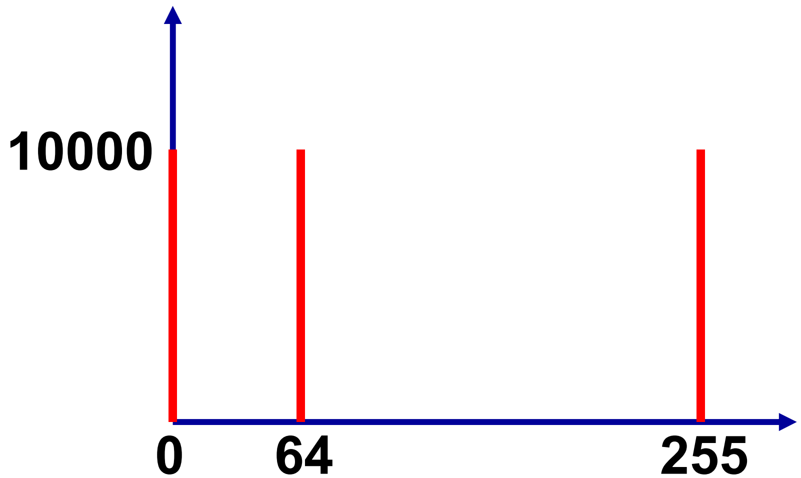
第一个问题。均衡化过程中，必须要保证两个条件：①像素无论怎么映射，一定要保证原来的大小关系不变，较亮的区域，依旧是较亮的，较暗依旧暗，只是对比度增大，绝对不能明暗颠倒；②如果是八位图像，那么像素映射函数的值域应在0和255之间的，不能越界。综合以上两个条件，累积分布函数是个好的选择，因为累积分布函数是单调增函数（控制大小关系），并且值域是0到1（控制越界问题），所以直方图均衡化中使用的是累积分布函数。

- **Normalization r → [0, 1]**
- **The gray-level transformation function used in histogram equalization:**

$$s_k = T(r_k) = \sum_{j=0}^{k} \frac{n_j}{n} = \sum_{j=0}^{k} p_r(r_j)$$

Example

- **An image (300X100)**
  - **Dark**



Example

$T(0) = 10000/30000 * 255 = 85$

$T(1) = T(0) + 0/30000 = 85$

$T(2) = T(1) + 0/30000 = 85$

…

$T(63) = T(62) + 0/30000 = 85$

$T(64) = (10000/30000 + 10000/30000)*255 = 170$

$T(65)= T(64) + 0/30000 = 170$

…

$T(253) = T(252) + 0/30000 = 170$

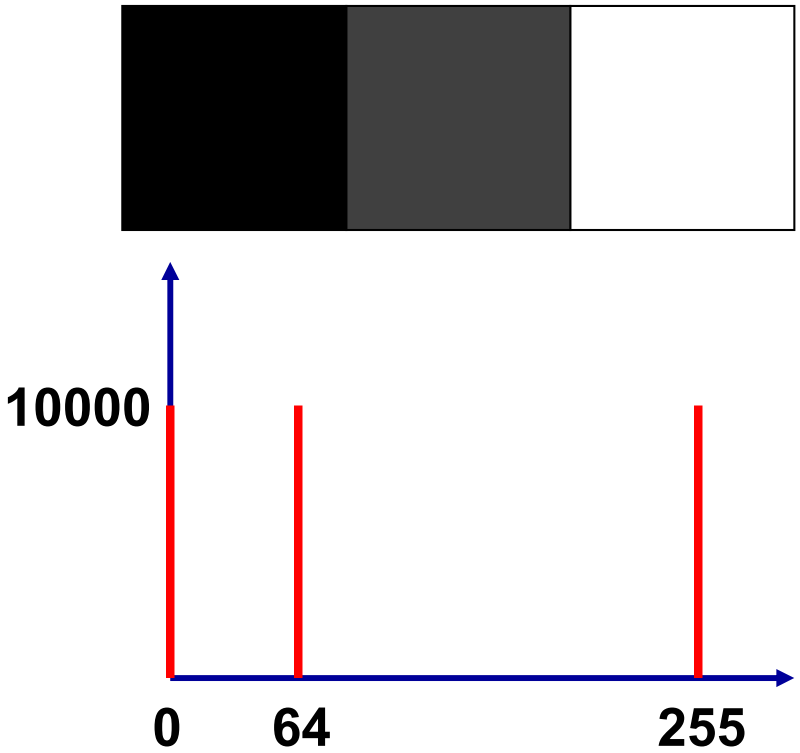
$T(254) = T(253) + 0/30000 = 170$

$T(255) = (10000/30000 + 10000/30000 + 10000/30000)*255 = 255$

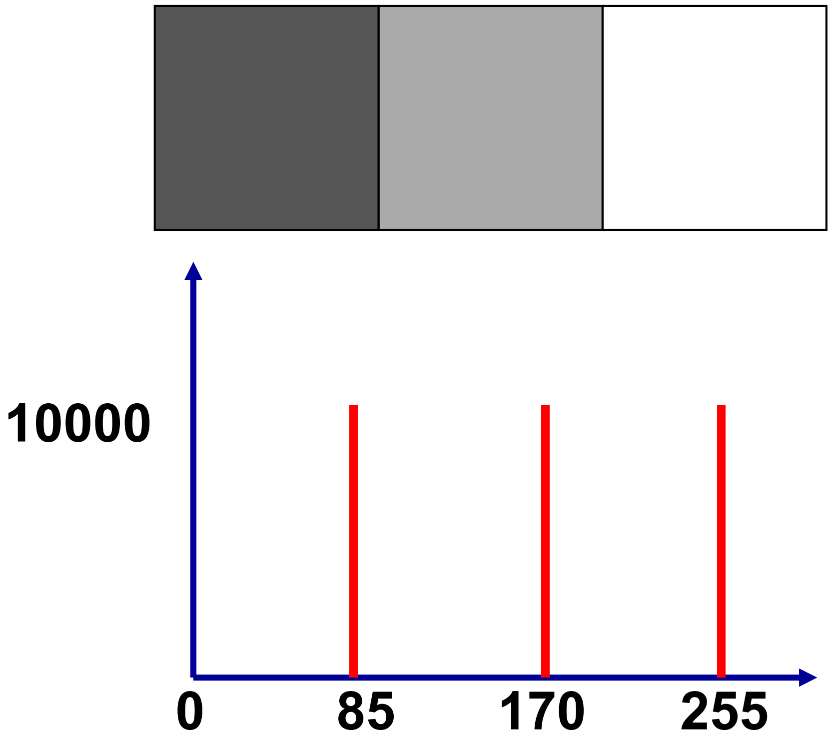
Example

- T(0) = 85 … T(63) = 85
- T(64) = 170 … T(254) = 170
- T(255) = 255

Original image



Enhanced image



Uniform?

- …

Transformation Functions

- **Consider continuous functions and transformations of the form**
  - $s = T(r) \quad r \in [0, 1]$
- **Subjected to that**
  - $T(r)$: **single-valued, monotonically increasing for** $r \in [0, 1]$
  - $T(r) \in [0, 1]$ **for** $r \in [0, 1]$
  - **Inverse transformation**
    - $r = T^{-1}(s), s \in [0, 1]$
- **Probability density function of** s

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

Transformation Functions

- **Probability density function of** s

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

- **Transformation function**

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

  - **Which will satisfy that**

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

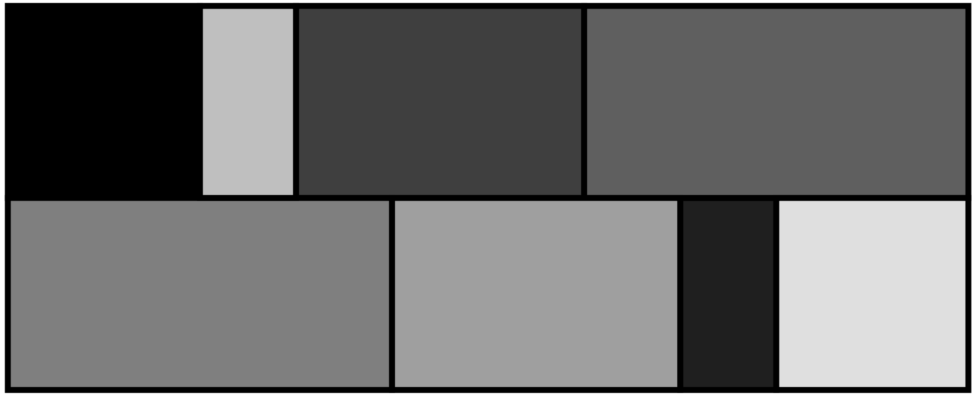
  - **Then s has**

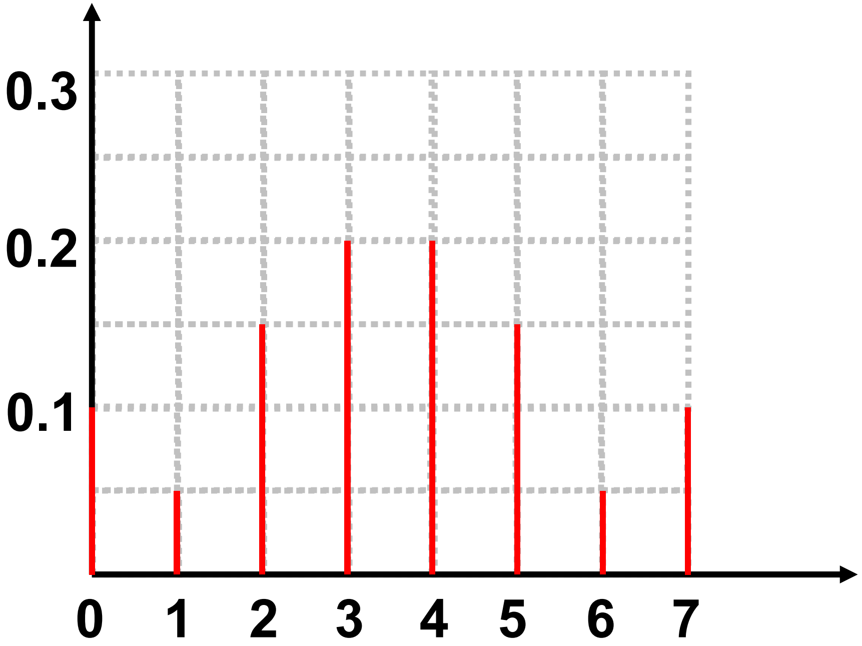
$$p_s(s) = p_r(r)\left|\frac{dr}{ds}\right| = p_r(r)\frac{1}{p_r(r)} = 1$$

The case of discrete

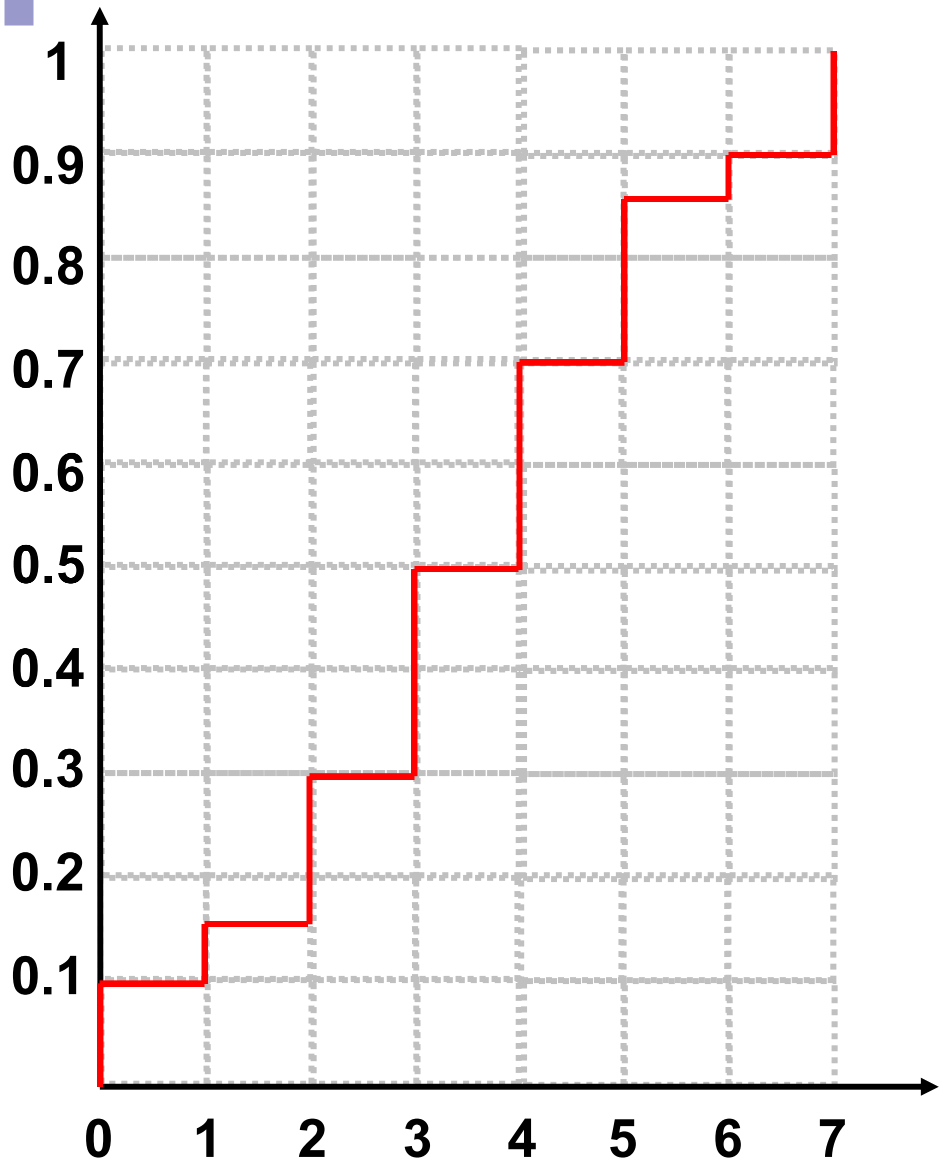
- Still uniform?

Example





Histogram of original image



Cumulate histogram

$T(0) = 0.1 \times 7 = 0.7 \rightarrow 1$

$T(1) = 0.15 \times 7 = 1.05 \rightarrow 1$

$T(2) = 0.3 \times 7 = 2.1 \rightarrow 2$

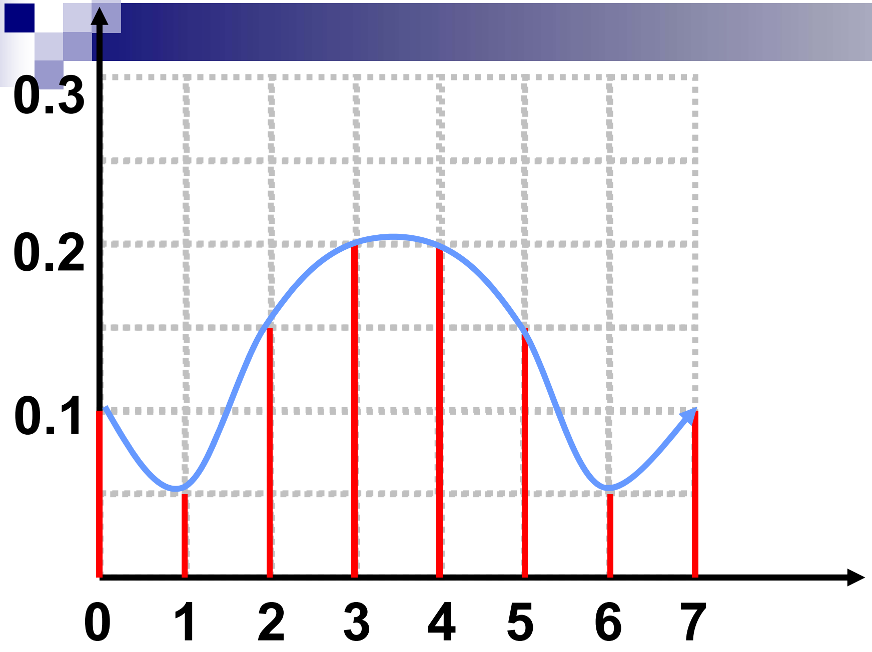
$T(3) = 0.5 \times 7 = 3.5 \rightarrow 4$

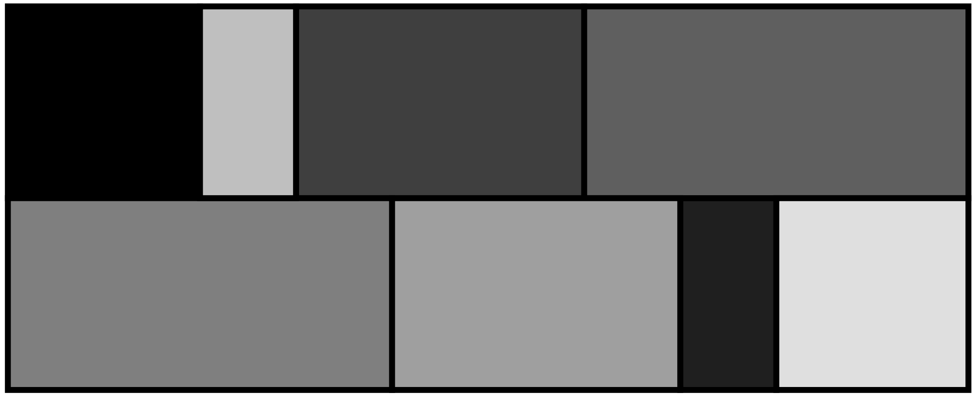
$T(4) = 0.7 \times 7 = 4.9 \rightarrow 5$

$T(5) = 0.85 \times 7 = 5.95 \rightarrow 6$

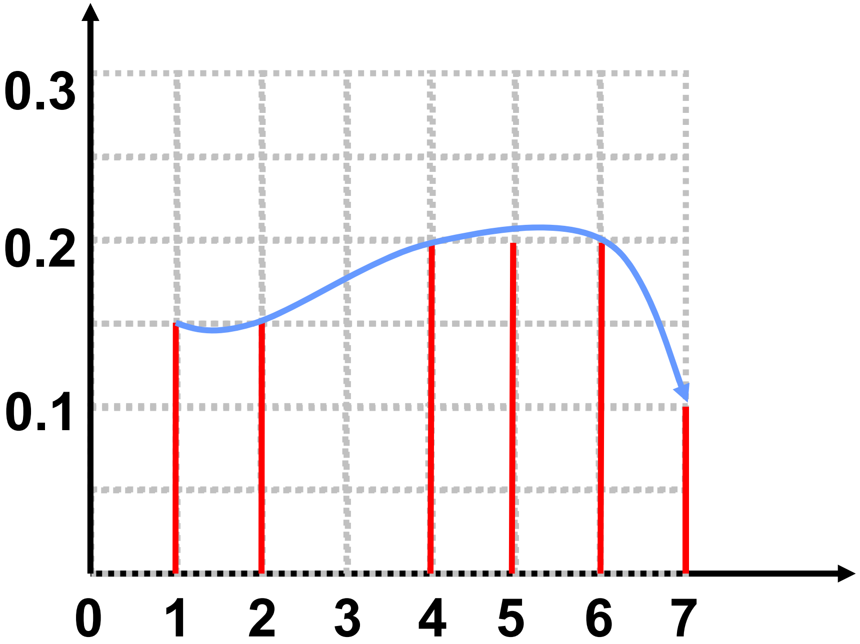
$T(6) = 0.9 \times 7 = 6.3 \rightarrow 6$

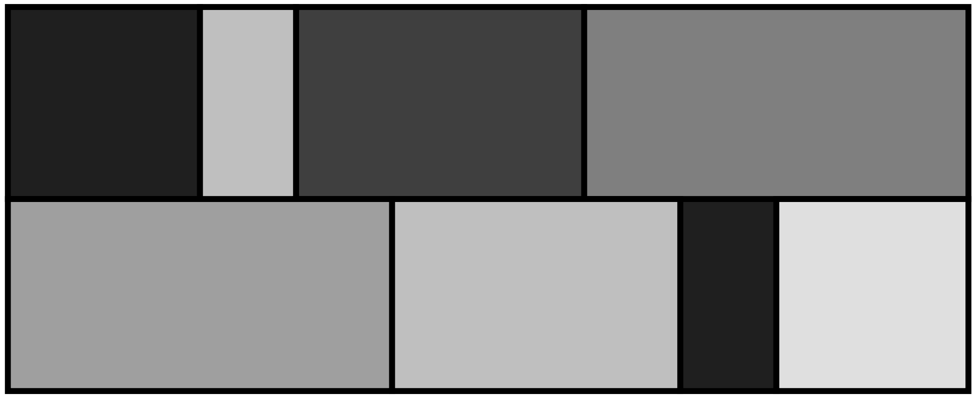
$T(7) = 1 \times 7 = 7 \rightarrow 7$





T(0) → 1, T(1) → 1,
T(2) → 2, T(3) → 4,
T(4) → 5, T(5) → 6,
T(6) → 6, T(7) → 7.





Why not uniform?

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

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

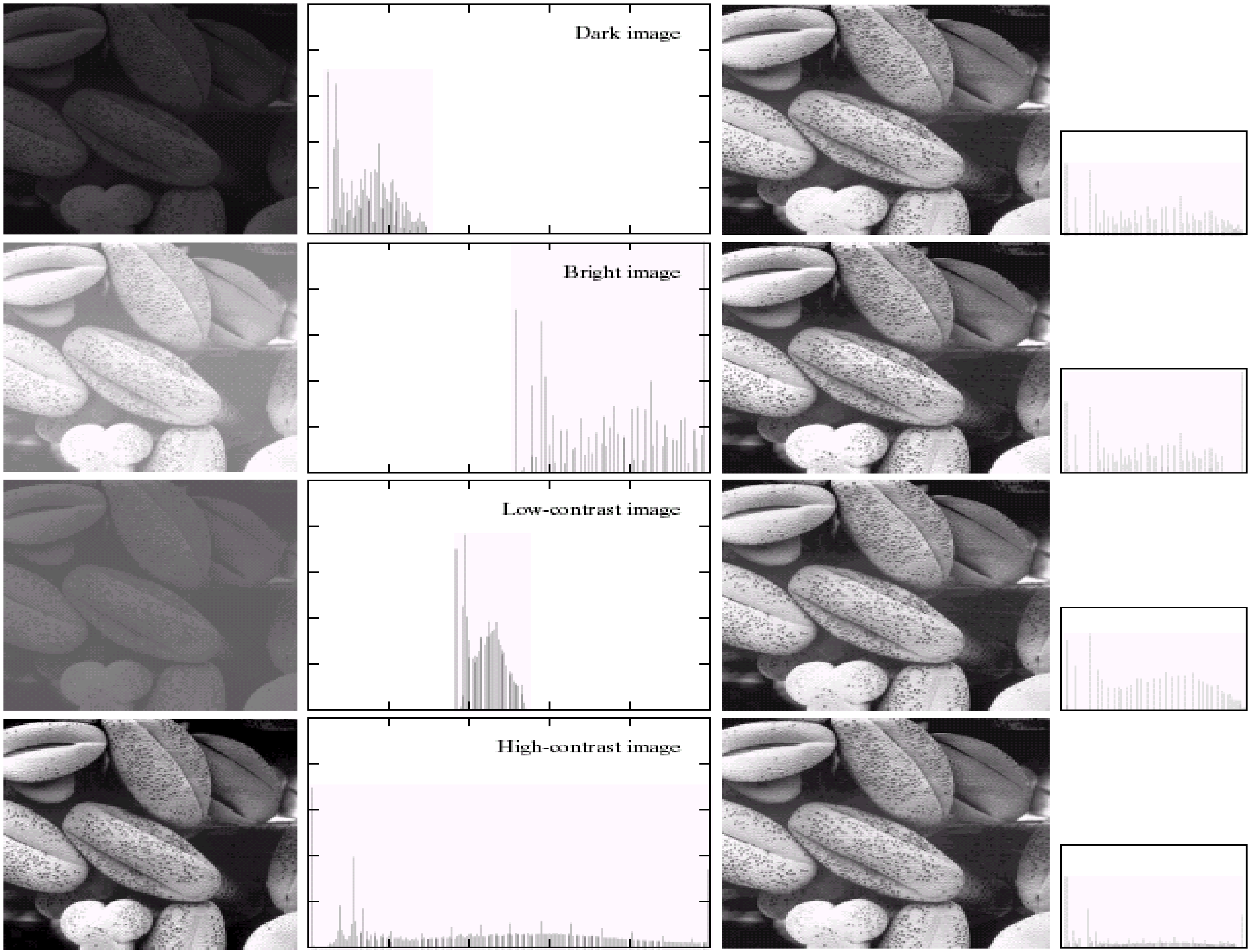
$$p_s(s) = p_r(r)\left|\frac{dr}{ds}\right| = p_r(r)\frac{1}{p_r(r)} = 1$$

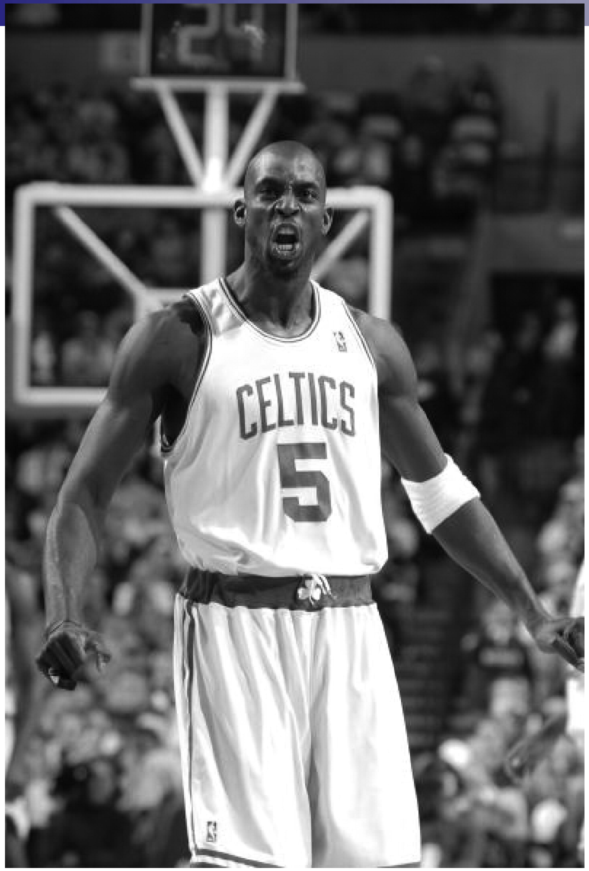
Histogram Equalization

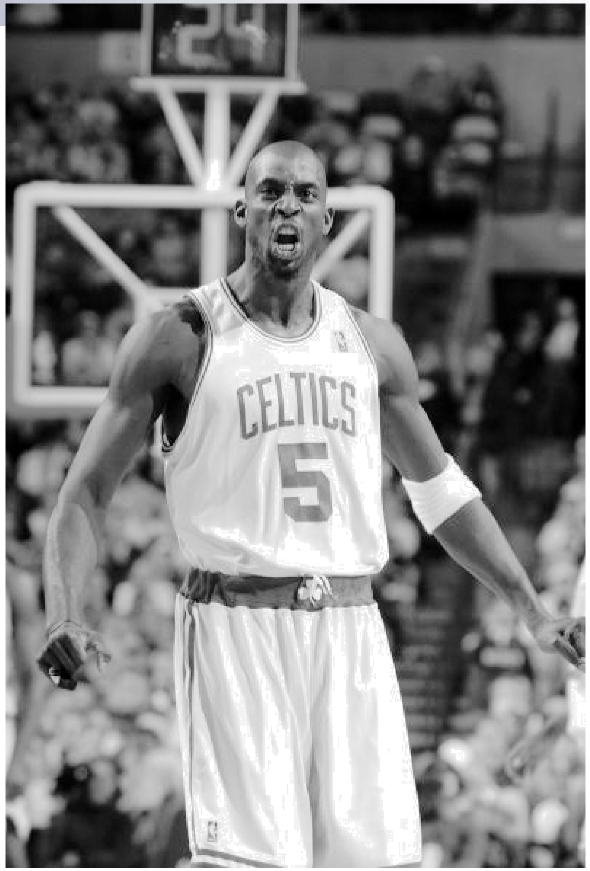
- The transformation :

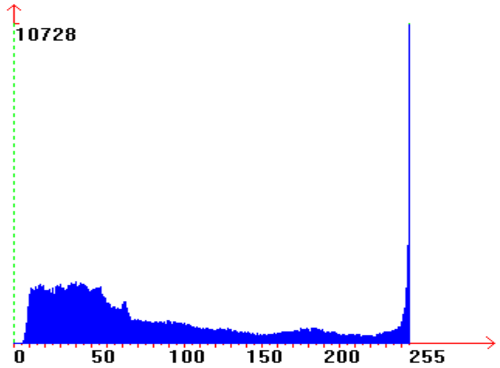
$$s_k = T(r_k) = \sum_{j=0}^{k} \frac{n_j}{n} = \sum_{j=0}^{k} p_r(r_j)$$

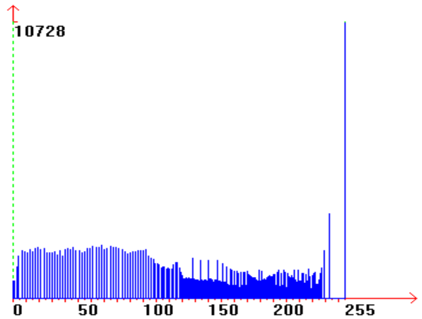
  - This will not produce a uniform histogram, but will tend to spread the histogram of the input image
  - Advantages:
    - Gray-level values cover entire scale (contrast enhancement)
    - Fully automatic

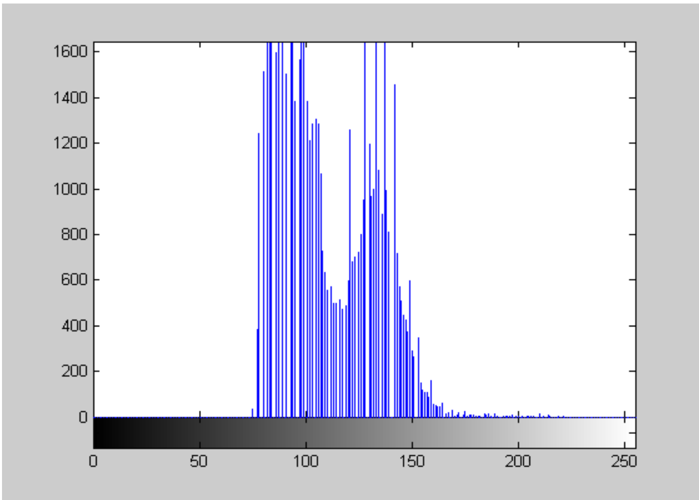

Dark image
Bright image
Low-contrast image
High-contrast image

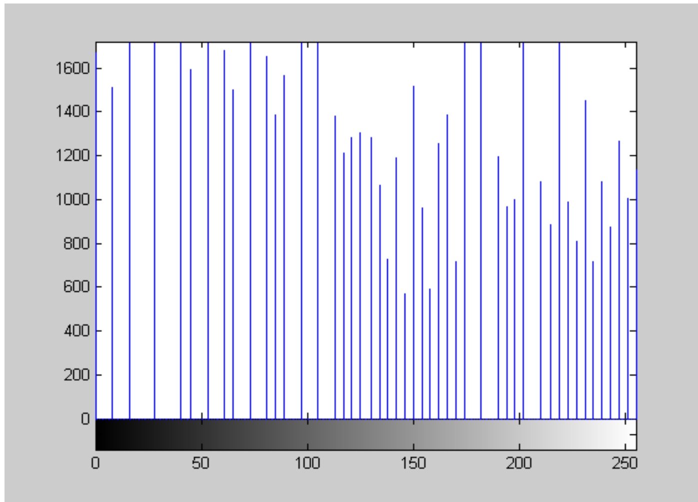


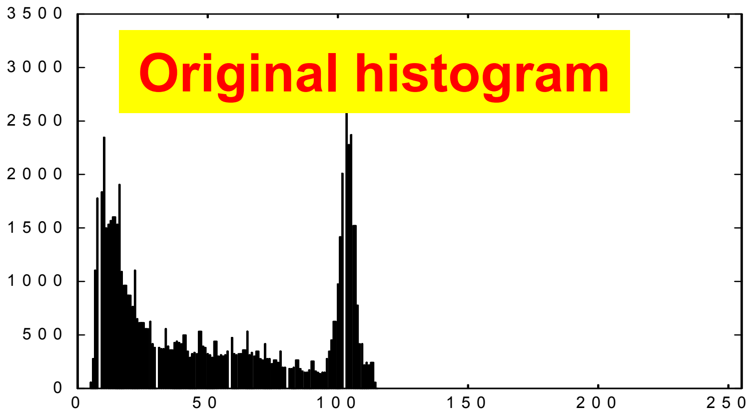


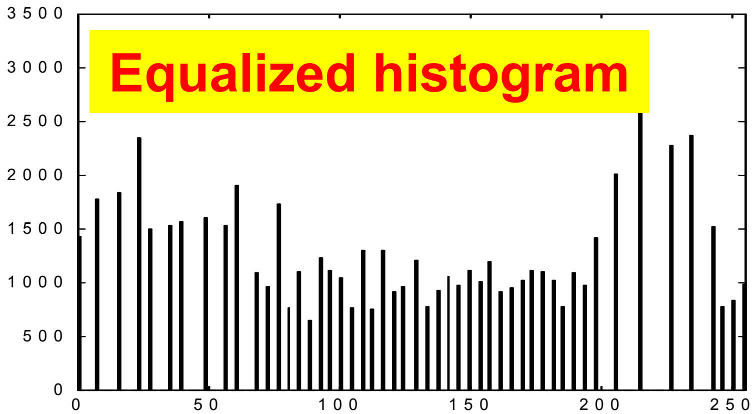

10728
0
50
100
150
200
255


10728
0
50
100
150
200
255


1600
1400
1200
1000
800
600
400
200
0
0
50
100
150
200
250


1600
1400
1200
1000
800
600
400
200
0
0
50
100
150
200
250


Original histogram
3500
3000
2500
2000
1500
1000
500
0
0
50
100
150
200
250


Equalized histogram
3500
3000
2500
2000
1500
1000
500
0
0
50
100
150
200
250

- Histogram Processing
  - Histogram Equalization
  - Histogram Matching
  - Local Enhancement
  - Enhancement Using Histogram Statistics

Histogram Matching

直方图均衡化，是我们利用直方图处理数字图像的一种方法。但实际情况中，我们有了一个模板，需要进行灰度级的转换，使得转换后的图像灰度按照给定的规律分布，即直方图与给定模板的匹配。 匹配的基本思路是，找到一个中间状态，即前面所说的均衡化状态。左右两个过程以均衡化为中心和连接点：

原直方图 <=> 均衡化直方图 <=> 匹配要求直方图

用r(k)表示原图灰度级，s(k)为原图均衡化后的灰度级；用z(k)表示匹配图的灰度级，t(k)为匹配图均衡化后的灰度级。

对于r(k)，可以找到s(k)对应，同时遍历z(k)，找到一个最小的z(k)使得z(k)所对应的t(k)最接近s(k)。此时，即建立了一个r(k)~z(k)的映射，而s(k)、t(k)只是进行匹配的中间状态

Original

Equalized

Matched

1、把原图和匹配图都均衡化，并得到匹配图逆均衡化的对应关系。
2、由于都是均衡化，可以理解为相等，所以把原图均衡化后的点，按照匹配图的逆均衡化对应关系，得到结果图。
如果用T(r)和G(z)分别表示原图和匹配图的均衡化变换函数。那么结果可以表示为：
z=G 1[T(r)]

Histogram Matching

- **Special request for histogram of enhanced image**
- **→ Histogram matching (or histogram specification)**
  - **Histogram equalization does not allow interactive image enhancement and generates only one result: an approximation to a uniform histogram.**
  - **Sometimes though, we need to be able to specify particular histogram shapes capable of highlighting certain gray-level ranges.**

Procedure for Histogram Matching

- **Equalize the levels of the original image using:**

$$s = T(r_k) = \sum_{j=0}^{k} \frac{n_j}{n}$$

- **Specify the desired density function and obtain the transformation function G(z):**

$$v = G(z) = \sum_{0}^{z} p_z(w) \approx \sum_{i=0}^{z} \frac{n_i}{n}$$

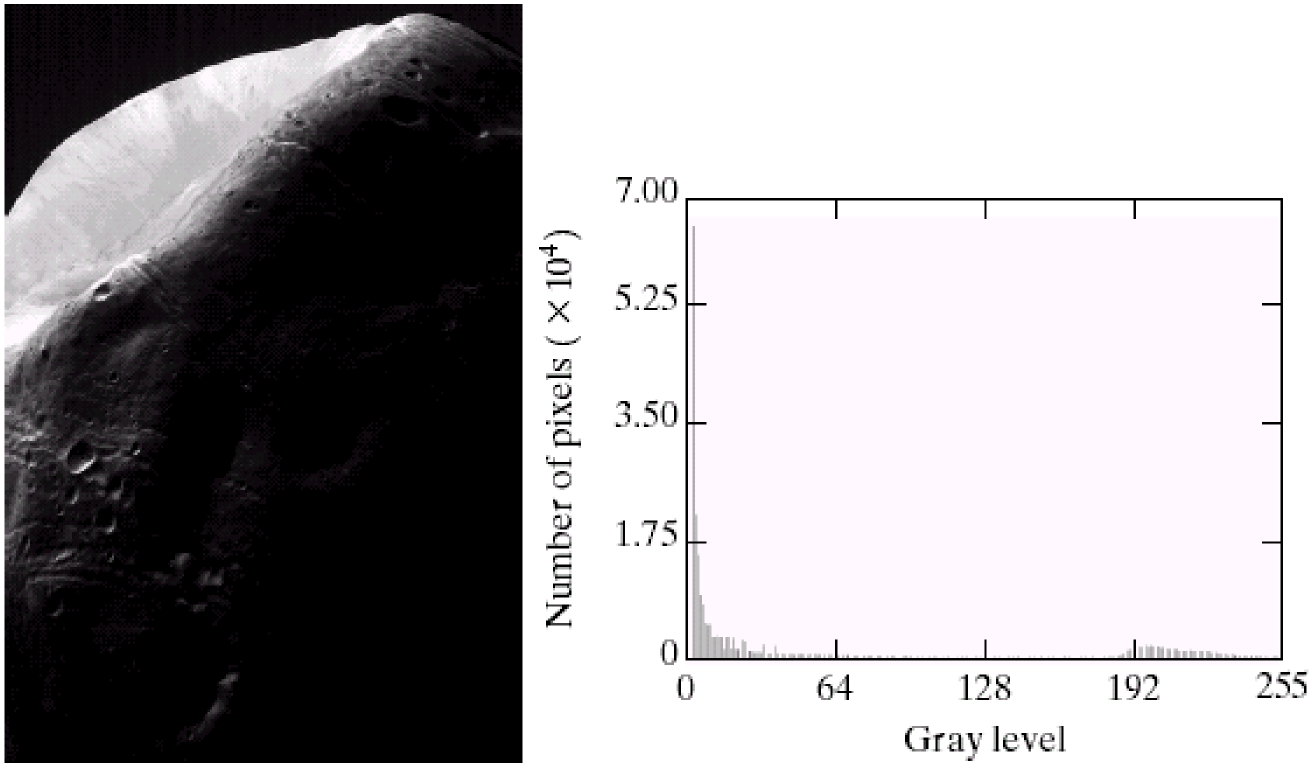
- **Apply the inverse transformation function $z = G^{-1}(s)$ to the levels obtained in step 1.**
- **The new, processed version of the original image consists of gray levels characterized by the specified density $p_z(z)$. In essence:**

$$z = G^{-1}(s) \rightarrow z = G^{-1}[T(r)]$$

Histogram Matching

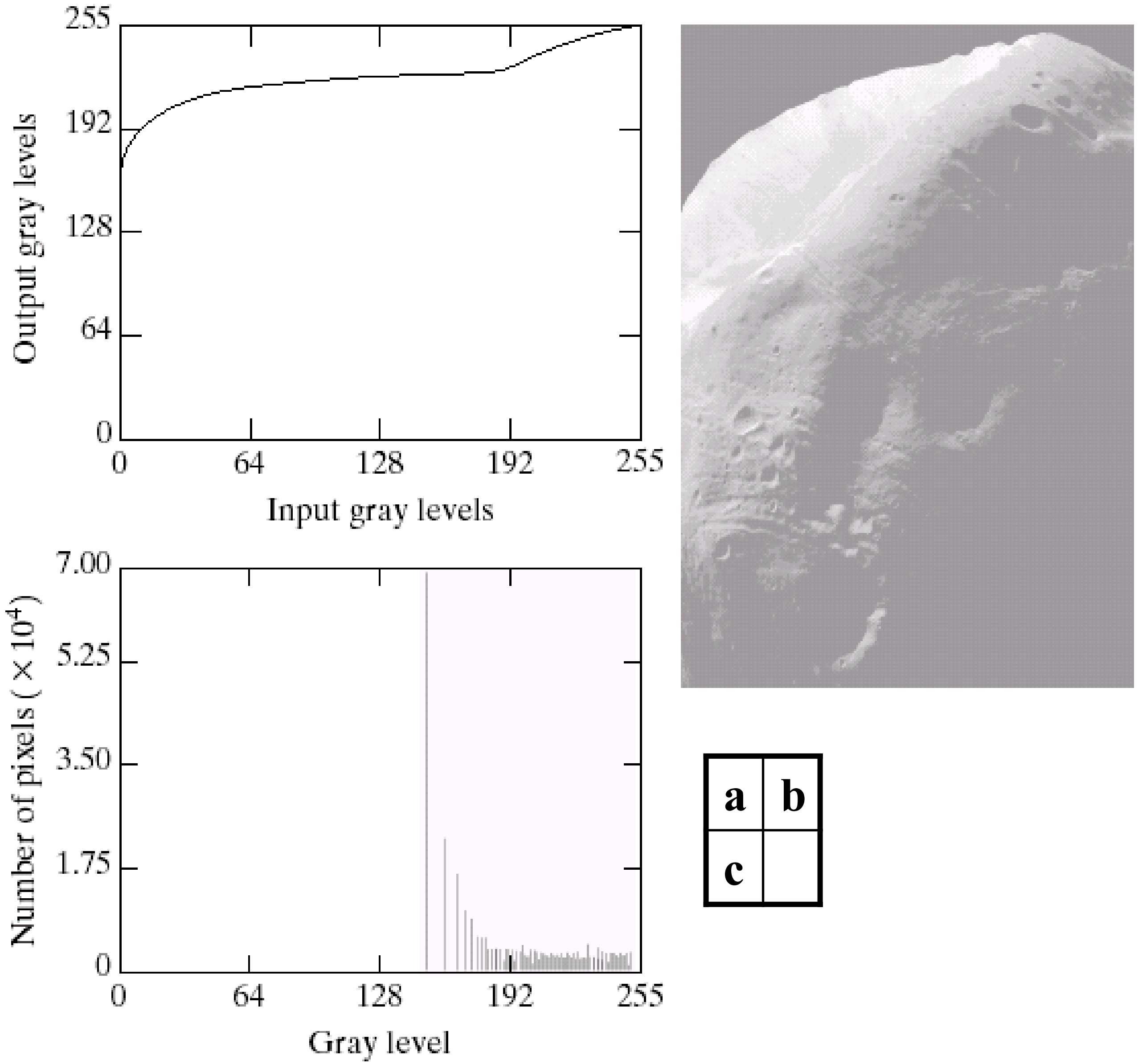
直方图匹配的难点在于构建有意义的匹配直方图

- The principal difficulty in applying the histogram specification method to image enhancement lies in being able to construct a meaningful histogram. So:
  - Either a particular probability density function (such as a Gaussian density) is specified and then a histogram is formed by digitizing the given function,
  - Or a histogram shape is specified on a graphic device and then is fed into the processor executing the histogram specification algorithm.

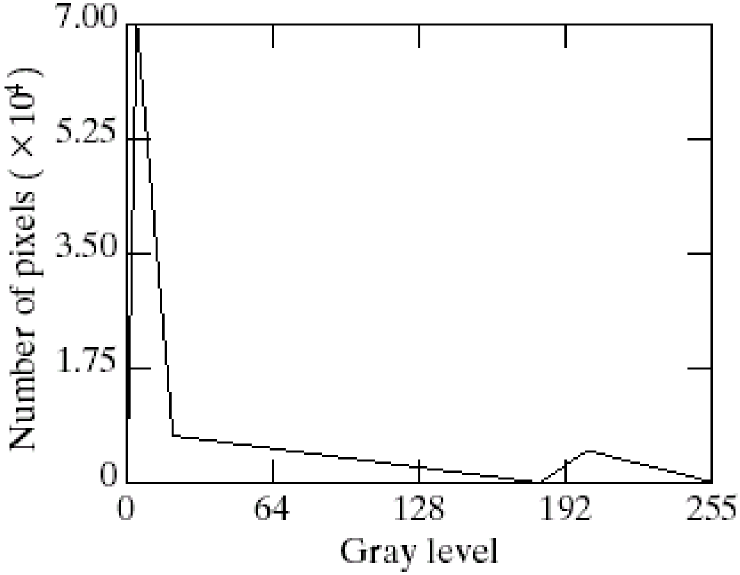


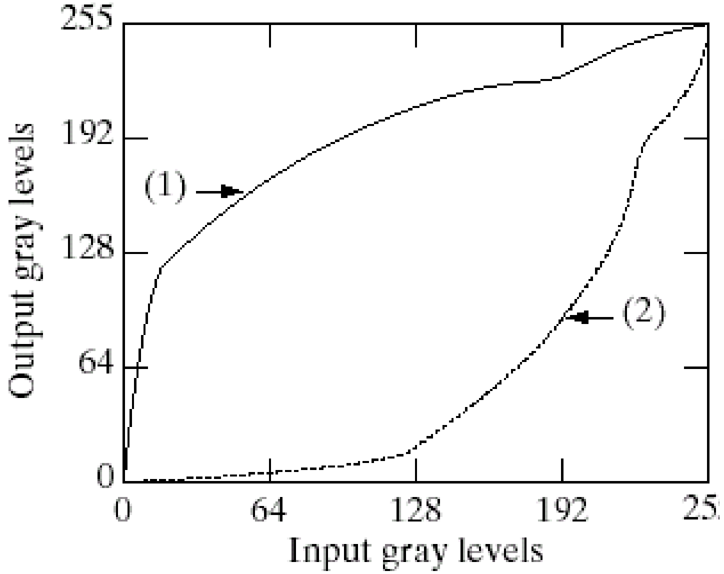
a | b

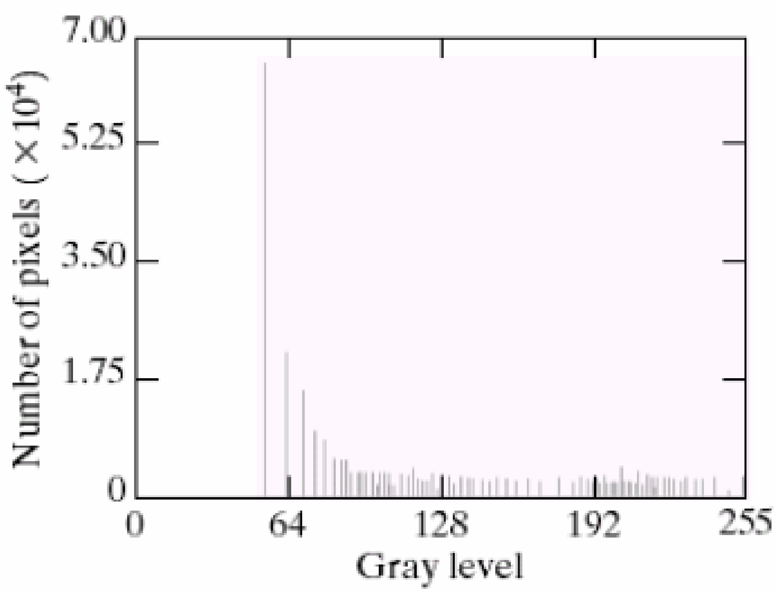
(a) Image of the Mars moon Photos taken by NASA's Mars Global Surveyor. (b) Histogram. (Original image courtesy of NASA.)



(a) Transformation function for histogram equalization. (b) Histogram equalized image (note the was held out appearance). (c) Histogram of (b).





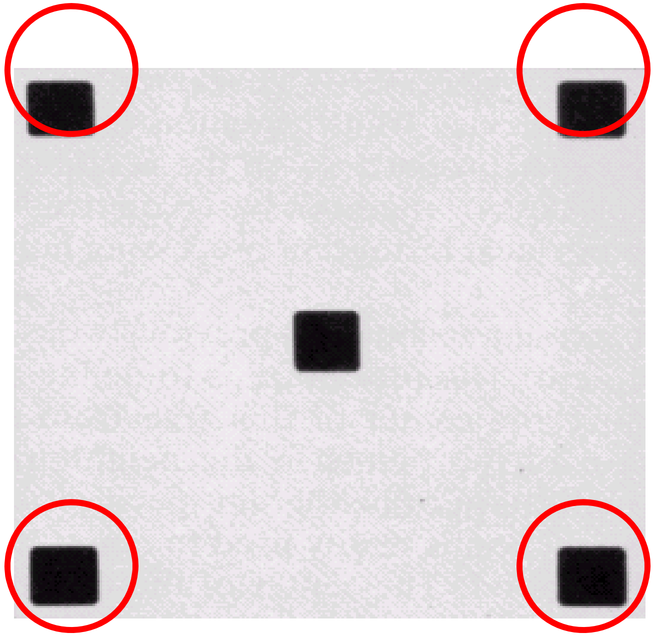


(a) Specified histogram. (b) Curve (1) is from Eq. (3.3-14), using the histogram in (a); curve (2) was obtained using the iterative procedure in Eq. (3.3-17). (c) Enhanced image using mappings from curve (2). (d) Histogram of (c).

a	**b**
c	**d**

- Histogram Processing
  - Histogram Equalization
  - Histogram Matching
  - Local Enhancement
  - Enhancement Using Histogram Statistics

Local Enhancement



Special request for local image…

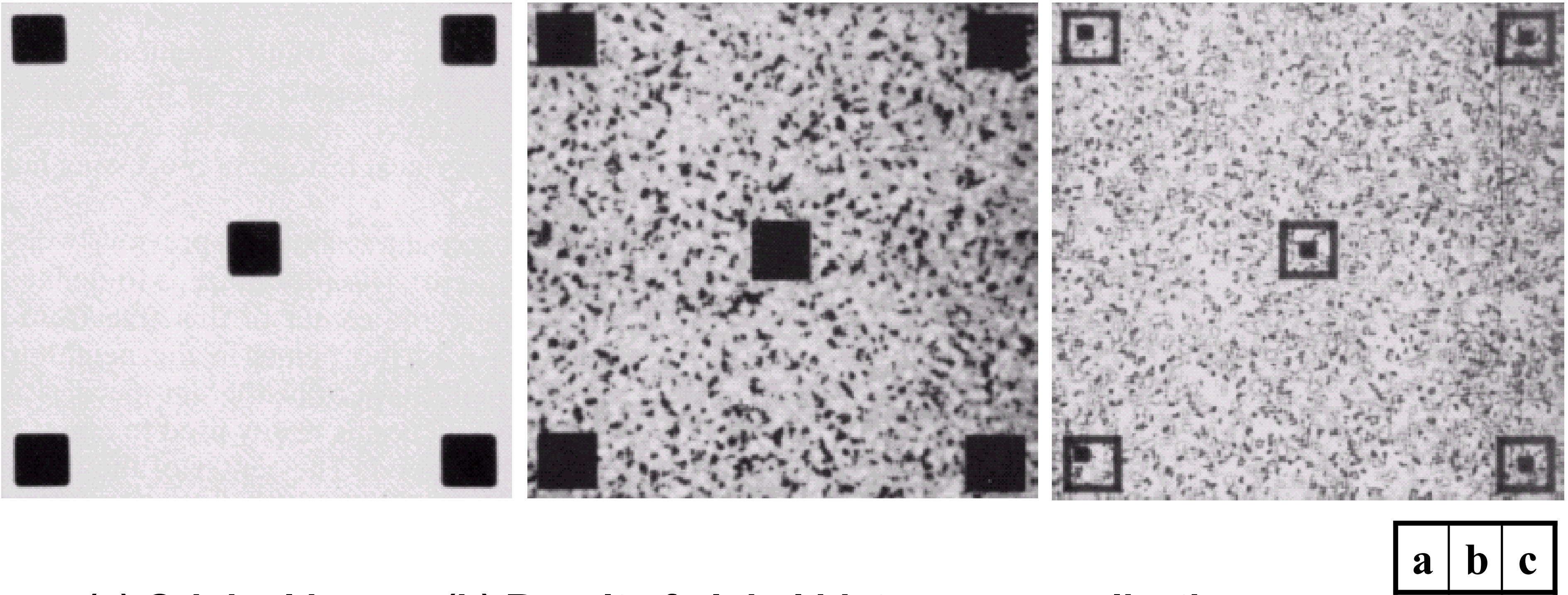
Local Enhancement

- **Previous methods were global.**
  - **When it is necessary to enhance details over smaller areas (local)?**
- **To devise transformation functions based on the gray-level distribution in the neighborhood of every pixel**

Procedure of Local Enhancement

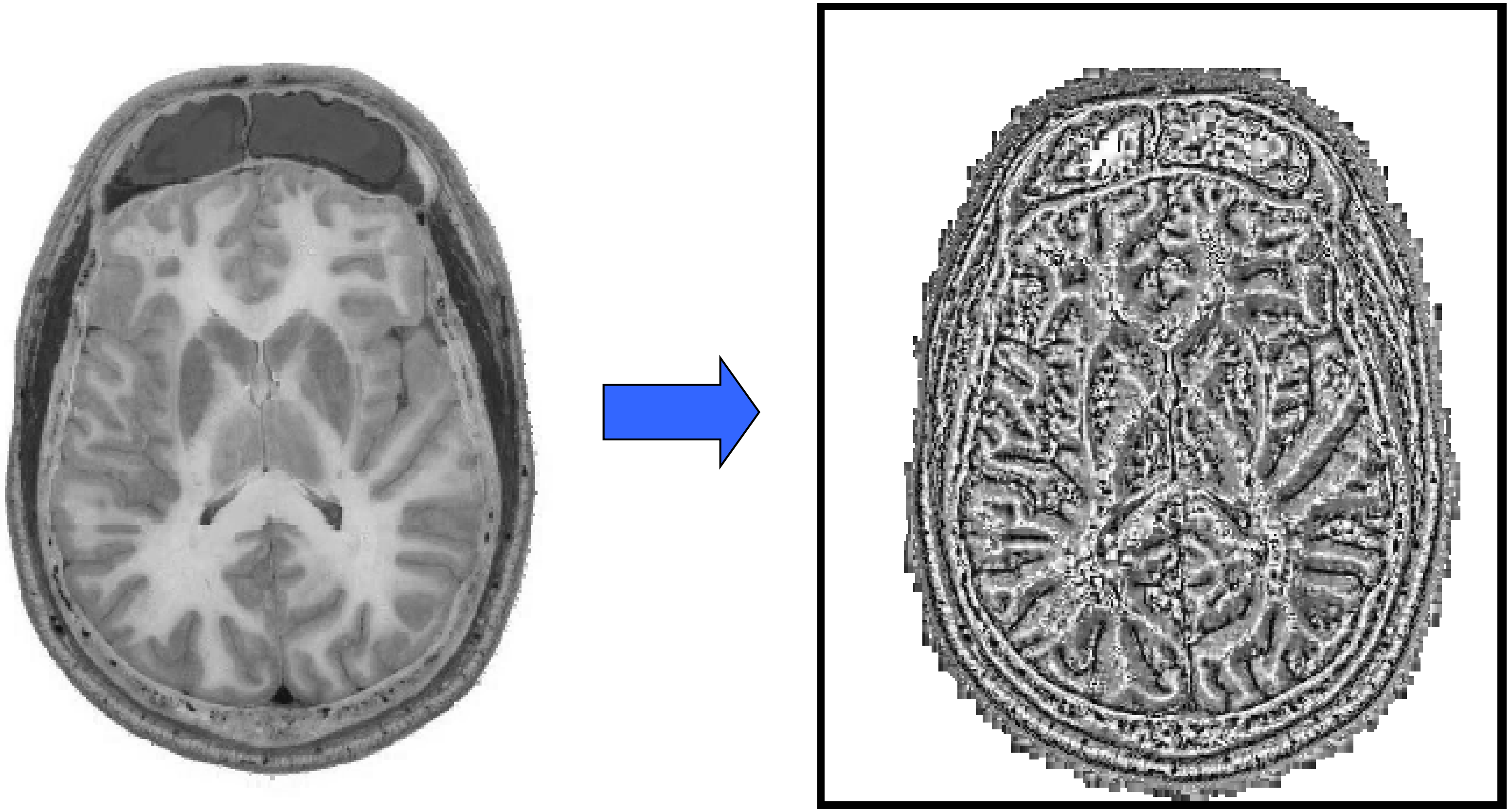
- Define a square (or rectangular) neighborhood and move the center of this area from pixel to pixel.
- For each pixel:
  - Calculate histogram of the points in the neighborhood
  - Obtain histogram equalization/specification function
  - Map gray level of pixel centered in neighborhood

Enhancement Using Local Histograms



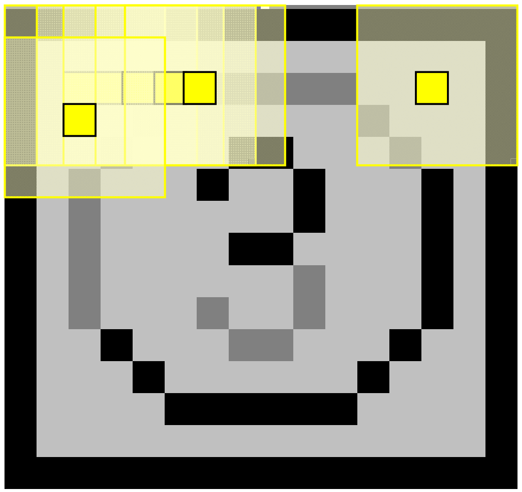
(a) Original image. (b) Result of global histogram equalization. (c) Result of local histogram equalization using a 7*7 neighborhood about each pixel.

Local Equalization

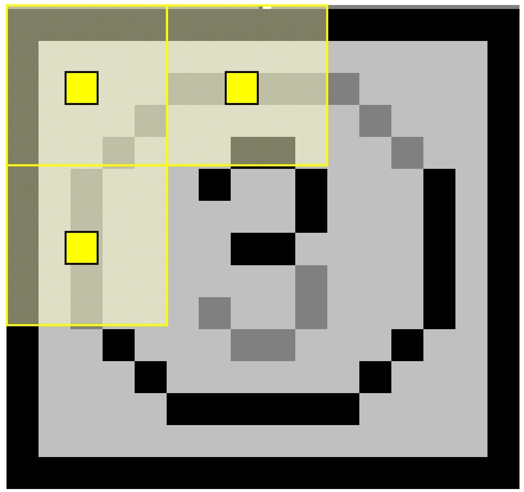


Local Enhancement

Pixel-to-pixel translation



Non-overlapping region



Reducing Computational Cost?

- **Additive construction of local histograms…**

- **Histogram Processing**
  - Histogram Equalization
  - Histogram Matching
  - Local Enhancement
  - **Enhancement Using Histogram Statistics**

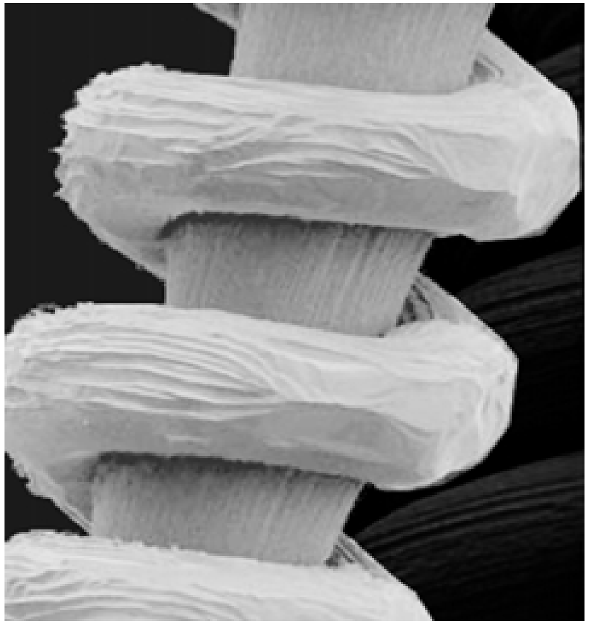
Enhancement Using Histogram Statistics

- Instead of using the image histogram directly for enhancement, we can use instead some statistical parameters obtainable directly from the histogram.
- Statistical parameters
  - Mean: measure of average gray level
  - Variance: measure of average contrast

Procedure

- Define a square (or rectangular) neighborhood and move the center of this area from pixel to pixel.
- For each pixel:
  - Calculate the mean (m_{Sxy}) and various (σ_{Sxy}) value of the gray-levels in subimage centered at (x, y)
    - If $m_{Sxy} \in [0, k_0 M_G]$ AND $\sigma_{Sxy} \in [k_1 D_G, k_2 D_G]$
    - Then $g(x, y) = E \bullet f(x, y)$;
    - Else $g(x, y) = f(x, y)$;
  - M_G: global mean
  - D_G: global standard deviation
  - E, k_0, k_1, k_2: parameters

Enhancement Based on Local Statistics

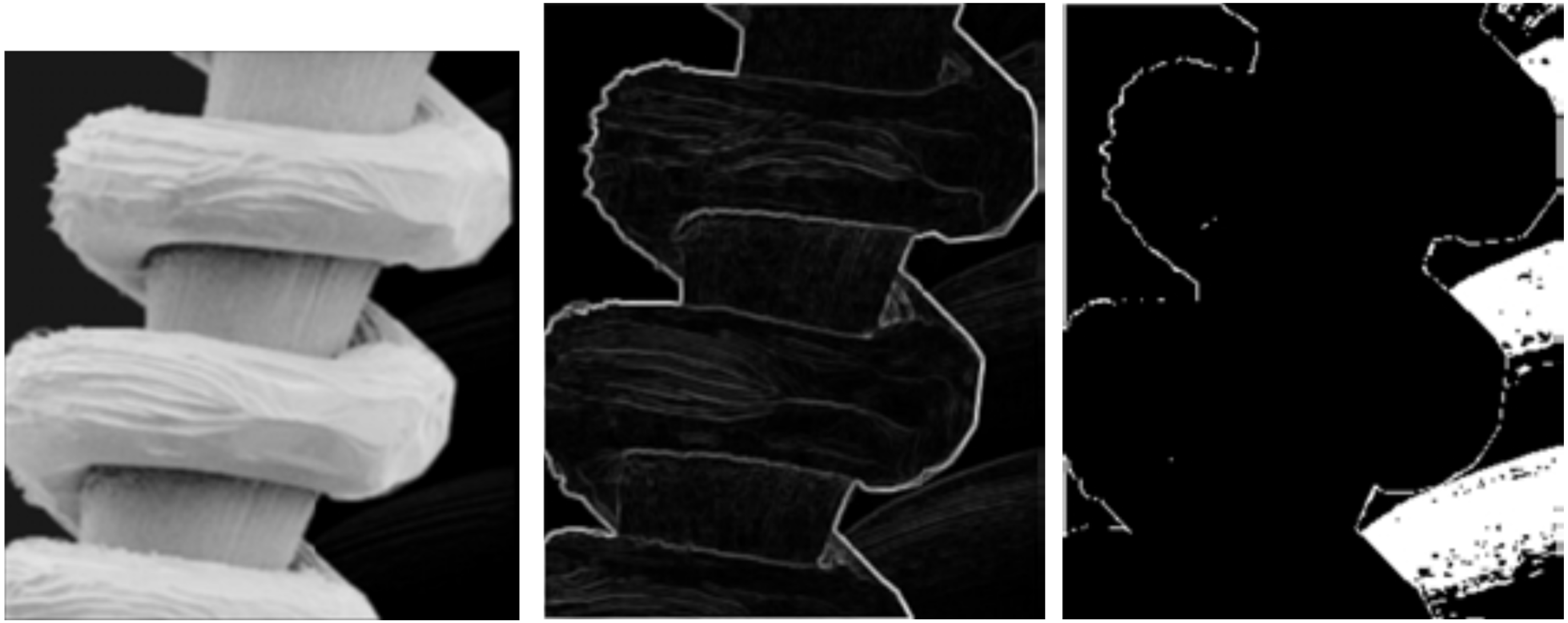


SEM image of a tungsten filament and support, magnified Approximately 130*(Original image courtesy of Mr. Michael Shaffer, Department of Geological Sciences, University of Oregon, Eugene).

Enhancement Based on Local Statistics

E = 4.0; k_0 = 0.4; k_1 = 0.02; k_2 = 0.4; (3×3) local region

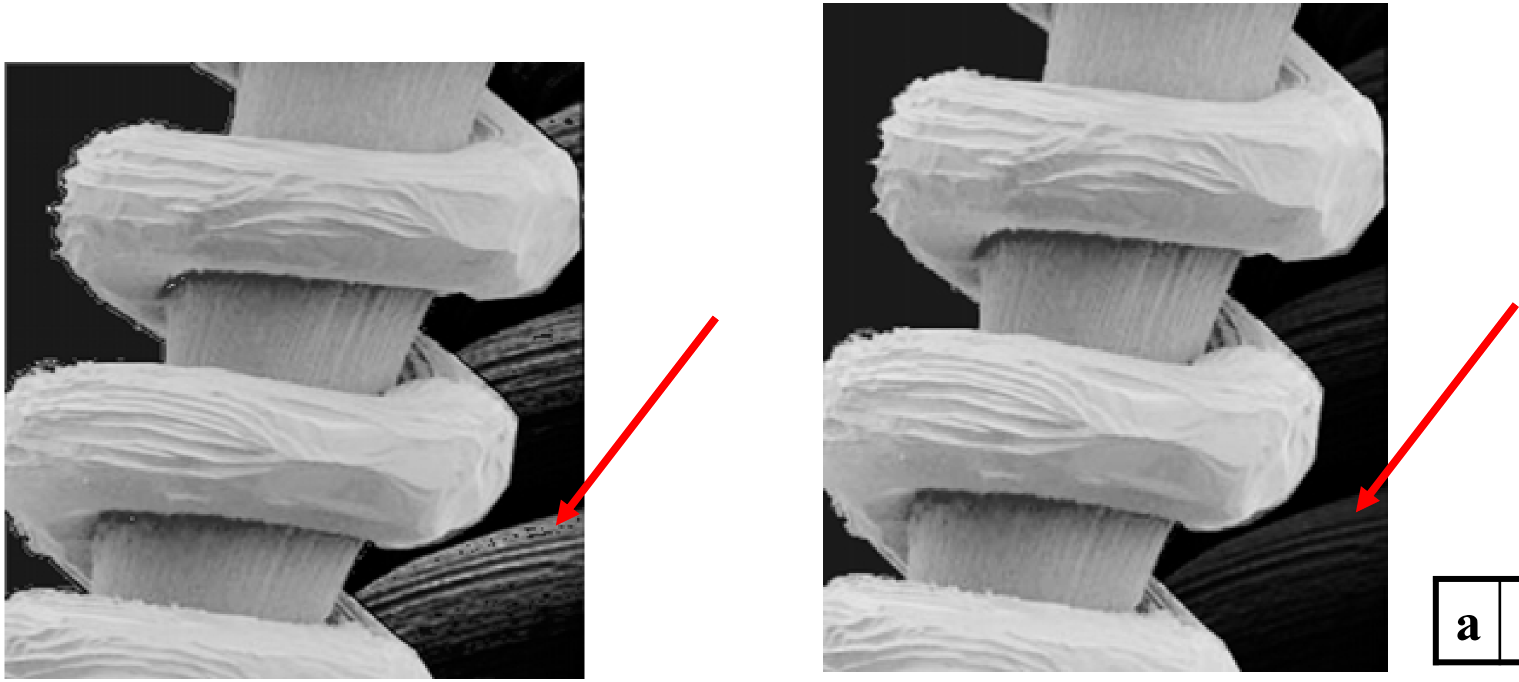
a	b	c



Local mean **Local std** **Multi. constant**

(a) Image formed from all local means obtained from Fig. 18 using Eq. (3.3-21). (b) Image formed from all local standard deviations obtained from Fig. 3.24 using Eq. (3.3-22). (c) Image formed from all multiplication constants used to produce the enhanced image shown in Fig. 20.

Enhancement Based on Local Statistics



a	b

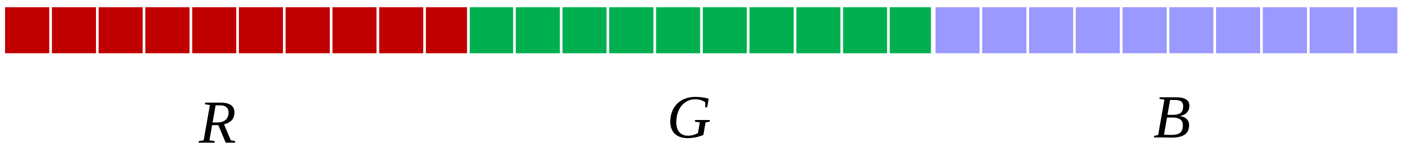
(a) Enhanced SEM image. (b) original image. Note in particular the enhanced area on the right, bottom side of the image.

Beyond Enhancement

- **Histogram as image/region descriptor**
  - **Image retrieval （图像检索）**
  - **Object tracking （目标跟踪）**
- **Histogram as probability distribution function (pdf)**
  - **Segmentation/matting （分割/抠图）**

Histogram of Color Images

- **By concatenation or using 3D table**



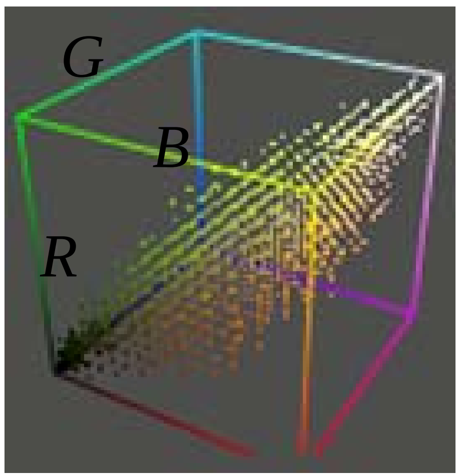
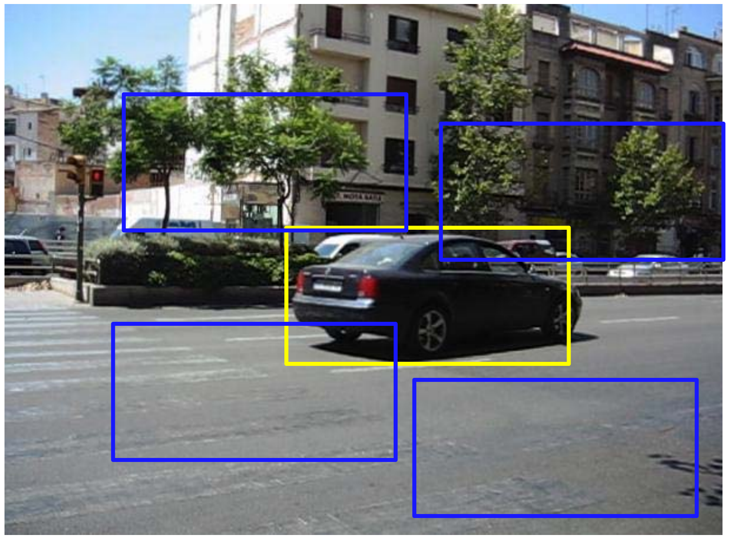


Image Retrieval by Histogram

- **Using the histogram of each image as index**

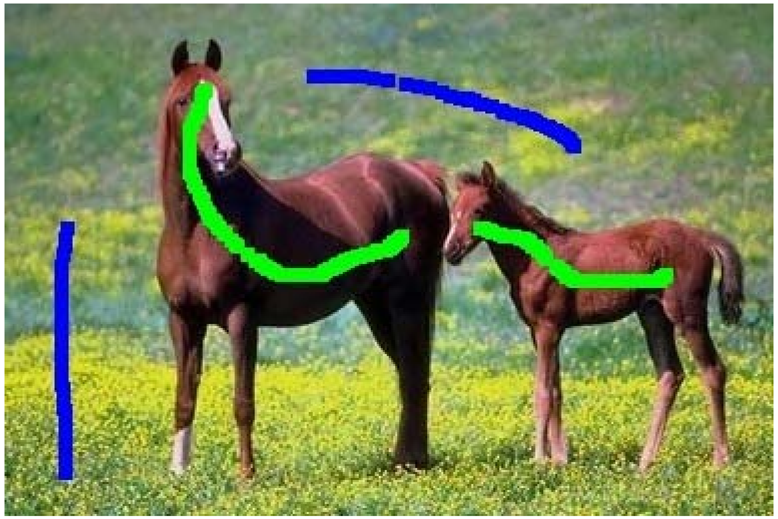
Object Tracking

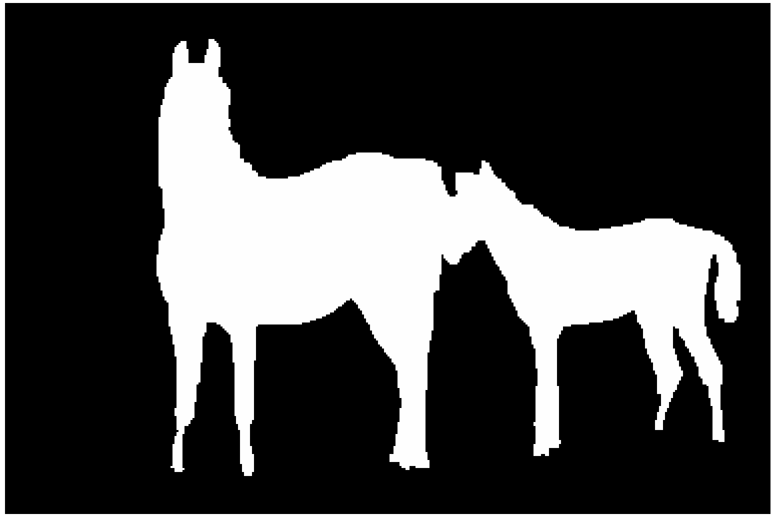
- **Search a region whose histogram best matched with the previous frame…**



Interactive Image Segmentation

- **Histogram to model color distribution of foreground and background**





图像的几何结构

像素邻域

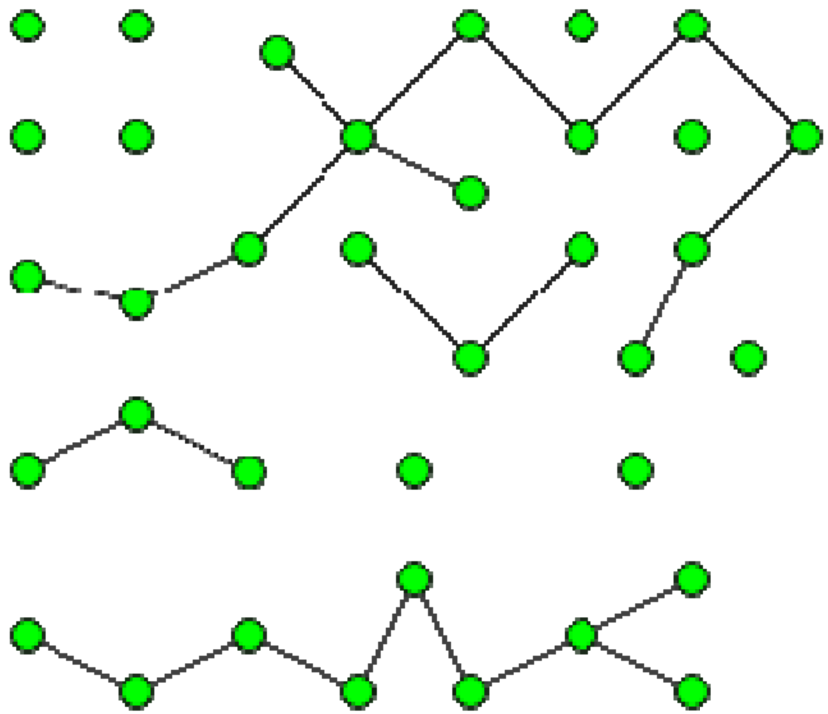
- …

图像 & 图

- 图像可以被看作是一张图，每个像素为一个结点，像素之间根据邻接关系和颜色值等以边相连：
  - 连通域
  - 最短路径/距离
  - 图切割
  - …

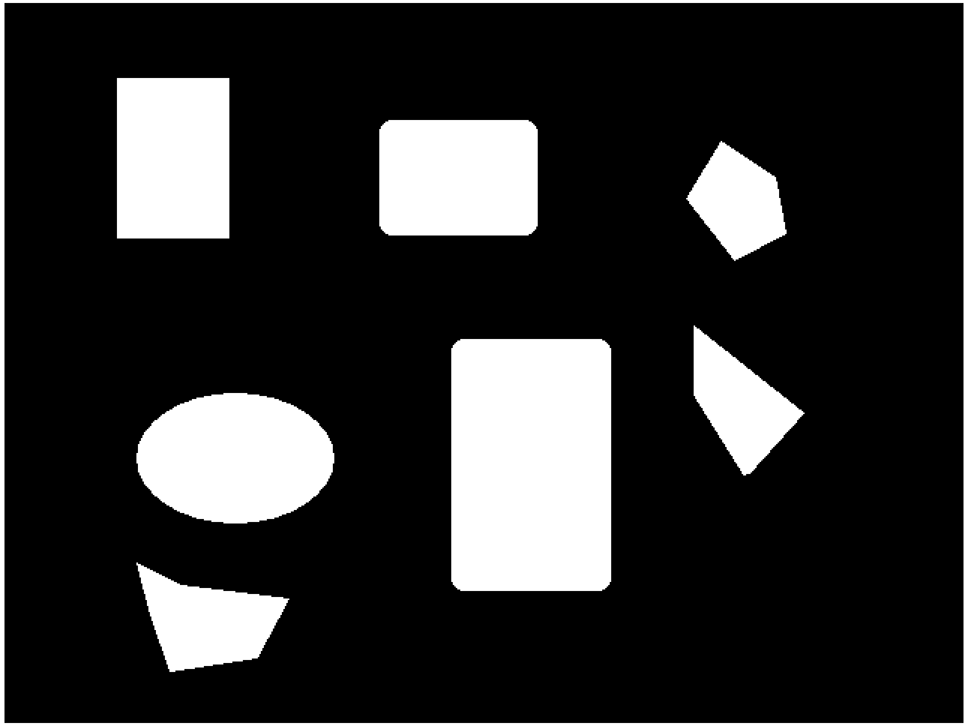
图的连通域

- 同一连通域的任意两结点都能以路径相连



图像连通域

- **D4/D8/Dm连通**
- 根据相邻像素的连通性，搜索连通区域



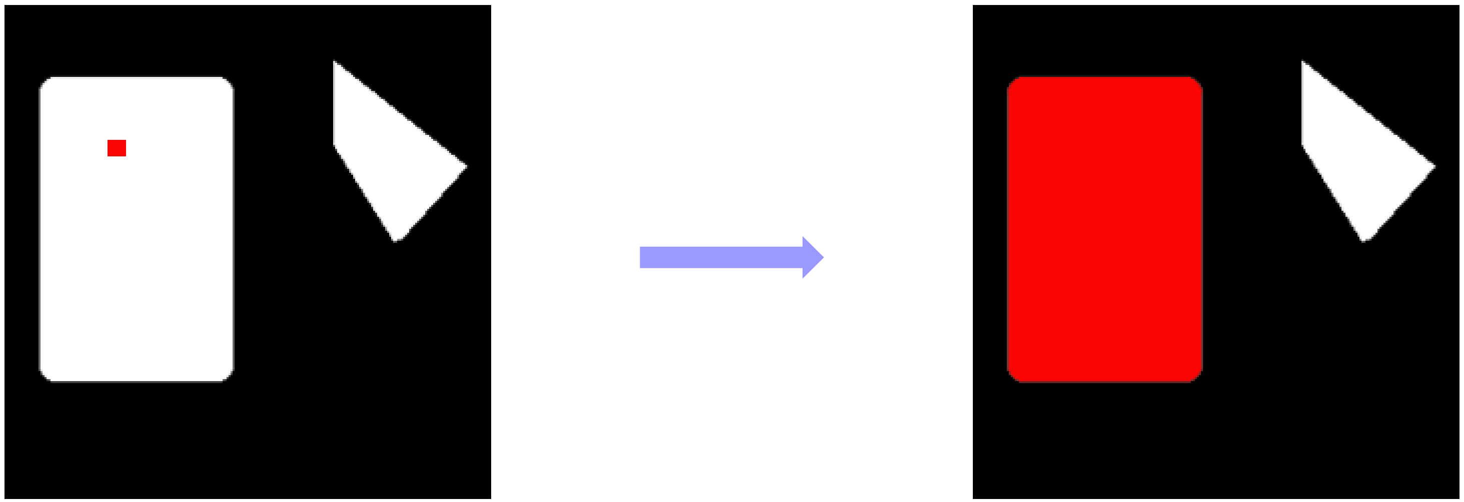
？ ？

方法？？

- ……

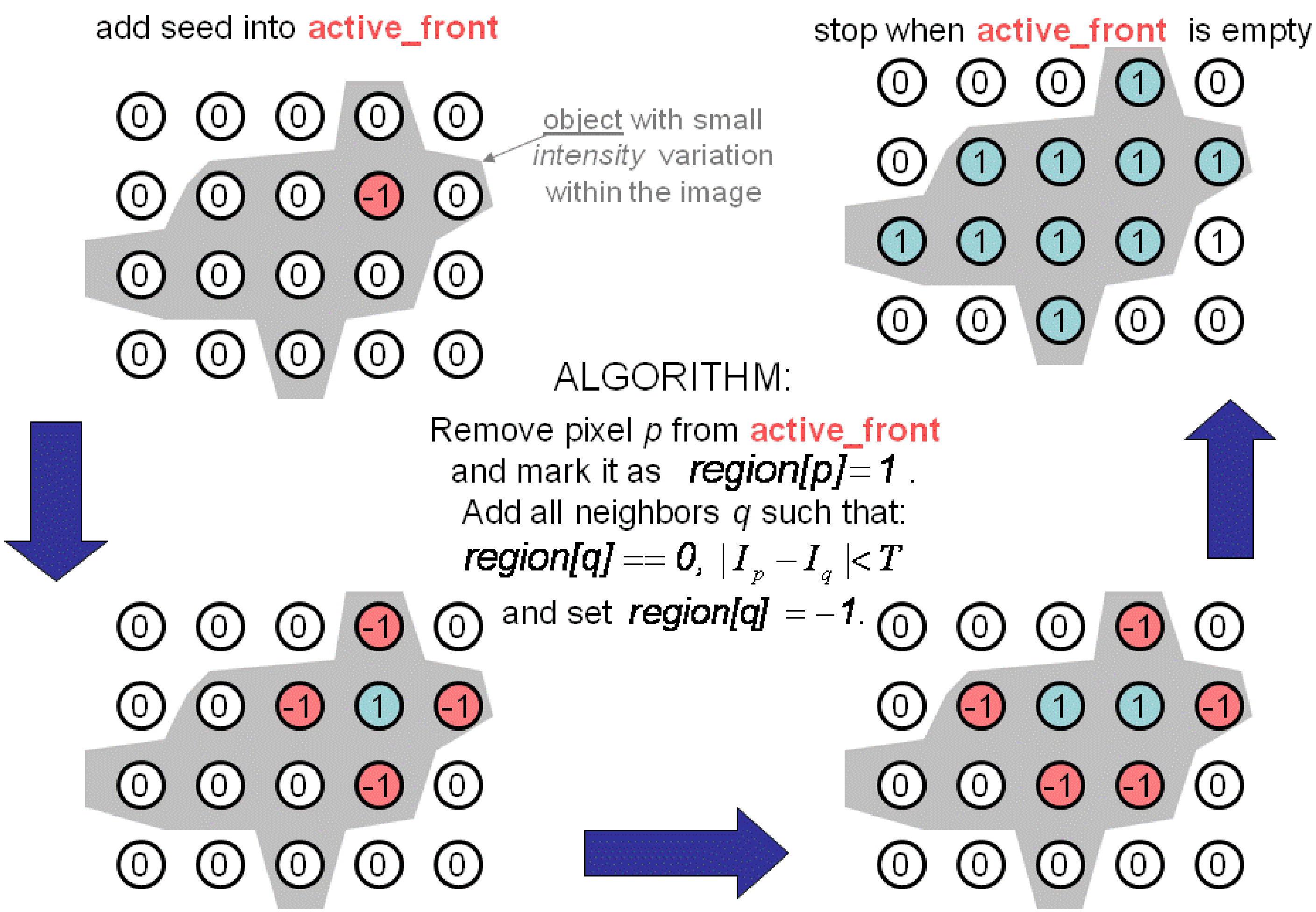
区域增长/种子填充

- 基于给定的种子像素，搜索最大连通区域



区域增长/种子填充

growRegion: red nodes are the "active_front" (queue or stack)



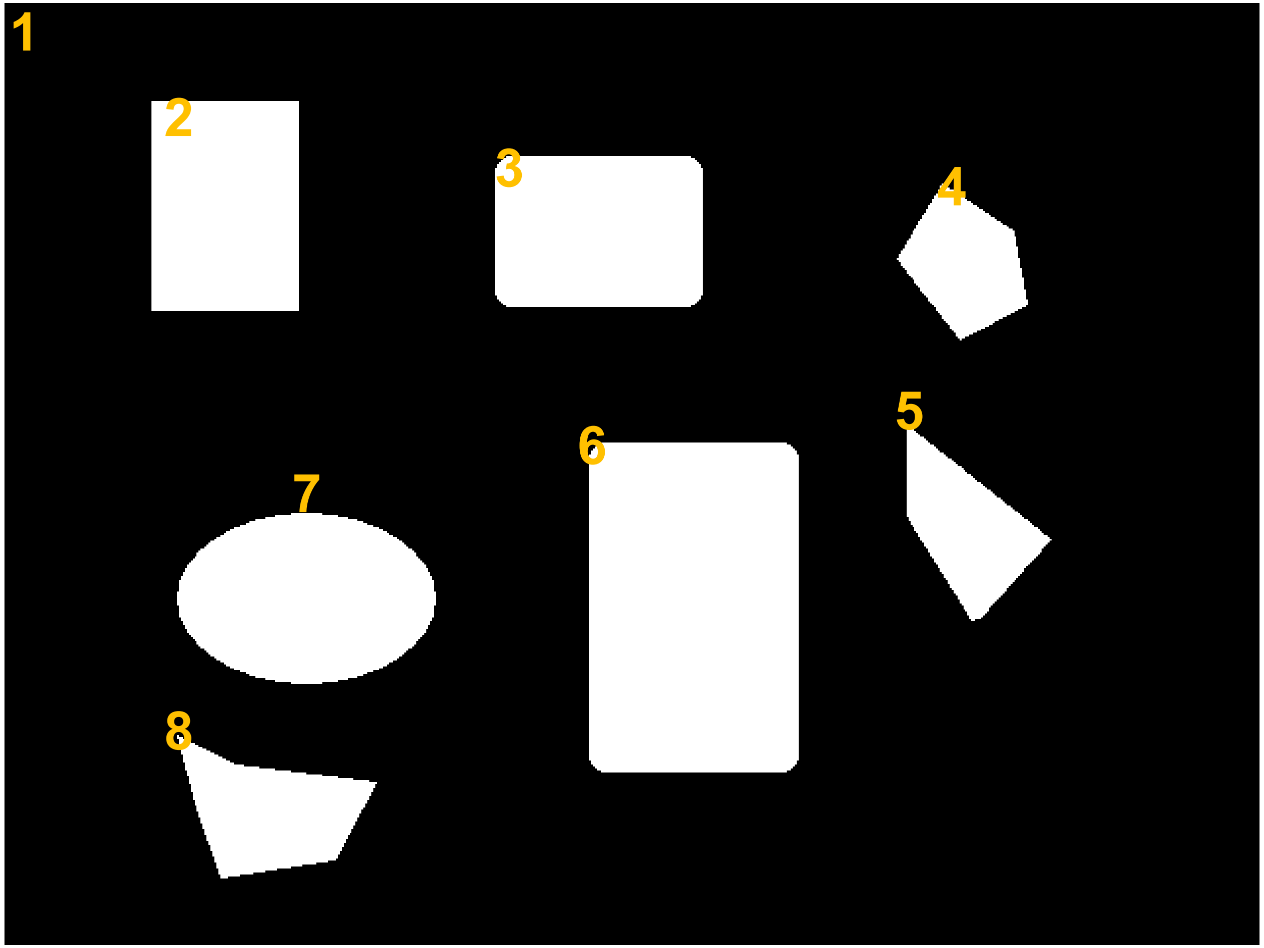
ALGORITHM:

Remove pixel p from active_front
and mark it as $region[p] = 1$.
Add all neighbors q such that:
$region[q] == 0, \; |I_p - I_q| < T$
and set $region[q] = -1$.

实现

- Push seeds into the stack
- While (stack is not empty)
  1. s = pop stack
  2. label s
  3. push unlabelled neighbors of s into the stack

图像连通域

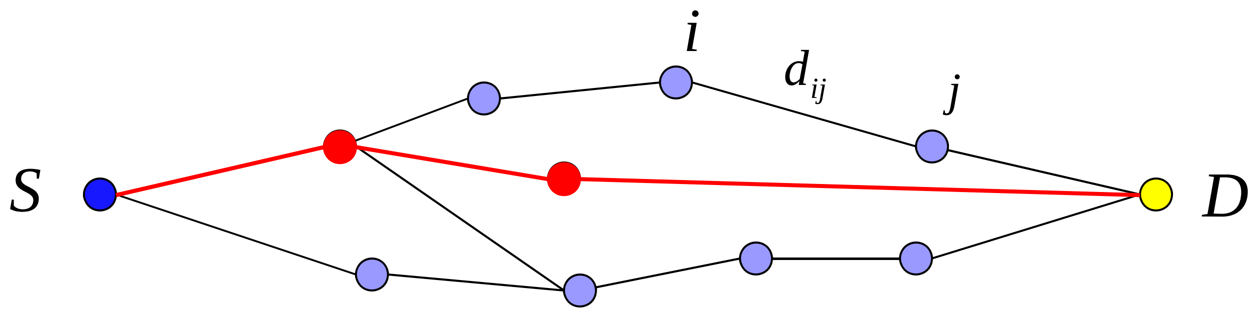


快速连通域算法

- 一次扫描 + 合并等价类

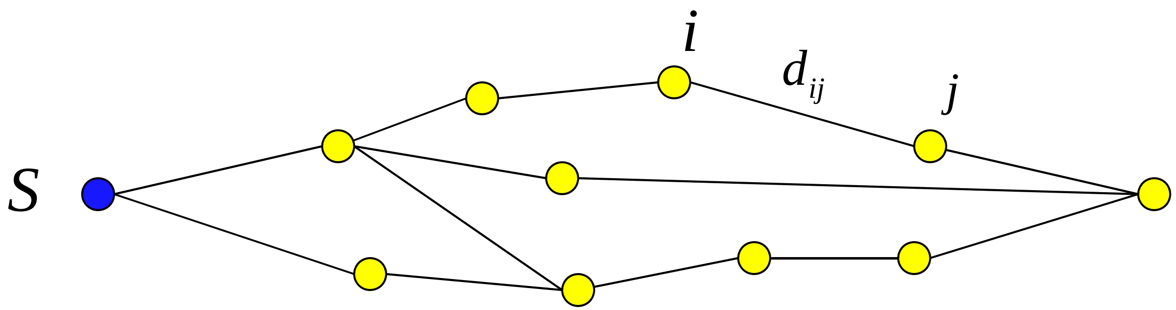
最短路径

- 两点之间



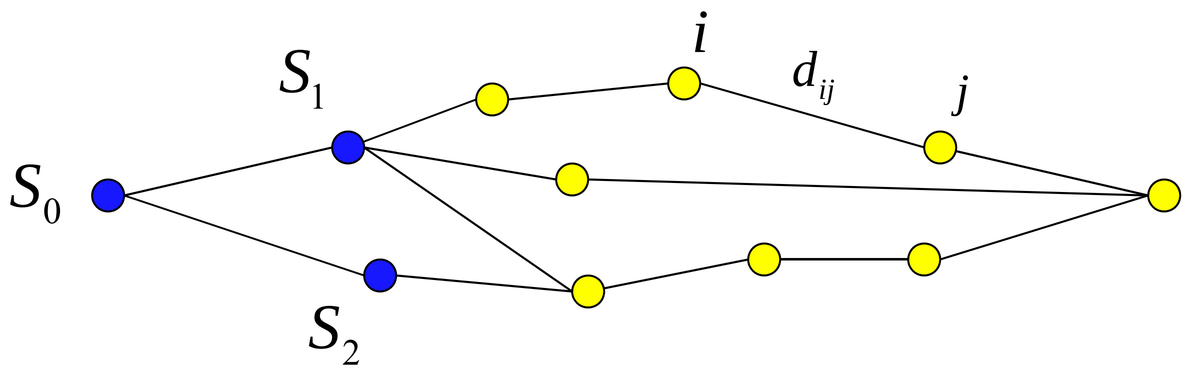
最短路径

- 单源最短路径： 所有结点到某个结点的最短路径 **(Dijkstra)**



最短路径

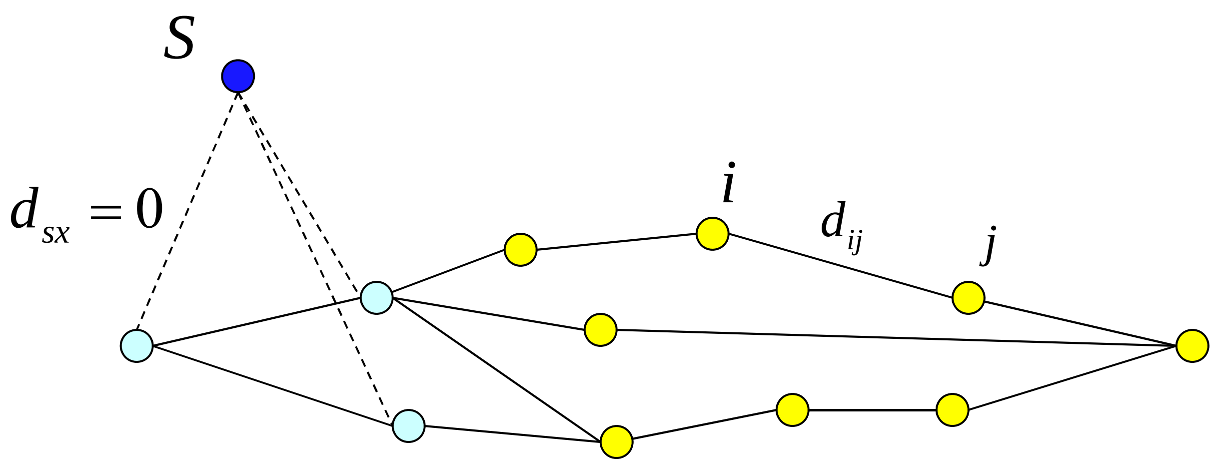
- 多源？？



求每个结点到 **S0 S1 S2** 的最短路径中最短的一条

最短路径

- 多源 =》 单源



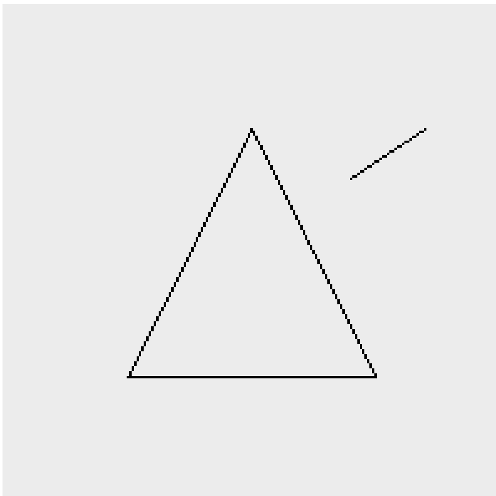
像素之间的最短路径

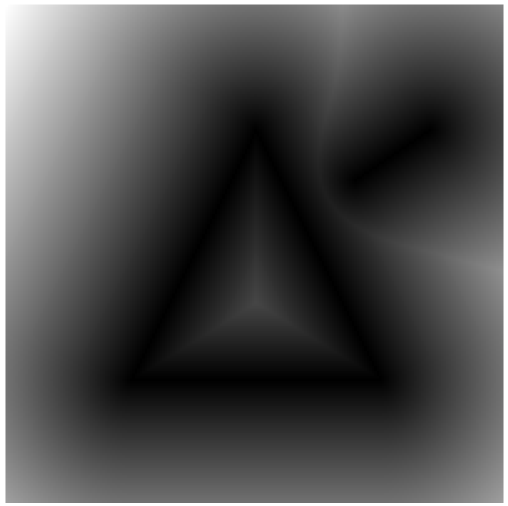
- 定义相邻像素之间的距离为其颜色差的函数，颜色差越大，距离越小：

$$d_{ij} \propto e^{-\beta \| I_i - I_j \|^2}$$

图像距离场

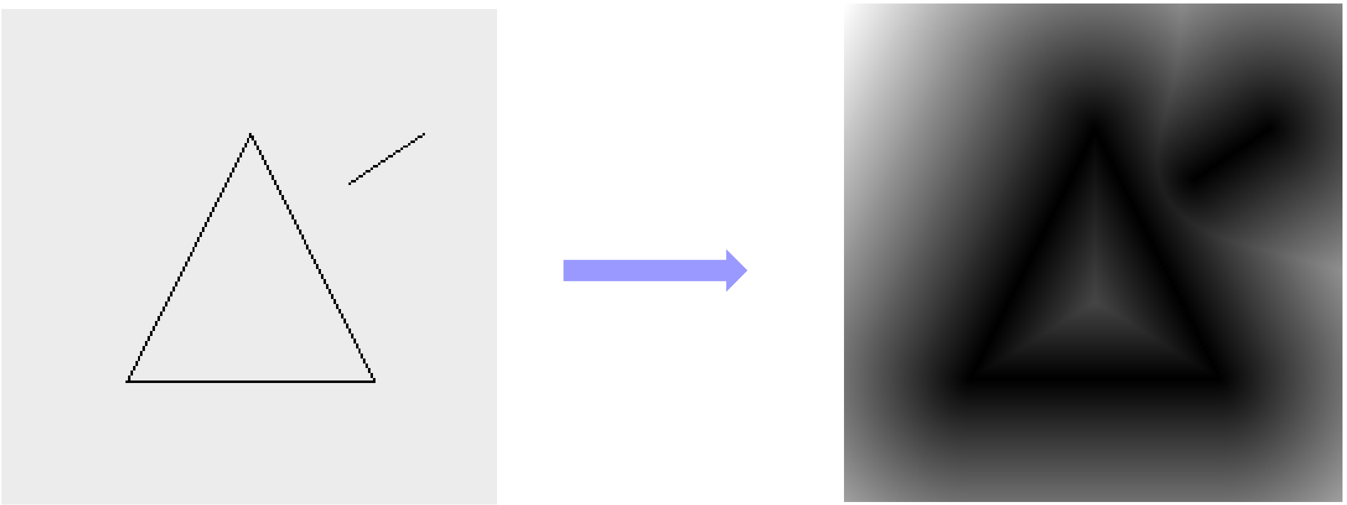
- 所有像素到种子像素的最短距离：





距离变换

- 利用像素规则布局的特点，对图像求距离场的快速方法



距离变换

(0,0)

0	1	2
1	1.4	2.1969
2	2.1969	2.8

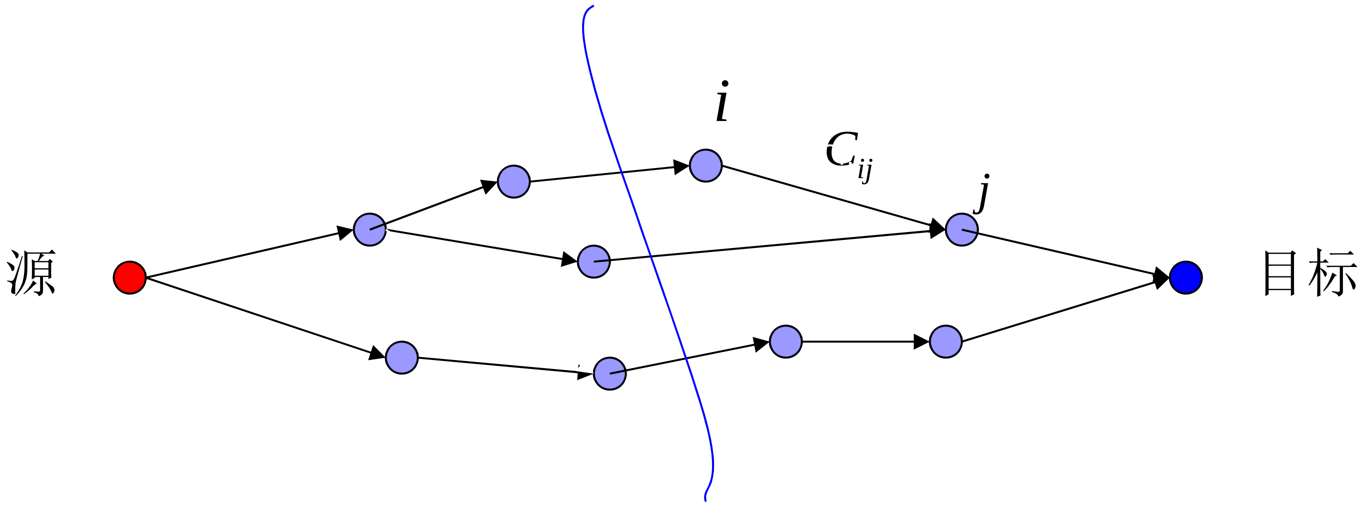
(2,2)

3*3 欧氏距离模板

距离变换

3,5969	2,8	2,1969	1,4	1	1,4 1,4142	2,1969 2,2361	2,8 2,8284	3,5969 3,6051	4,3938 4,4721
3,1969	2,1969	1,4	1	0	1	1,4 1,4142	2,1969 2,2361	3,1969 3,1623	4,1969 4,1231
3	2	1	0	0	0	1	2	3	4
3	2	1	0	0	0	1	2	3	4
3,1969	2,1969	1,4	1	0	1	1,4	2,1969	3,1969	4,1969
3,5969	2,8	2	1	0	1	2	2,8	3,5969	4,3938
4,1969	3,1969	2,1969	1,4	1	1,4	2,1969	3,1969	4,1969	4,9969

图切割/最小割



边的容量 ：每一条边允许通过的最大流量（c_{ij}）；

图的切割 ：能将源和目标之间所有路径切断的图的剖分；

最小割 ：所有图的切割中所切断的边的容量之和最小的一个，对应于网络流的瓶劲，即最大流；

图像切割（分割）