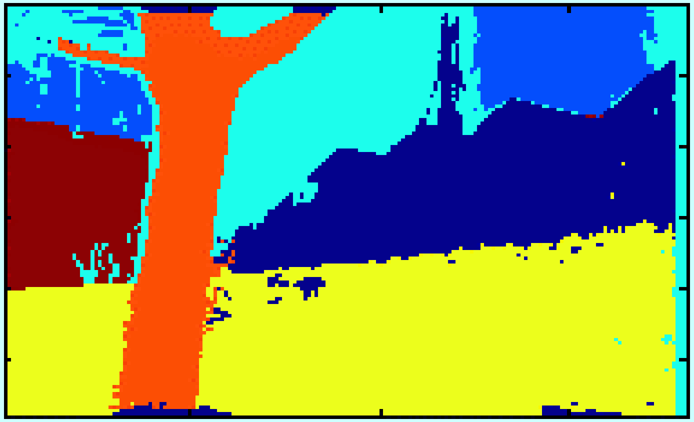
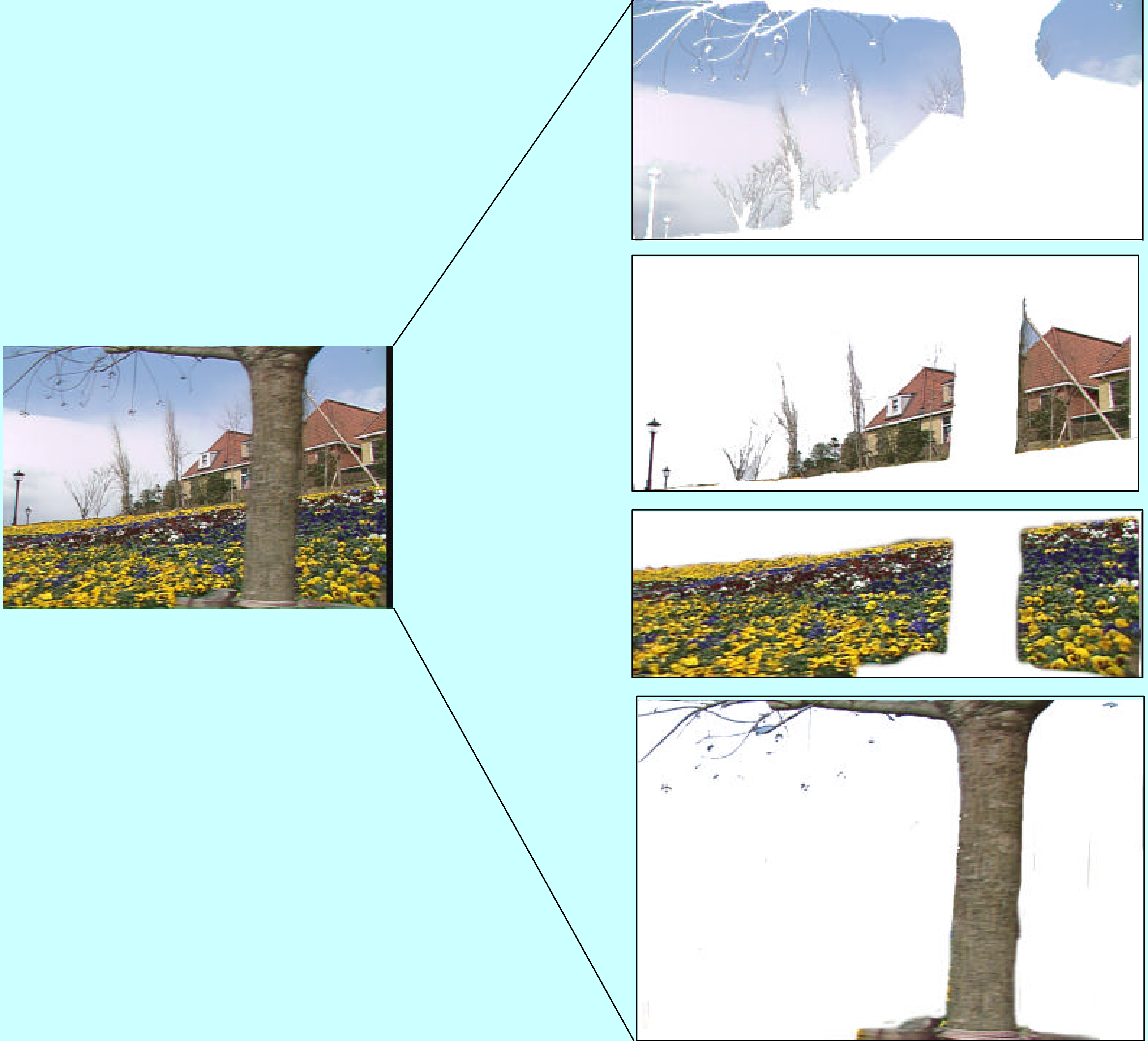
Lecture-9

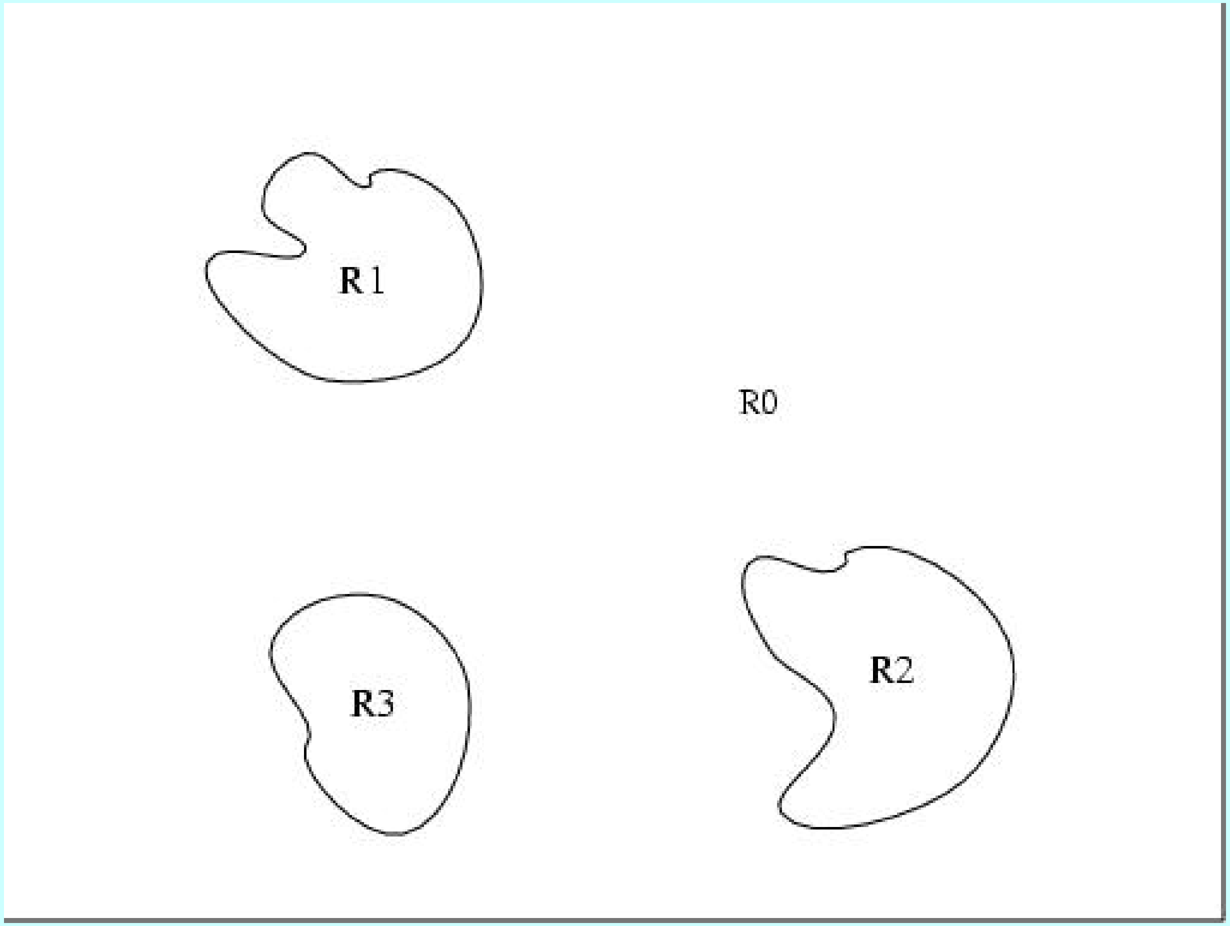
Region Segmentation

Region Segmentation





Segmentation



Segmentation

- Partition $f(x,y)$ into sub-images: R_1, R_2, …..,R_n such that the following constraints are satisfied:

  – Sum of all regions equal the given image

  – No two regions are overlapping

  –

$$\bigcup_{i=1}^{n} R_i = f(x, y)$$

$$R_i \cap R_j = f, i \neq j$$

- Each sub-mage satisfies one of the following predicates:
  - All pixels in any sub-image musts have the same gray levels.
  - All pixels in any sub-image must not differ more than some threshold
  - All pixels in any sub-image may not differ more than some threshold from the mean of the gray of the region
  - The standard deviation of gray levels in any sub-image must be small.

Simple Segmentation

$$B(x,y) = \begin{cases} 1 & \text{if } f(x,y) < T \\ 0 & \text{Otherwise} \end{cases}$$

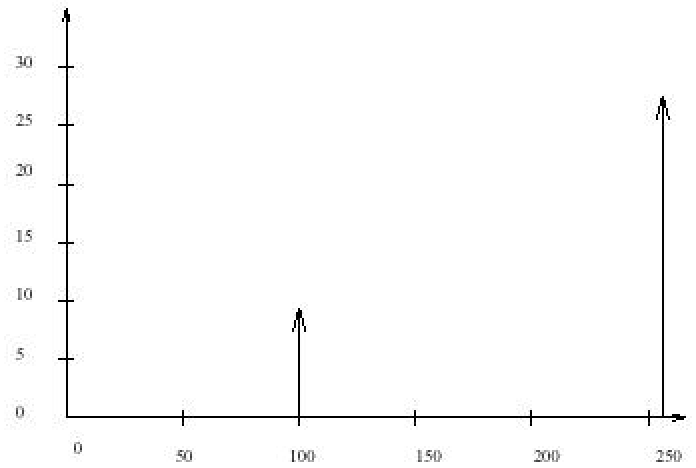
$$B(x,y) = \begin{cases} 1 & \text{if } T_1 < f(x,y) < T_2 \\ 0 & \text{Otherwise} \end{cases}$$

$$B(x,y) = \begin{cases} 1 & \text{if } f(x,y) \in Z \\ 0 & \text{Otherwise} \end{cases}$$

Histogram

Histogram graphs the number of pixels in an image with a Particular gray level as a function of the image of gray levels.

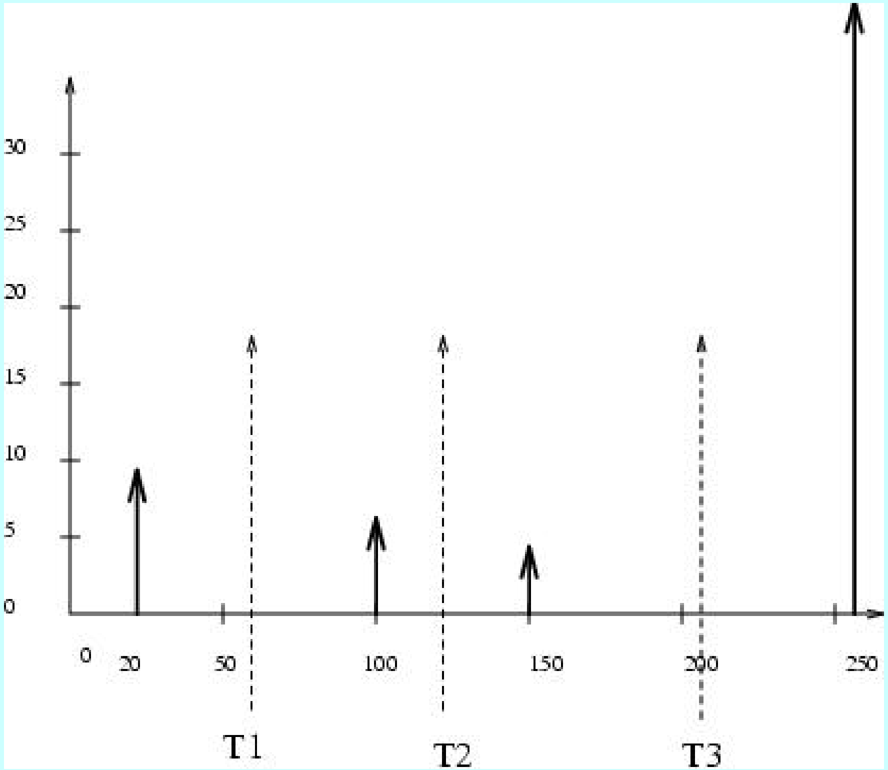
255	255	255	255	255	255
255	255	255	255	255	255
255	255	100	100	100	255
255	255	100	100	100	255
255	255	100	100	100	255
255	255	255	255	255	255




```
For (I=0, I<m, I++)
 For (J=0, J<m, J++)
   histogram[f(I,J)]++;
```

Example

255	255	255	255	255	255	255	20
255	255	255	100	100	255	20	20
255	255	255	100	100	255	20	20
255	255	255	100	100	255	20	20
255	255	255	255	255	255	20	20
255	255	255	255	255	255	255	255
150	150	255	255	255	255	255	255
150	150	255	255	255	255	255	255



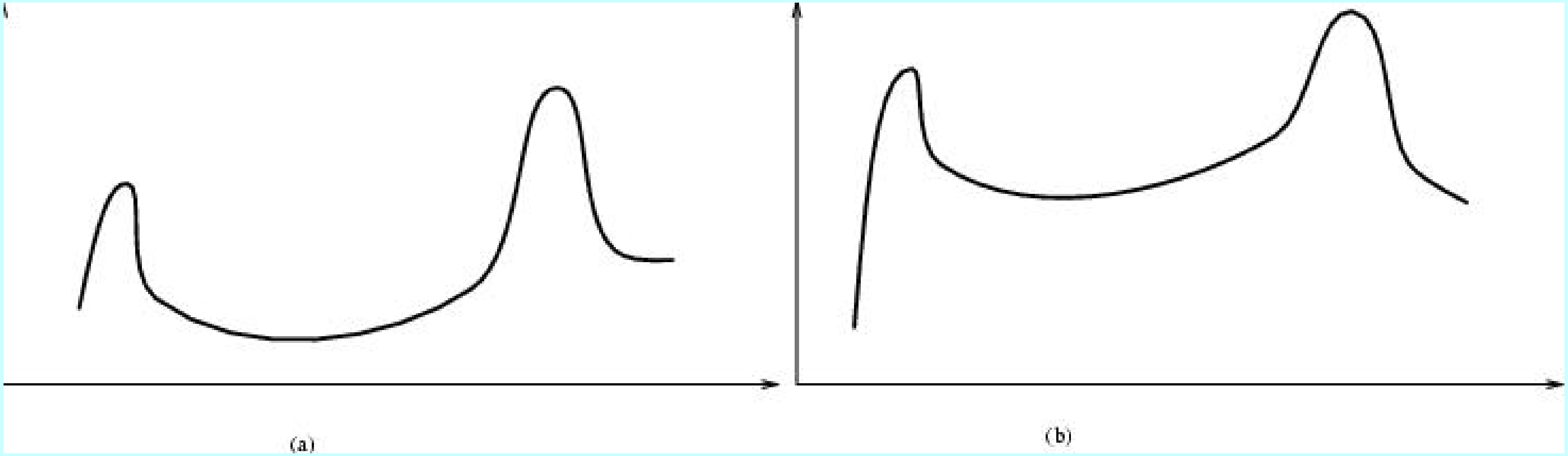
Segmentation Using Histogram

$$B_1(x,y) = \begin{cases} 1 & \text{if } 0 < f(x,y) < T_1 \\ 0 & \text{Otherwise} \end{cases}$$

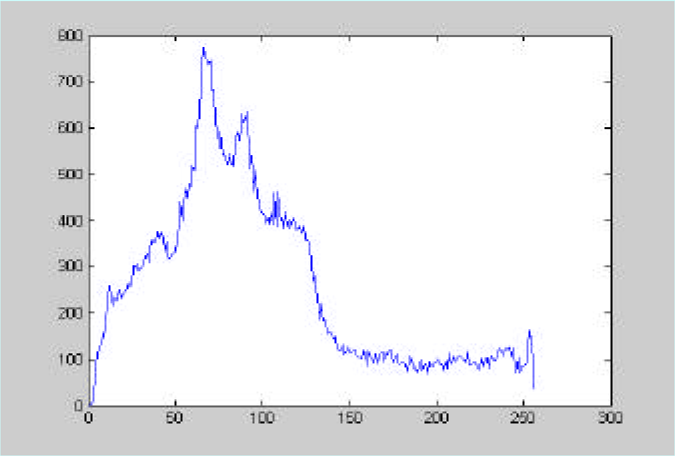
$$B_2(x,y) = \begin{cases} 1 & \text{if } T_1 < f(x,y) < T_2 \\ 0 & \text{Otherwise} \end{cases}$$

$$B_3(x,y) = \begin{cases} 1 & \text{if } T_2 < f(x,y) < T_3 \\ 0 & \text{Otherwise} \end{cases}$$

Realistic Histogram

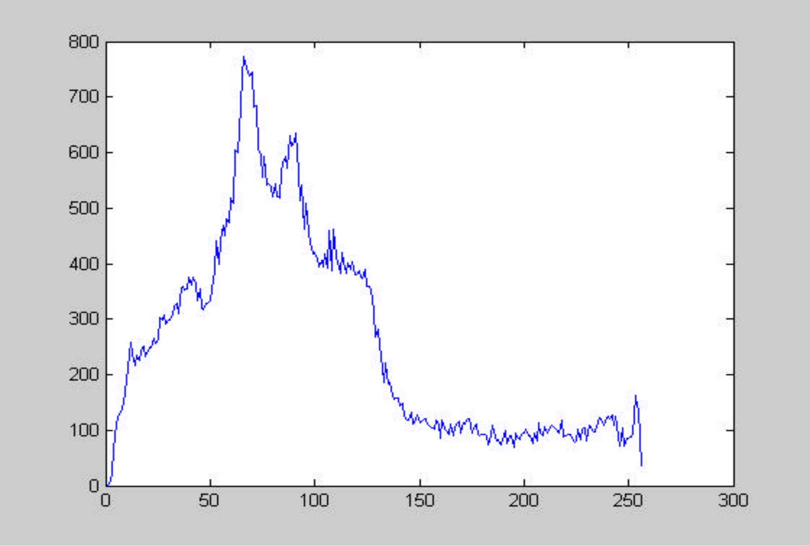


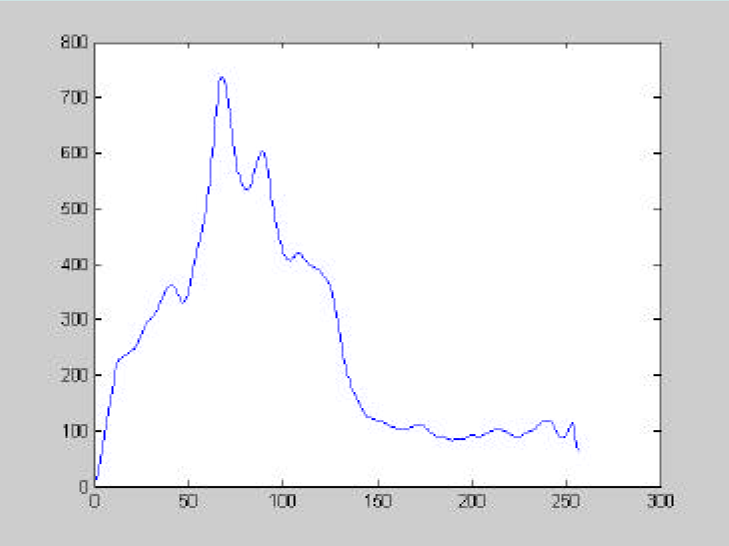
Realistic Histogram



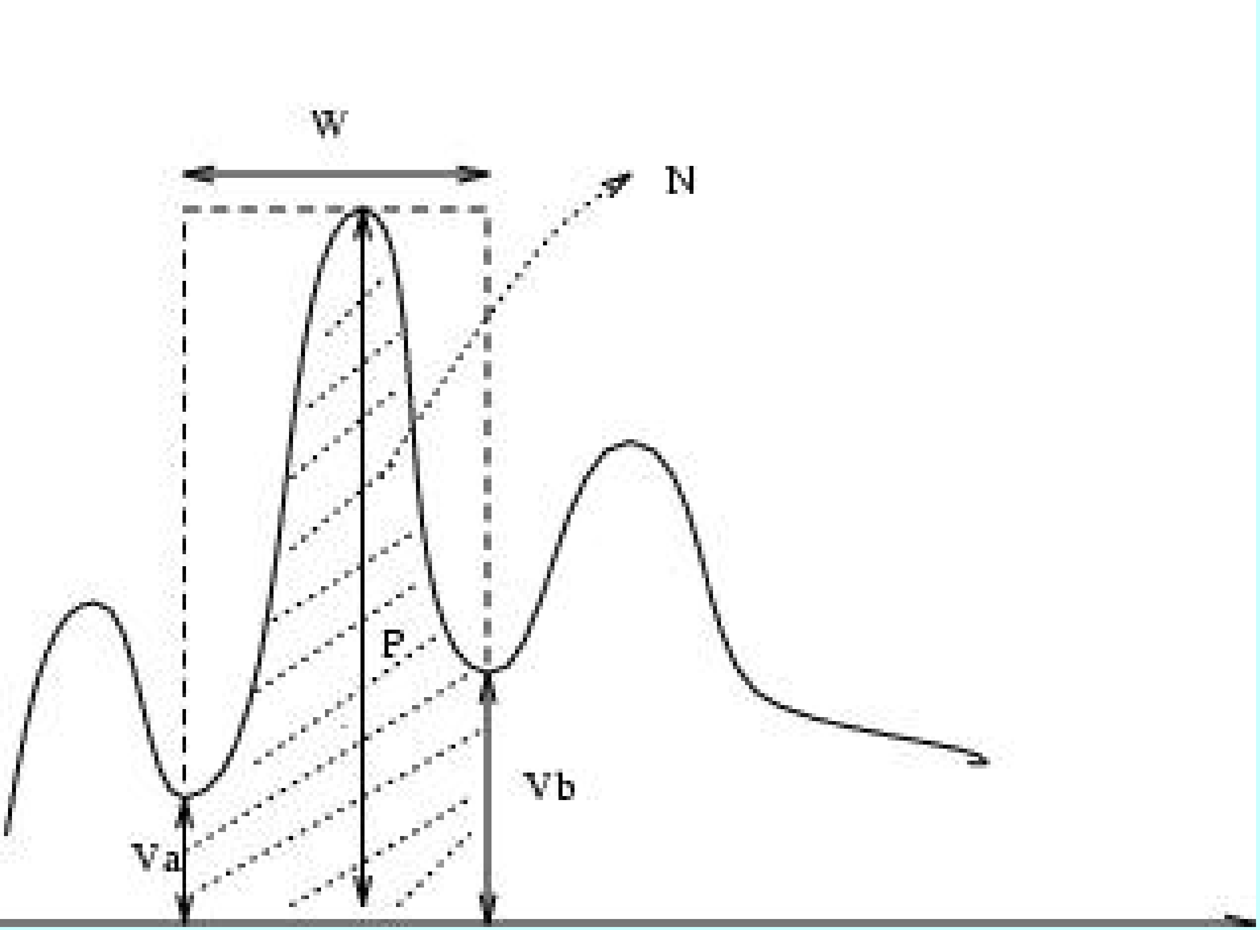
Smoothing the Histogram

Convolve by averaging
or Gaussian Filter





Finding peaks

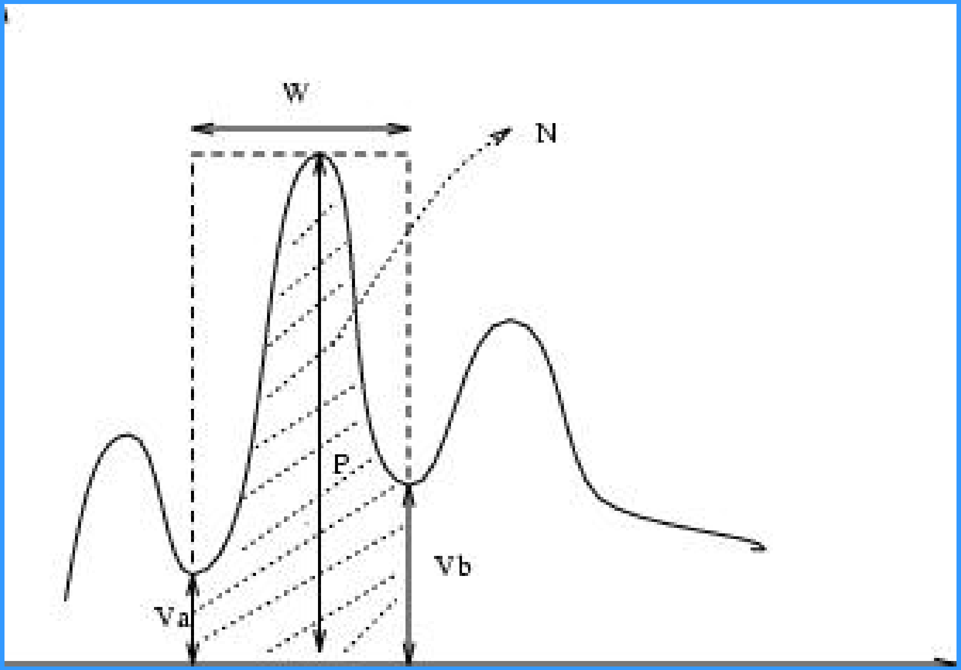


Sharpness of a peak:

W / N. P

Worst case: 1

Peakiness Test



$$Peakiness = \left(1 - \frac{(V_a + V_b)}{2P}\right) \cdot \left(1 - \frac{N}{(W.P)}\right)$$

Connected Component

$$\begin{bmatrix} 0 & 0 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 \end{bmatrix} \quad \begin{bmatrix} 0 & 0 & 0 & a & 0 \\ b & b & 0 & a & a \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & c & c & 0 \\ 0 & d & 0 & c & 0 \end{bmatrix}$$

4

Connectedness

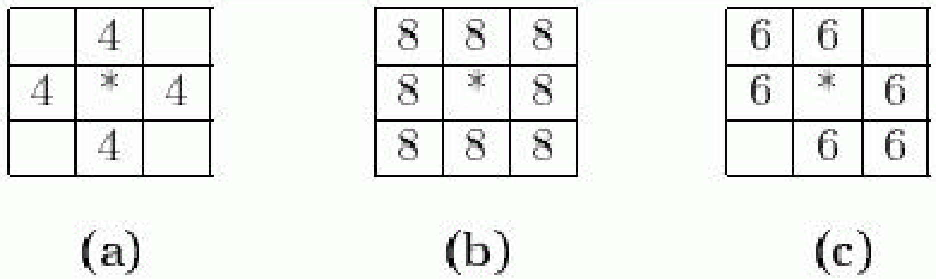


Figure 3.6: Pixel connectedness. (a) 4-connected. (b) 8-connected. (c) 6-connected.

4 - Connectedness

	4	
4	x	4
	4	

Connected Component

$$\begin{bmatrix} 0 & 0 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 \end{bmatrix} \quad \begin{bmatrix} 0 & 0 & 0 & a & 0 \\ b & b & 0 & a & a \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & c & c & 0 \\ 0 & c & 0 & c & 0 \end{bmatrix}$$

Recursive Connected Component Algorithm

1. Scan the binary image left to right and top to bottom.
2. Assign a new label to a unlabeled pixel with value ‘1’.
3. Recursively search neighbors of that pixel and assign same label if they have value ‘1’ and are still unlabeled.
4. Stop when all pixels with value ‘1’ have been labeled.

Recursive

$$\begin{bmatrix} 0 & 0 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 & 0 \end{bmatrix} \quad \begin{bmatrix} 0 & 0 & 0 & a & 0 \\ b & b & 0 & a & a \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & c & c & 0 \\ 0 & c & c & c & 0 \end{bmatrix}$$

Sequential Connected Component Algorithm

1. Scan the binary image left to right and top to bottom.
2. Label assignment to a unlabeled pixel with value ‘1’ is done as follows.

0	0	0	0
0 1	0 L	L 1	L L
L	L	L	L
0 1	0 L	M 1	M L

3. Determine equivalence classes of labels.

4. In the second pass over the image, all elements belonging to an equivalence class are assigned the same label .

Sequential

$$\begin{bmatrix} 0 & 0 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 & 0 \end{bmatrix} \qquad \begin{bmatrix} 0 & 0 & 0 & a & 0 \\ b & b & 0 & a & a \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & c & c & 0 \\ 0 & d & c & c & 0 \end{bmatrix} \quad d=c$$

Rules for 8-connectivity

$$\begin{array}{ccc}0&0&0\\0&1&\end{array}\rightarrow\begin{array}{ccc}0&0&0\\0&L&\end{array}$$

$$\begin{array}{ccc}L&0&0\\0&1&\end{array}\rightarrow\begin{array}{ccc}L&0&0\\0&L&\end{array}$$

$$\begin{array}{ccc}0&L&0\\0&1&\end{array}\rightarrow\begin{array}{ccc}0&L&0\\0&L&\end{array}$$

$$\begin{array}{ccc}0&0&L\\0&1&\end{array}\rightarrow\begin{array}{ccc}0&0&L\\0&L&\end{array}$$

$$\begin{array}{ccc}*&*&0\\L&1&\end{array}\rightarrow\begin{array}{ccc}0&0&0\\L&L&\end{array}$$

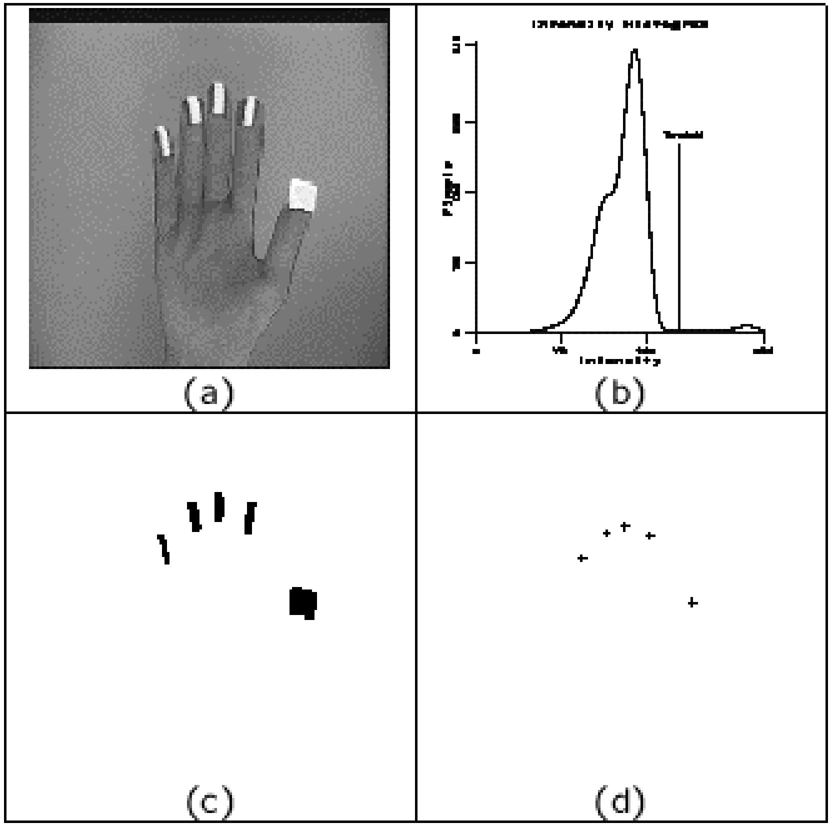
$$\begin{array}{ccc}*&*&L\\M&1&\end{array}\rightarrow\begin{array}{ccc}*&*&L\\M&L&\end{array}$$

(Set $L=M$)

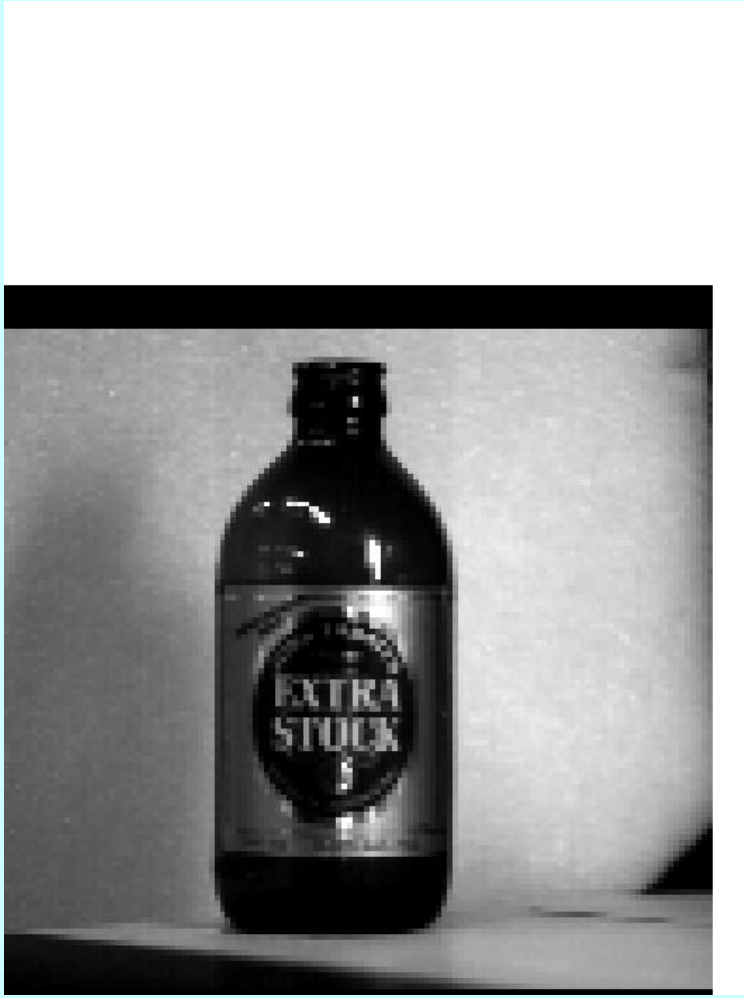
Steps in Segmentation Using Histogram

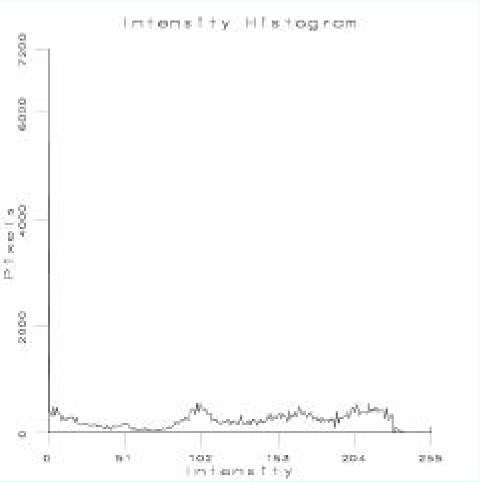
1. Compute the histogram of a given image.
2. Smooth the histogram by averaging peaks and valleys in the histogram.
3. Detect good peaks by applying thresholds at the valleys.
4. Segment the image into several binary images using thresholds at the valleys.
5. Apply connected component algorithm to each binary image find connected regions.

Example: Detecting Fingertips



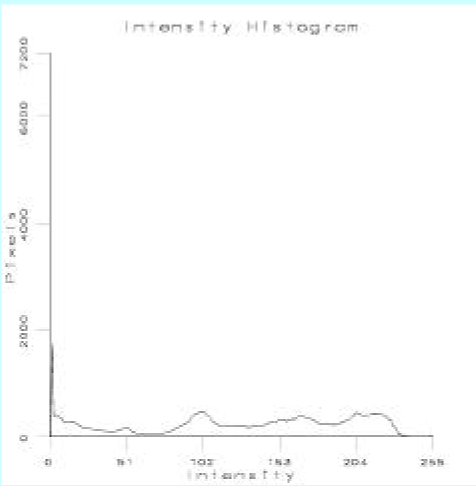
Example-II



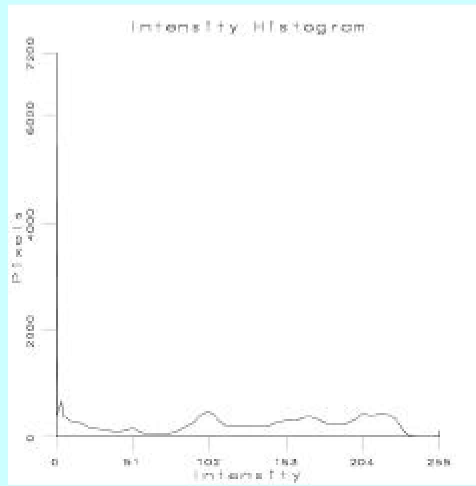


93 peaks

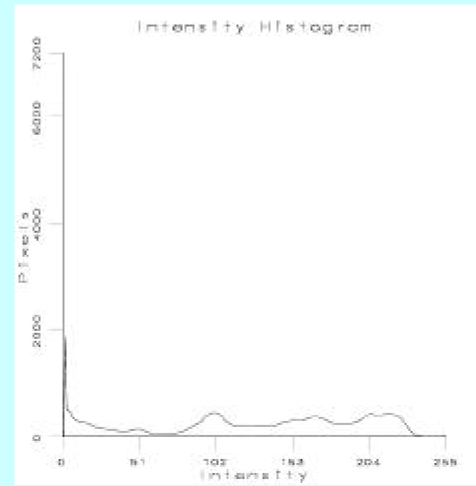
Smoothed Histograms



Smoothed histogram
(averaging using mask
Of size 5, one pass
gives 54 peaks
Peakiness test gives
18 peaks



Twice Smoothed
histogram
21 peaks
After peakiness
Gives 7 peaks



After 3 Smoothings

11 peaks
After peakiness
Gives 4 peaks

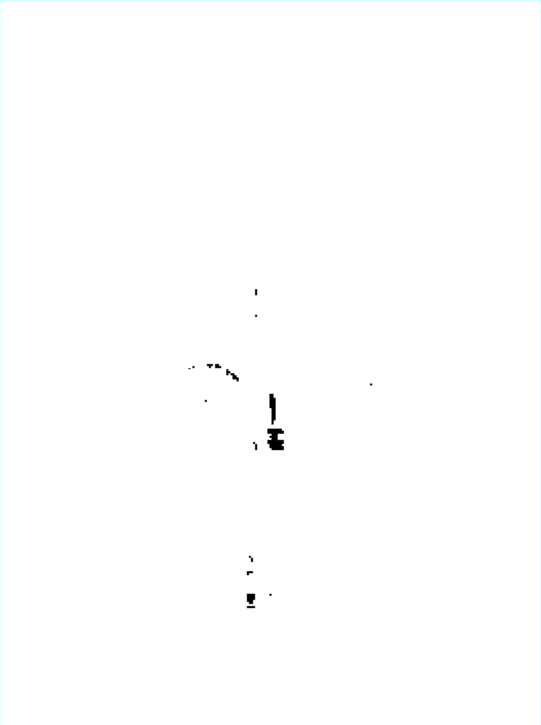
Regions

Regions from peak1
(0,…..,40)

Regions from peak2
(40,….,116)

Regions

Regions from peak 3
(116,….,243)



Regions from peak 4
(243,….,255)

Steps in Seed Segmentation Using Histogram

1. Compute the histogram of a given image.
2. Smooth the histogram by averaging peaks and valleys in the histogram.
3. Detect good peaks by applying thresholds at the valleys.
4. Segment the image into several binary images using thresholds at the valleys.
5. Apply connected component algorithm to each binary image find connected regions.