Lecture 20-Thresholding (chapter 10.3)

Yuyao Zhang PhD
zhangyy8@shanghaitech.edu.cn
SIST Building 2 302-F

上海科技大学
ShanghaiTech University

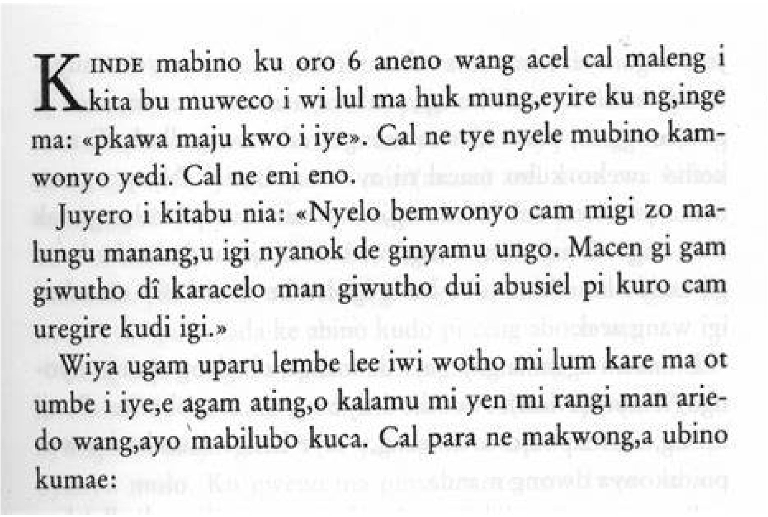
What to do?

- **Definition**

$$g(x,y)=\begin{cases}1, & f(x,y)>T \quad \text{(object points)}\\ 0, & f(x,y)\le T \quad \text{(background points)}\end{cases}$$

```
foreground = rgb2gray(im)<150;
background = rgb2gray(im)>150;
```

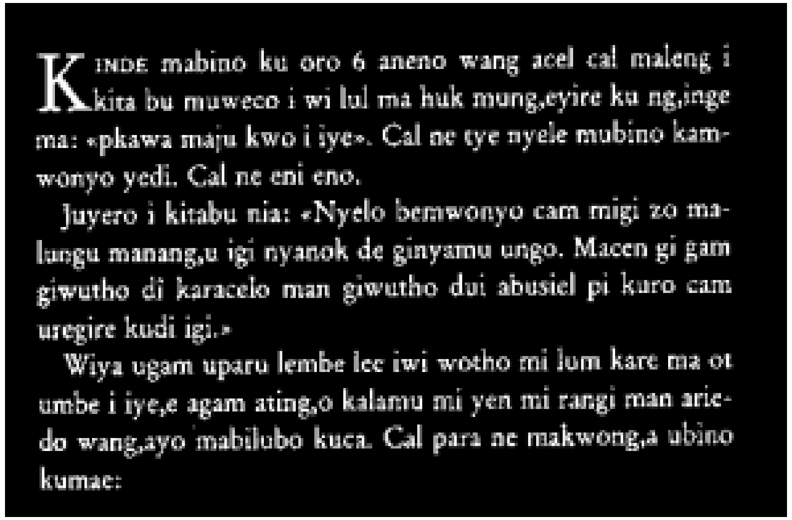
Input image


KINDE mabino ku oro 6 aneno wang acel cal maleng i kita bu muweco i wi lul ma huk mung,eyire ku ng,inge ma: «pkawa maju kwo i iye». Cal ne tye nyele mubino kamwonyo yedi. Cal ne eni eno.

Juyero i kitabu nia: «Nyelo bemwonyo cam migi zo malungu manang,u igi nyanok de ginyamu ungo. Macen gi gam giwutho dî karacelo man giwutho dui abusiel pi kuro cam uregire kudi igi.»

Wiya ugam uparu lembe lee iwi wotho mi lum kare ma ot umbe i iye,e agam ating,o kalamu mi yen mi rangi man ariedo wang,ayo mabilubo kuca. Cal para ne makwong,a ubino kumae:

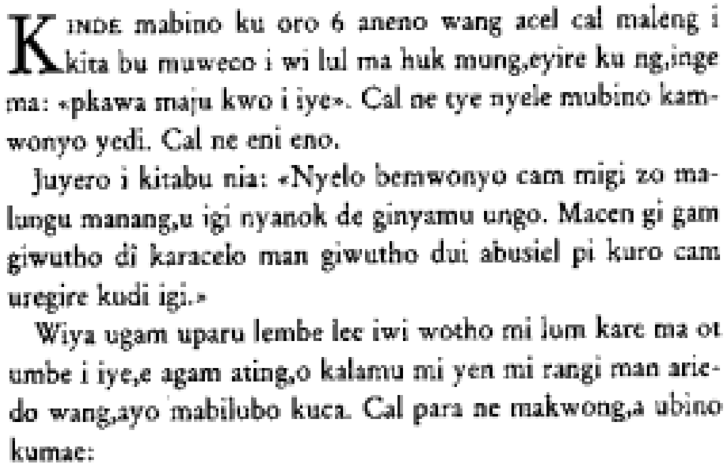
foreground


KINDE mabino ku oro 6 aneno wang acel cal maleng i kita bu muweco i wi lul ma huk mung,eyire ku ng,inge ma: «pkawa maju kwo i iye». Cal ne tye nyele mubino kamwonyo yedi. Cal ne eni eno.

Juyero i kitabu nia: «Nyelo bemwonyo cam migi zo malungu manang,u igi nyanok de ginyamu ungo. Macen gi gam giwutho dî karacelo man giwutho dui abusiel pi kuro cam uregire kudi igi.»

Wiya ugam uparu lembe lee iwi wotho mi lum kare ma ot umbe i iye,e agam ating,o kalamu mi yen mi rangi man ariedo wang,ayo mabilubo kuca. Cal para ne makwong,a ubino kumae:

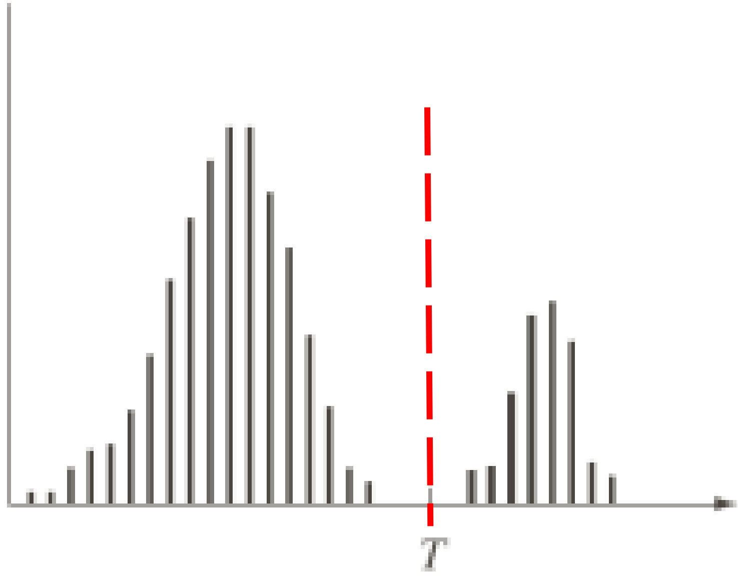
background


KINDE mabino ku oro 6 aneno wang acel cal maleng i kita bu muweco i wi lul ma huk mung,eyire ku ng,inge ma: «pkawa maju kwo i iye». Cal ne tye nyele mubino kamwonyo yedi. Cal ne eni eno.

Juyero i kitabu nia: «Nyelo bemwonyo cam migi zo malungu manang,u igi nyanok de ginyamu ungo. Macen gi gam giwutho dî karacelo man giwutho dui abusiel pi kuro cam uregire kudi igi.»

Wiya ugam uparu lembe lee iwi wotho mi lum kare ma ot umbe i iye,e agam ating,o kalamu mi yen mi rangi man ariedo wang,ayo mabilubo kuca. Cal para ne makwong,a ubino kumae:

Intensity Valley

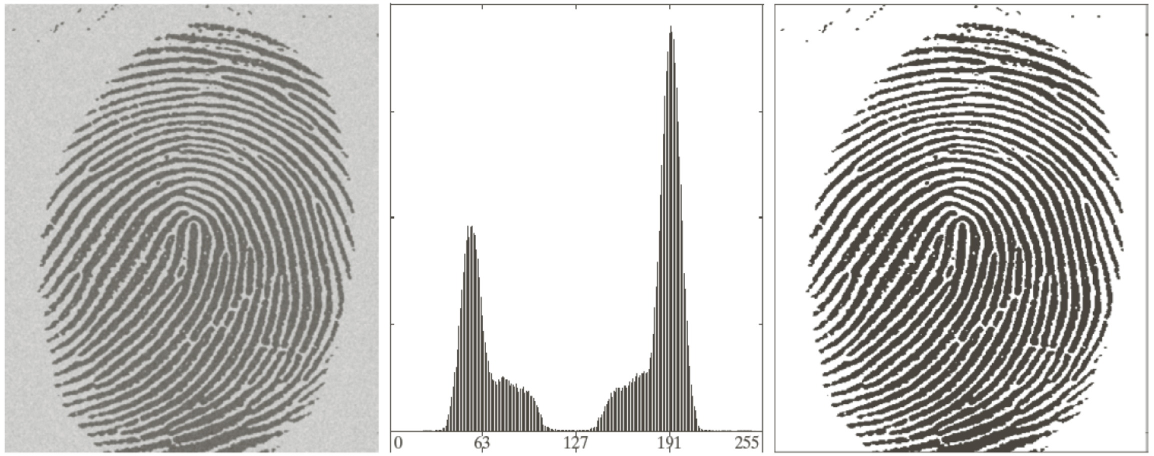


$$g(x,y) = \begin{cases} 1, & f(x,y) > T \quad \text{(object points)} \\ 0, & f(x,y) \leq T \quad \text{(background points)} \end{cases}$$

Basic Global Thresholding

- **Steps:**

1. Select an initial estimate of the global threshold **T**;
2. Segment the image using **T** to two groups $G_1(>T)$ and $G_2(\leq T)$;
3. Compute average intensity m_1 and m_2 for G_1 and G_2 respectively;
4. Compute new threshold $\mathbf{T}=(m_1 + m_2)/2$;
5. Repeat 2-4 until the difference between **T** in successive iteration is smaller than requirement.



上海科技大学
ShanghaiTech University

Otsu's method

- Maximize the between-class variance.
- A good threshold should separate pixels into tight cluster.

[1] Otsu N. A threshold selection method from gray-level histogram. IEEE Trans,1979;SMC-9;62-66

Otsu's method

Image PMF:

p_i = probability that $I(x,y) = i, i = 1,2,\cdots,L-1$

$m_G = \sum_{i=1}^{L-1} i * p_i$

$\sigma_G = \sum_{i=1}^{L-1} (i - m_G)^2 * p_i$

where

m_G: average intensity of entire image (global mean).

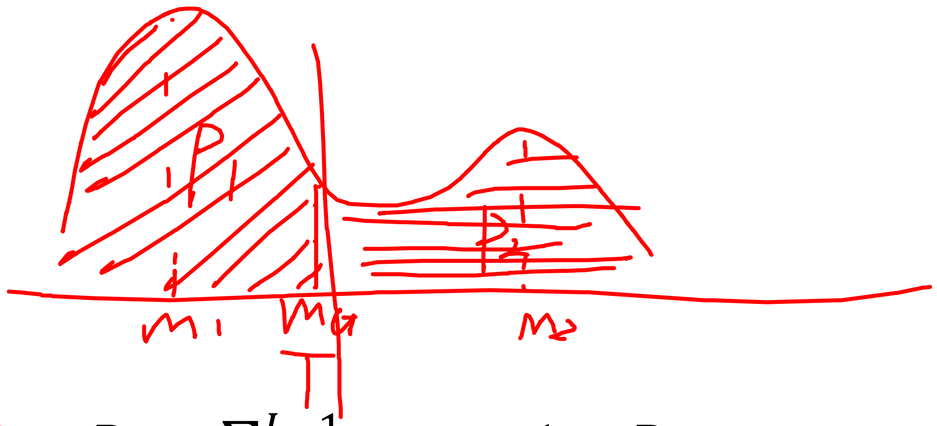
σ_G: global variance.

Otsu's method

Suppose we select a threshold T.

$$C_1 = \{(x,y) | I(x,y) < T\}$$

$$C_2 = \{(x,y) | I(x,y) > T\}$$



Then

$$P_1 = \sum_{i=1}^{T} p_i \qquad P_2 = \sum_{i=T+1}^{L-1} p_i = 1 - P_1$$

Class conditional

$$m_1 = \sum_{i=1}^{T} i * p_i \qquad m_2 = \sum_{i=T+1}^{L-1} i * p_i$$

Between-class variance is defined as:

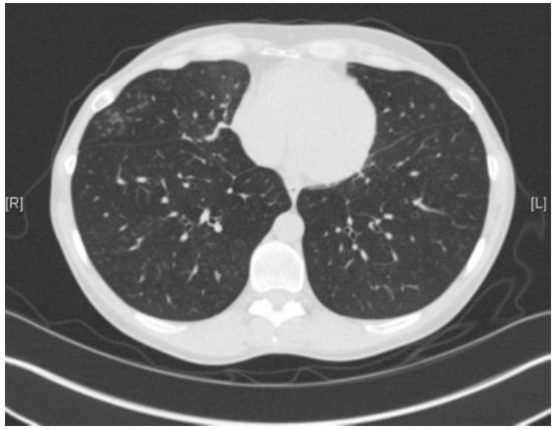
$$\sigma_B = P_1(m_1 - m_G)^2 + P_2(m_2 - m_G)^2$$

In practice, we just consider all possible T, and choose the T that maximizes σ_B

0-255

Try this

Input image



Output binary map

Fig 3:Histogram with T labeled

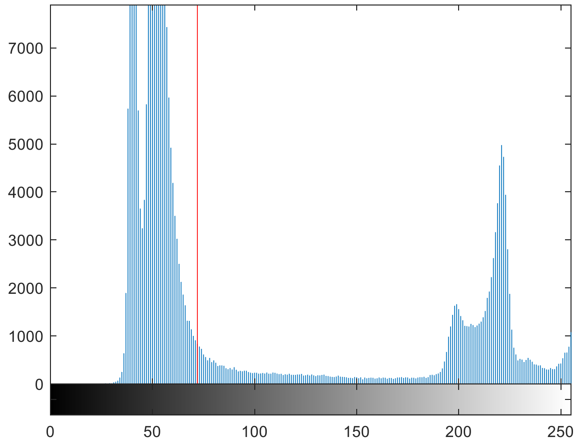
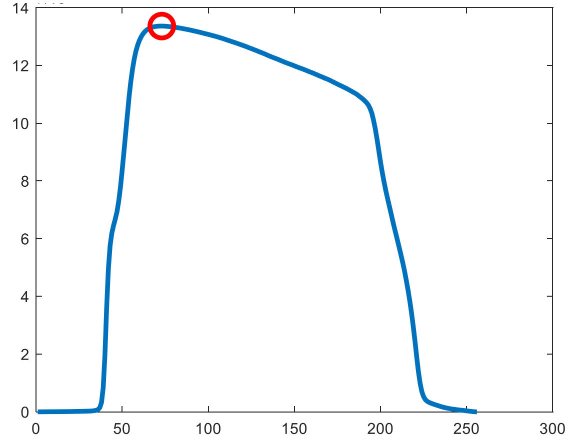
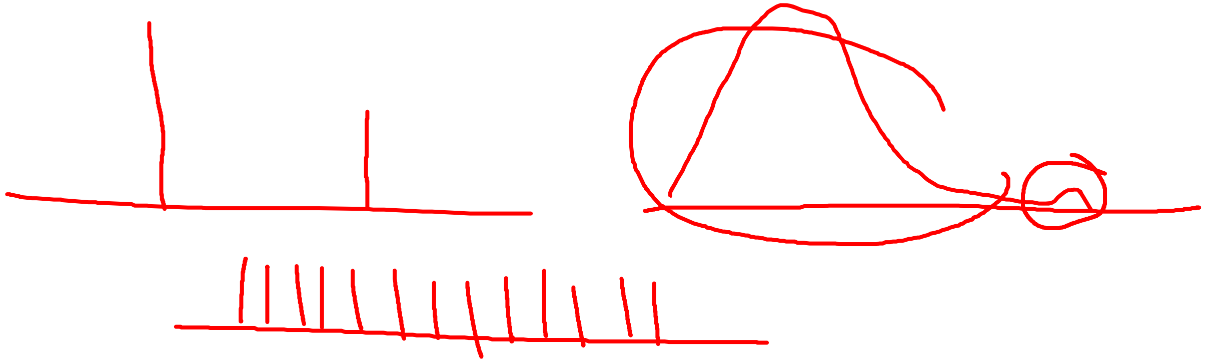


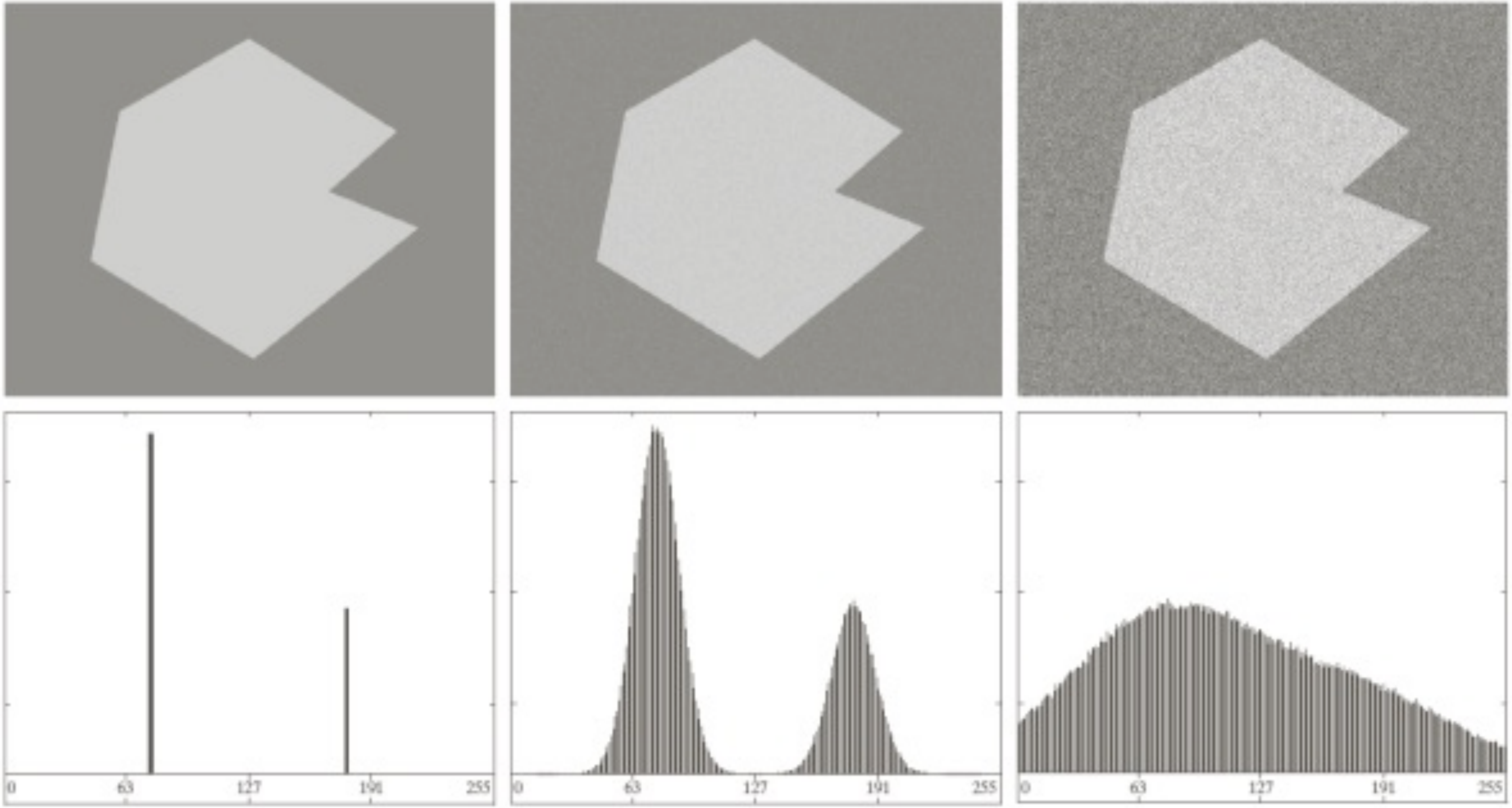
Fig 4:Between-class variance



- Find the matlab function [level,EM] = graythresh(I) . Read it and try to modify it to output fig3 and fig4.
- Otsu can fail when:
  - -no strong peaks in the histogram
  - -object is small (with respect to) background

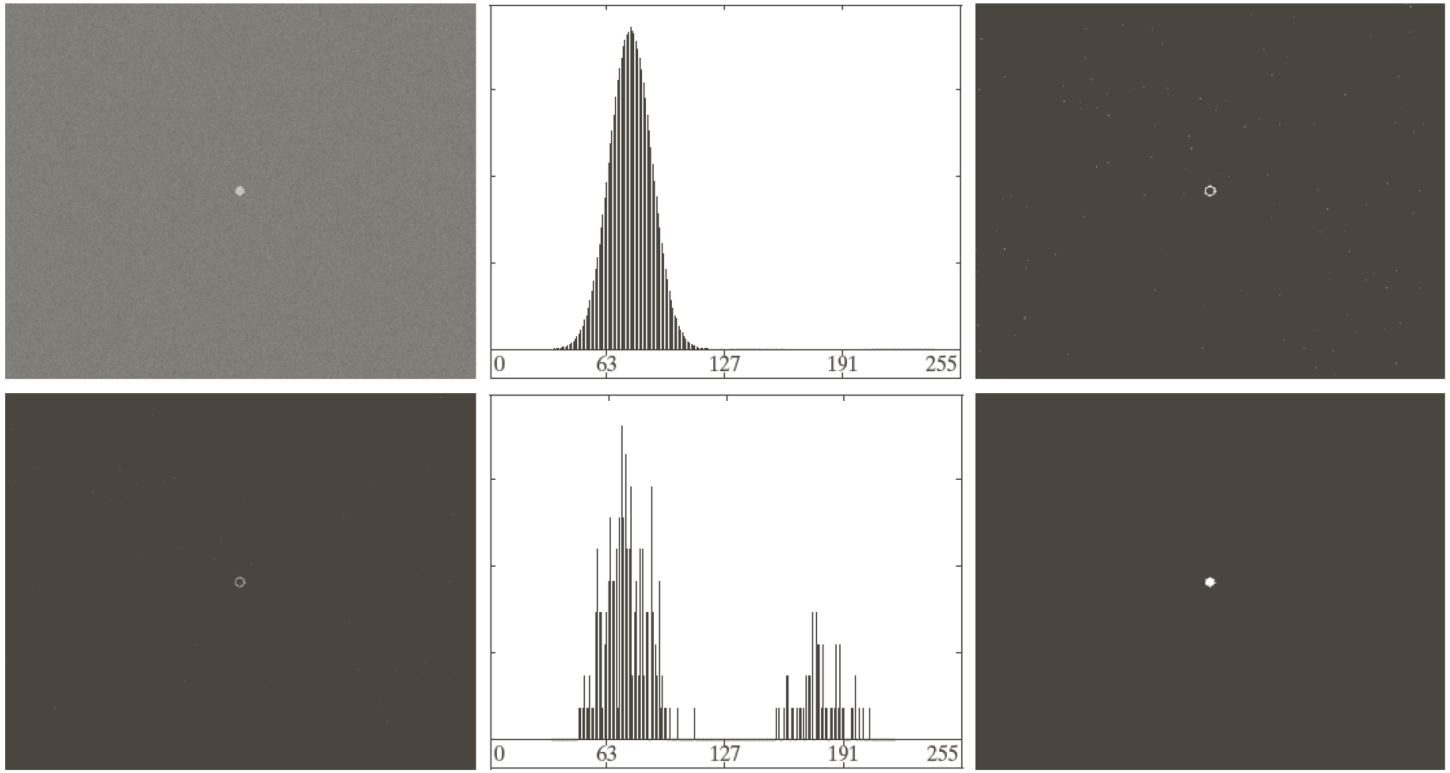


Influence of Noise



Improve Global Thresholding

- **Using edges:**



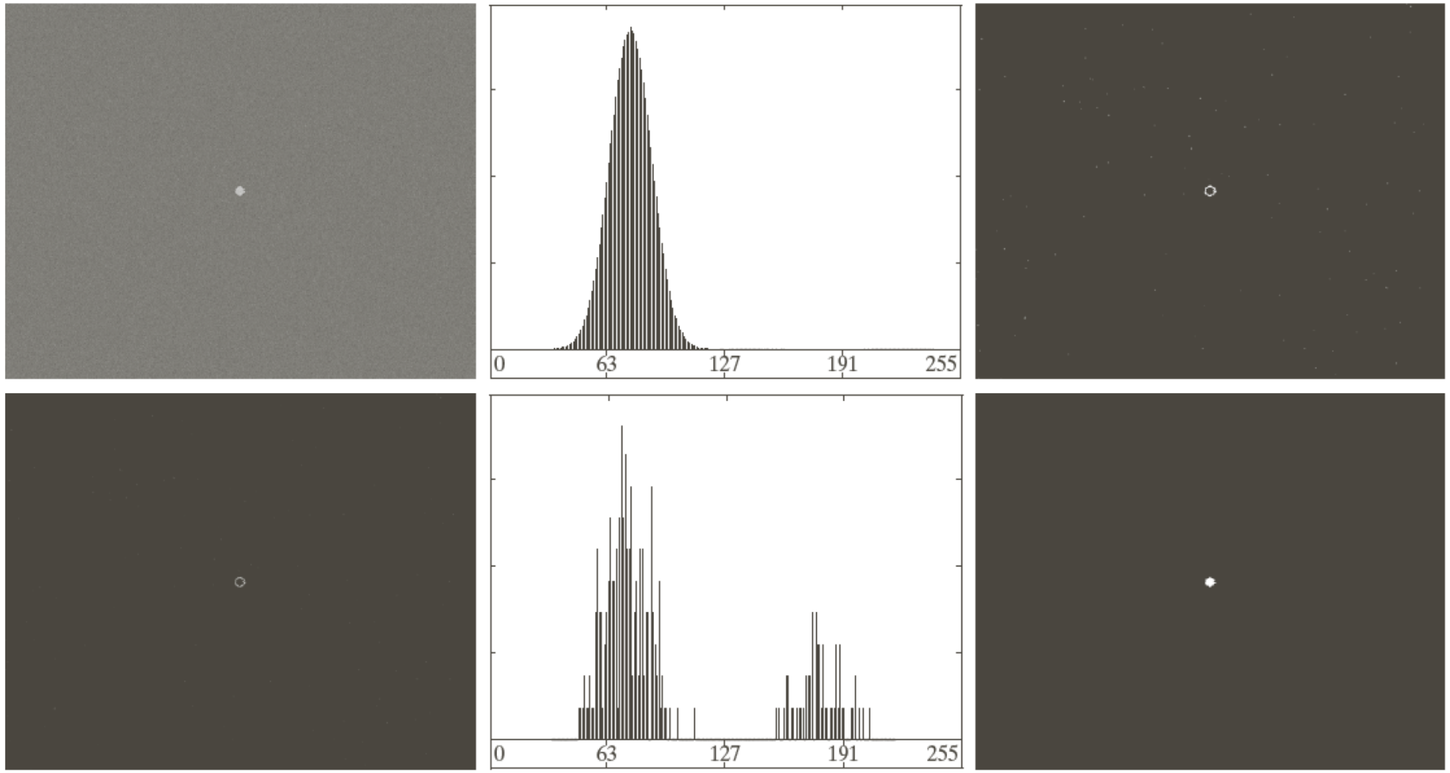
Improve Global Thresholding

- **Using edges:**

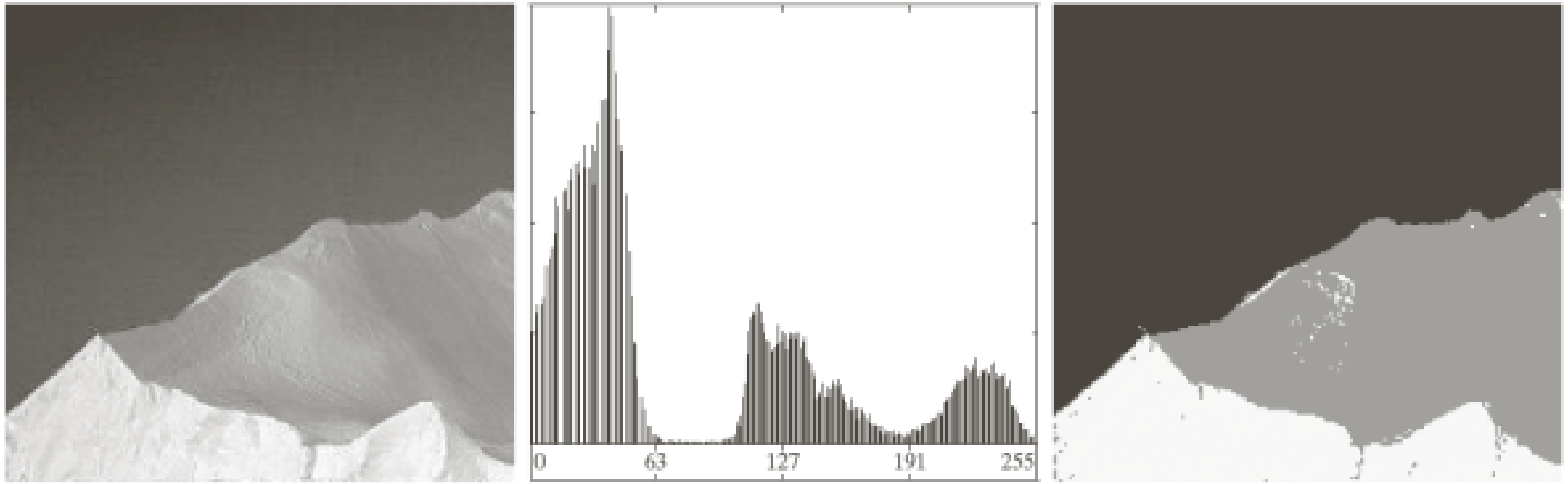
  1. compute an edge image from the input image $f(x, y)$ using any edge detector;
  2. specify a threshold value T;
  3. Threshold the edge image using T to produce a binary image $g_T(x, y)$
  4. compute a histogram using only the pixels in $f(x, y)$ that correspond to the locations of the 1-valued pixels in $g_T(x, y)$
  5. use the histogram to segment $f(x, y)$;

Improve Global Thresholding

➢ **Using edges:**



Multiple thresholds



Multiple thresholds

- **Between-class variance (类间方差):**

$$\sigma_B^2 = P_1(m_1 - m_G)^2 + P_2(m_2 - m_G)^2 + P_3(m_3 - m_G)^2$$

Where

$$P_1 = \sum_{i=0}^{k_1} p_i \quad P_2 = \sum_{i=k_1+1}^{k_2} p_i \quad P_3 = \sum_{i=k_2+1}^{L-1} p_i$$

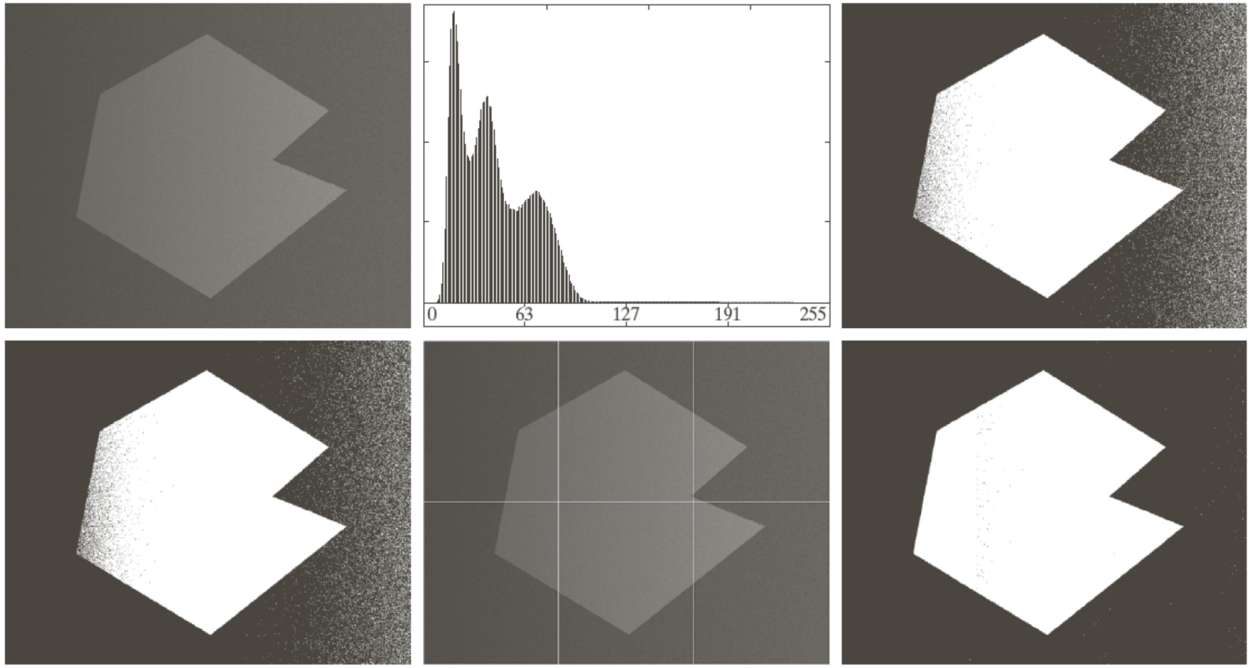
$$m_1 = \sum_{i=0}^{k_1} ip_i \quad m_2 = \sum_{i=k_1+1}^{k_2} ip_i \quad m_3 = \sum_{i=k_2+1}^{L-1} ip_i$$

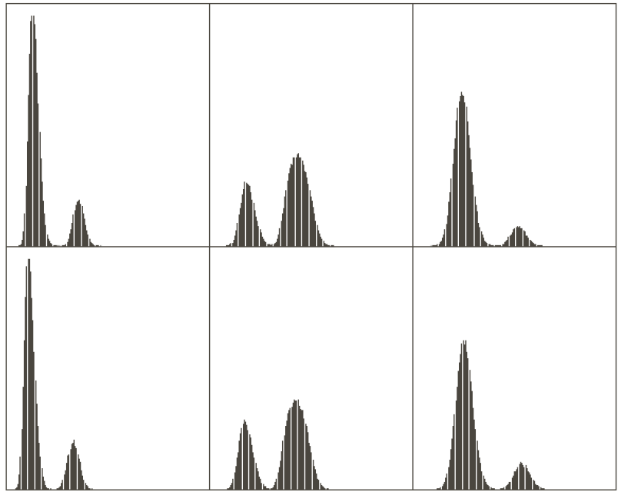
$$P_1 m_1 + P_2 m_2 + P_3 m_3 = m_G \qquad P_1 + P_2 + P_3 = 1$$

The two optimum thresholds k_1^* and k_2^* are the values that maximize $\sigma_B^2(k_1, k_2)$, then

$$g(x,y) = \begin{cases} a, & f(x,y) \le k_1^* \\ b, & k_1^* < f(x,y) \le k_2^* \\ c, & f(x,y) > k_2^* \end{cases} \quad \text{and} \quad \eta(k_1^*, k_2^*) = \frac{\sigma_B^2(k_1^*, k_2^*)}{\sigma_G^2}$$

Image partitioning（图像分块）



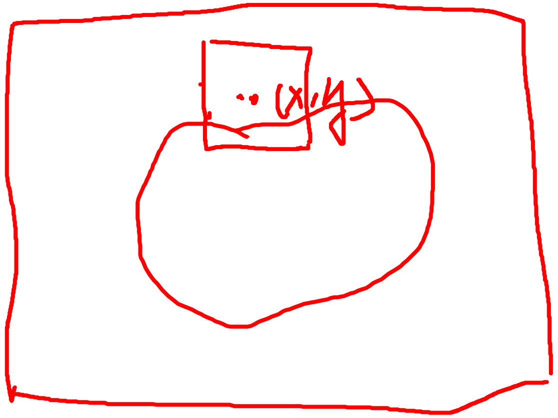


Variable thresholding based on local image properties

- We can make the rules like this:

Super bright

$$g(x,y) = \begin{cases} 1 & I(x,y) > \underbrace{\mu_{xy}}_{\text{mean}} + 2\underbrace{\sigma_{xy}}_{\text{var}} \\ 0 & else \end{cases}$$



- Turn this pixel on. If locally brighter than others or

$$g(x,y) = \begin{cases} 1 & I(x,y) > \mu_{xy} \\ 0 & else \end{cases}$$

different from avg

$$g(x,y) = \begin{cases} 1 & I(x,y) > |I(x,y) - \mu_{xy}| > 2\sigma_{xy} \\ 0 & else \end{cases}$$

MG

$$g(x,y) = \begin{cases} 1 & I(x,y) > \mu_{xy} + 2\sigma_{xy}\ and I(x,y) > \tau_{min} \\ 0 & else \end{cases}$$

locally bright

globally bright

Variable thresholding based on local image properties

- **Algorithm:**

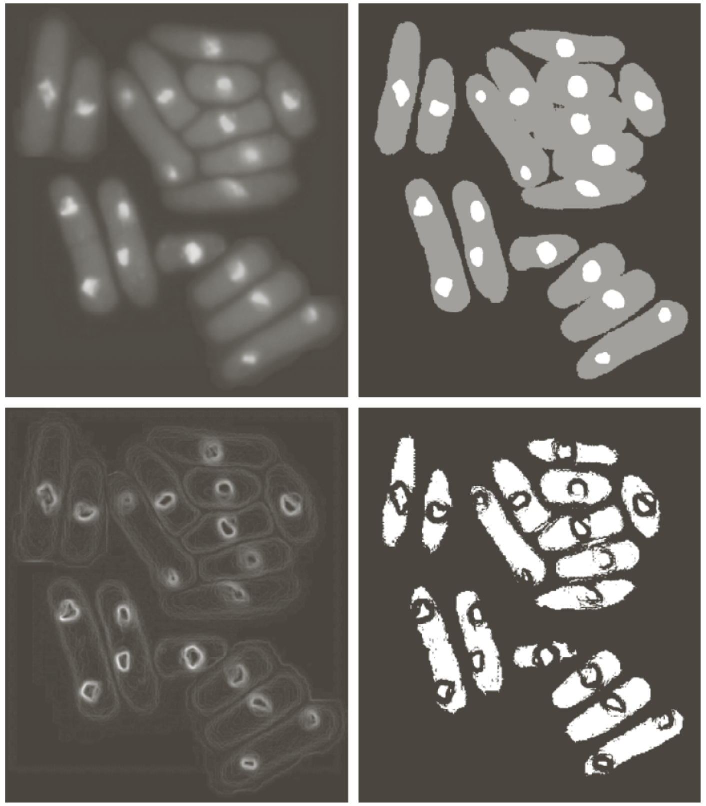
$$T_{xy} = a\sigma_{xy} + bm_{xy}$$

or

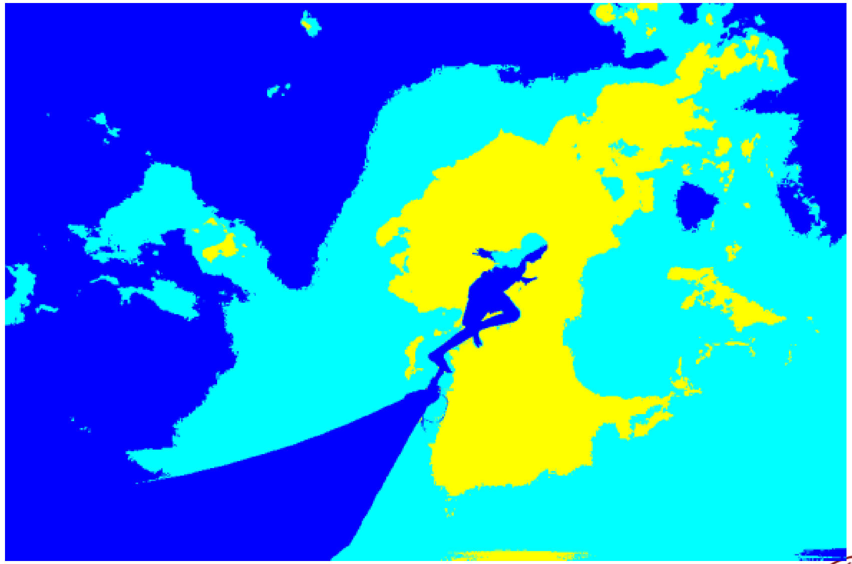
$$T_{xy} = a\sigma_{xy} + bm_G$$

- **Matlab function:**

```
g = stdfilt(f, nhood);
```



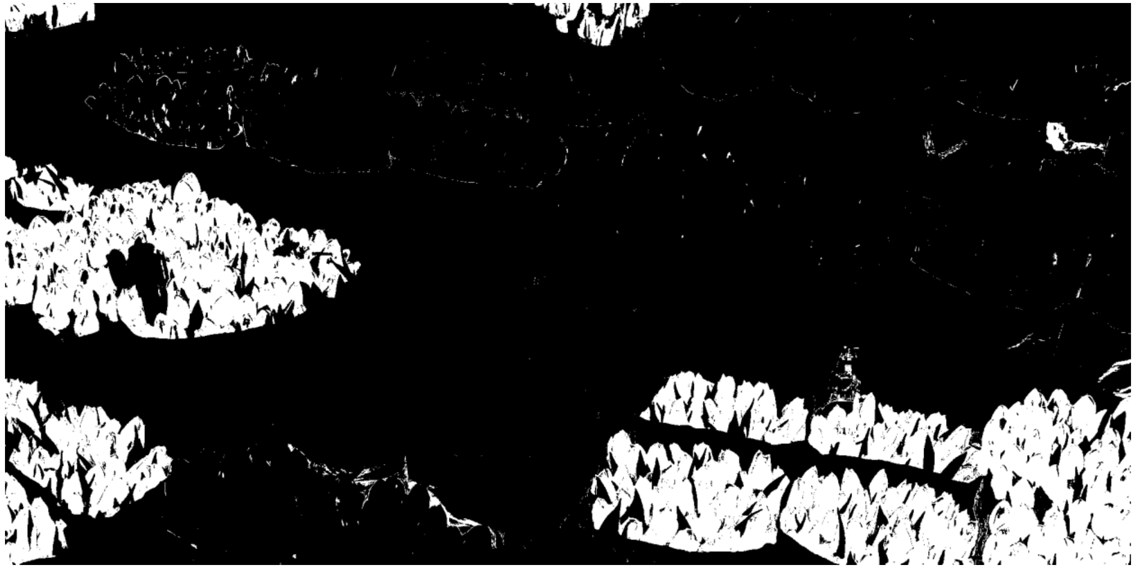


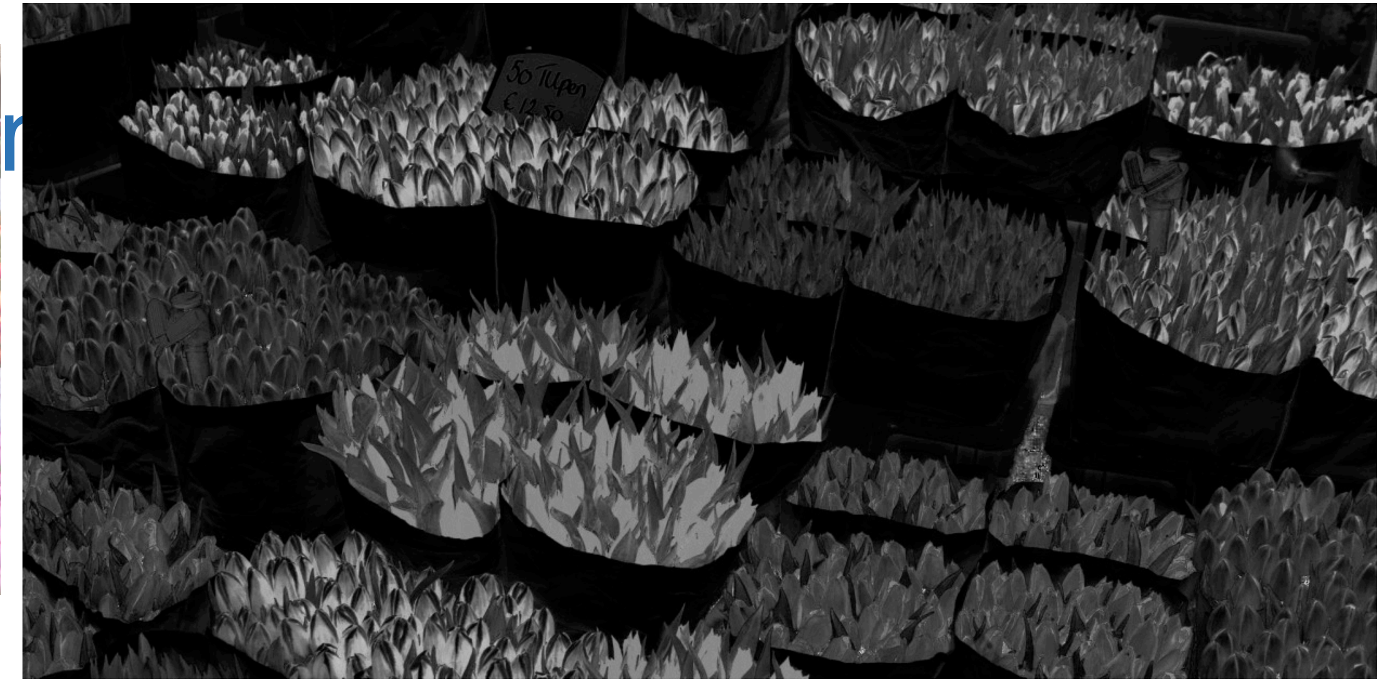



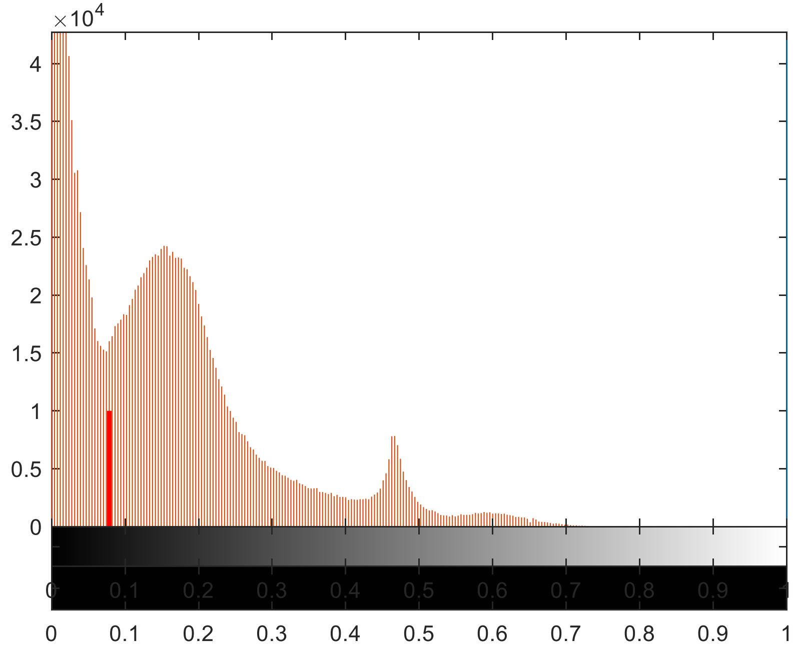
Apply thresholding in RGB color images

- Thresholding independently on RGB channels
- Combine channels

$$||I(x,y) - c|| < \tau$$







×10^4
4
3.5
3
2.5
2
1.5
1
0.5
0
0 0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9 1

Thresholding

- Global thresholding (全局阈值处理)
  - Basic global thresholding
  - Optimal global thresholding using Otsu’ s method
  - Improve global thresholding by using edges
  - Multiple thresholding
- Varible thresholding （可变阈值处理）
  - Image partitioning （图像分块）
  - Variable thresholding based on local image properties

Thresholding

- Global thresholding (全局阈值处理)
  - Basic global thresholding
  - Optimal global thresholding using Otsu’ s method
  - Improve global thresholding by using edges
  - Multiple thresholding
- Varible thresholding （可变阈值处理）
  - Image partitioning （图像分块）
  - Variable thresholding based on local image properties