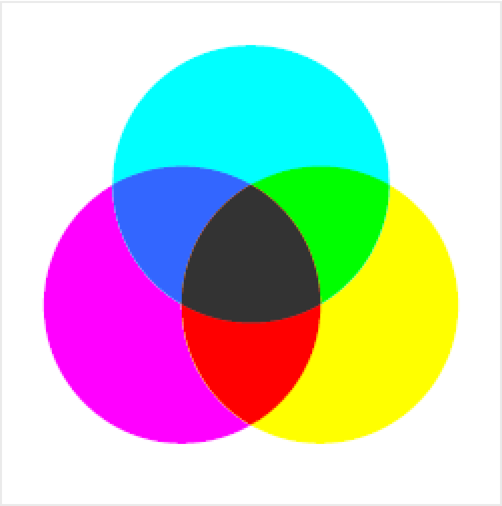
Lecture 2

Cheng-Ming Huang

EE, NTUT

Color Spaces

- Additive colors red, green, and blue
  - can be mixed to produce cyan, magenta, yellow, and white
- Subtractive colors cyan, magenta, and yellow
  - can be mixed to produce red, green, blue, and black



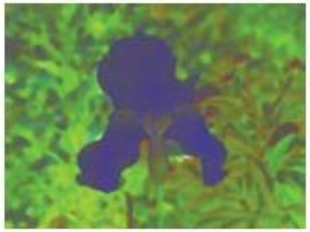
Color Spaces


(a) RGB

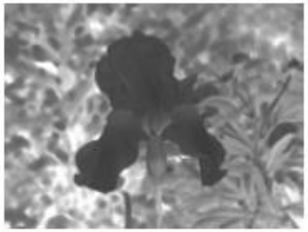

(b) R


(c) G


(d) B


(e) rgb


(f) r


(g) g


(h) b

$$r = \frac{R}{R+G+B}$$

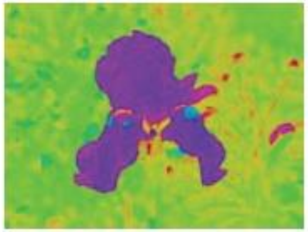
$$g = \frac{G}{R+G+B}$$

$$b = \frac{B}{R+G+B}$$


(i) L*


(j) a*


(k) b*


(l) H


(m) S


(n) V

Color Spaces

- XYZ color space
  - contain all of the pure spectral colors within its positive octant
  - separate luminance from chrominance

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \frac{1}{0.17697} \begin{bmatrix} 0.49 & 0.31 & 0.20 \\ 0.17697 & 0.81240 & 0.01063 \\ 0.00 & 0.01 & 0.99 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix}$$

- L*a*b* color space
  - predict how humans perceive differences in color or luminance

$$L^* = 116 f\left(\frac{Y}{Y_n}\right) \qquad f(t) = \begin{cases} t^{1/3} & t > \delta^3 \\ t/(3\delta^2) + 2\delta/3 & \text{else,} \end{cases}$$

$$a^* = 500\left[f\left(\frac{X}{X_n}\right) - f\left(\frac{Y}{Y_n}\right)\right] \text{ and } b^* = 200\left[f\left(\frac{Y}{Y_n}\right) - f\left(\frac{Z}{Z_n}\right)\right]$$

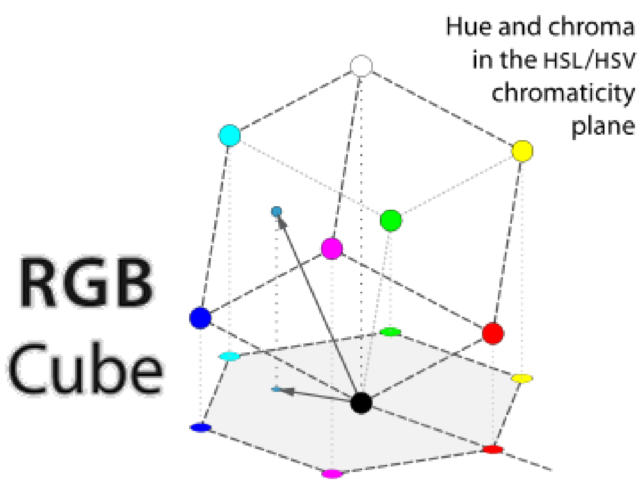
HSV Color Space

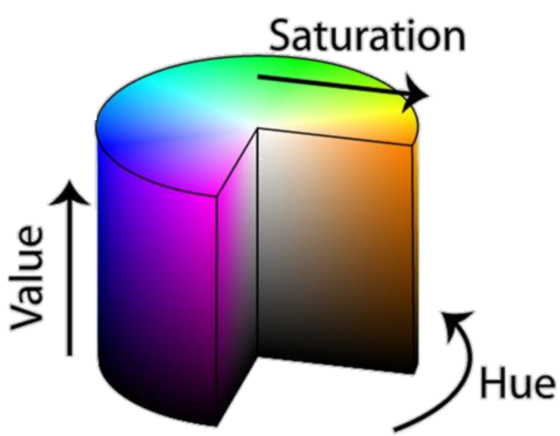
- Cylindrical-coordinate representation of RGB model
  - more perceptually relevant than Cartesian representation
  - more distinguishable on skin color

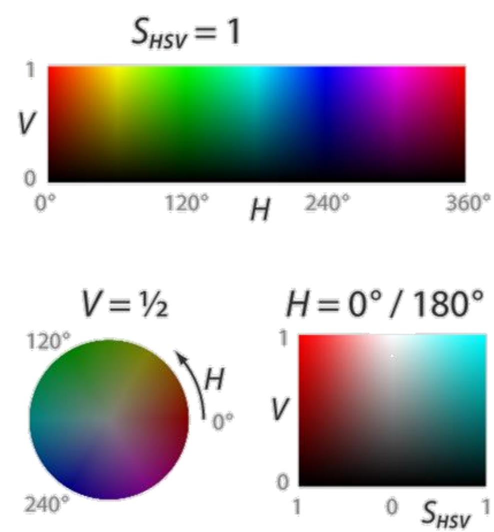
$$V = \max(R, G, B)$$

$$S = \frac{V - \min(R, G, B)}{V}$$

$$H = \begin{cases} 60 * \left(\frac{G-B}{S} \bmod 6 \right), & \text{if } V = R \\ 180 + 60 * \frac{B-R}{S}, & \text{if } V = G \\ 240 + 60 * \frac{R-G}{S}, & \text{if } V = B \end{cases}$$



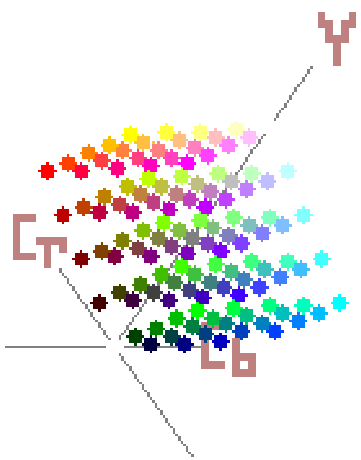


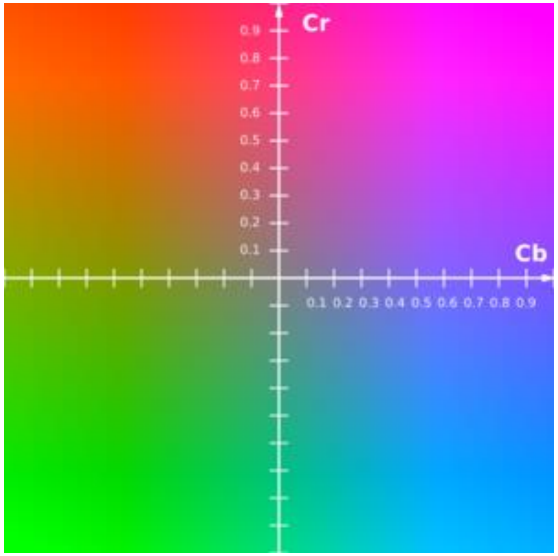


YCbCr Color Space

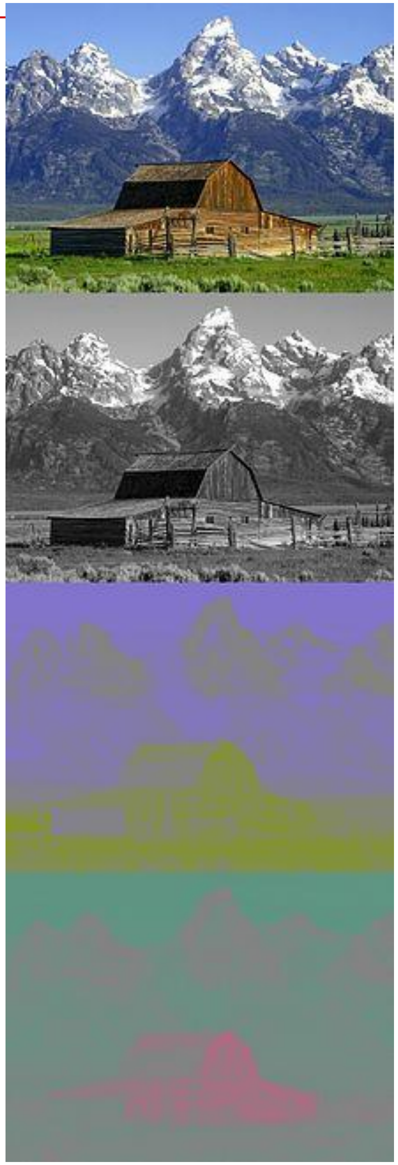
- JPEG, composite TV signal
  - Closely related to YUV, YIQ
  - Y: gray-level image

$$\begin{bmatrix} Y' \\ C_b \\ C_r \end{bmatrix} = \begin{bmatrix} 0.299 & 0.587 & 0.114 \\ -0.168736 & -0.331264 & 0.5 \\ 0.5 & -0.418688 & -0.081312 \end{bmatrix} \begin{bmatrix} R' \\ G' \\ B' \end{bmatrix} + \begin{bmatrix} 0 \\ 128 \\ 128 \end{bmatrix}$$





The CbCr plane at constant luma Y'=0.5



Y

Cb

Cr

Feature Selection on Color Spaces

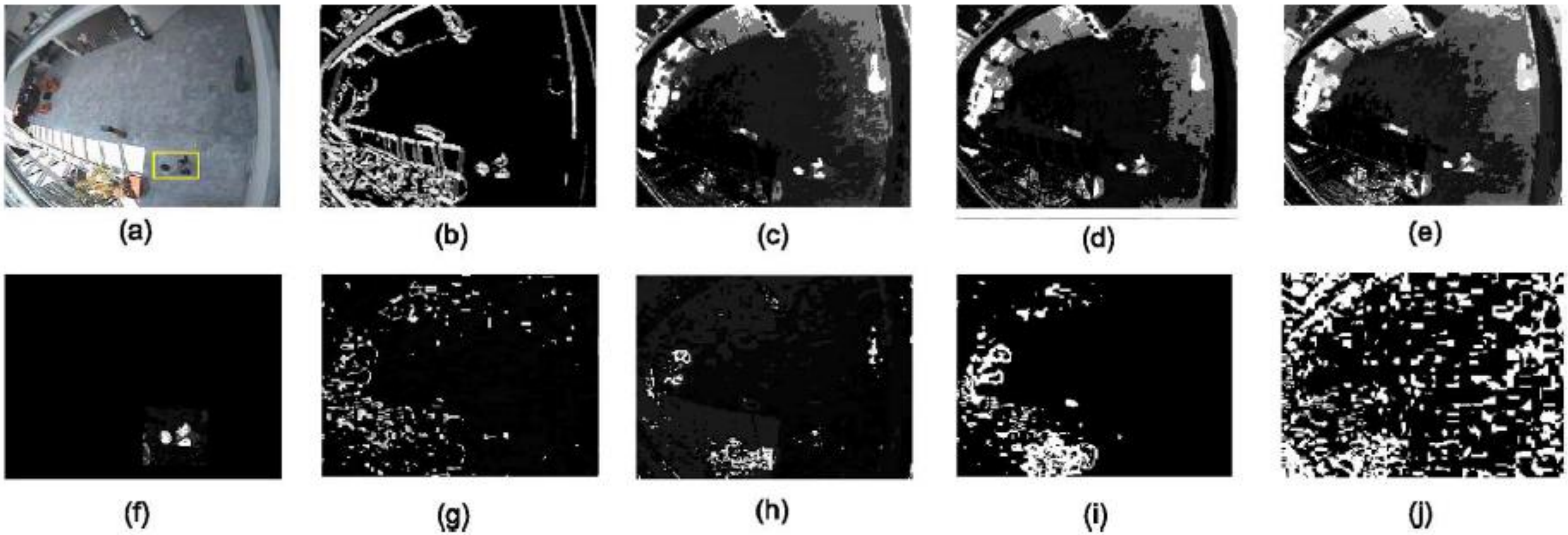
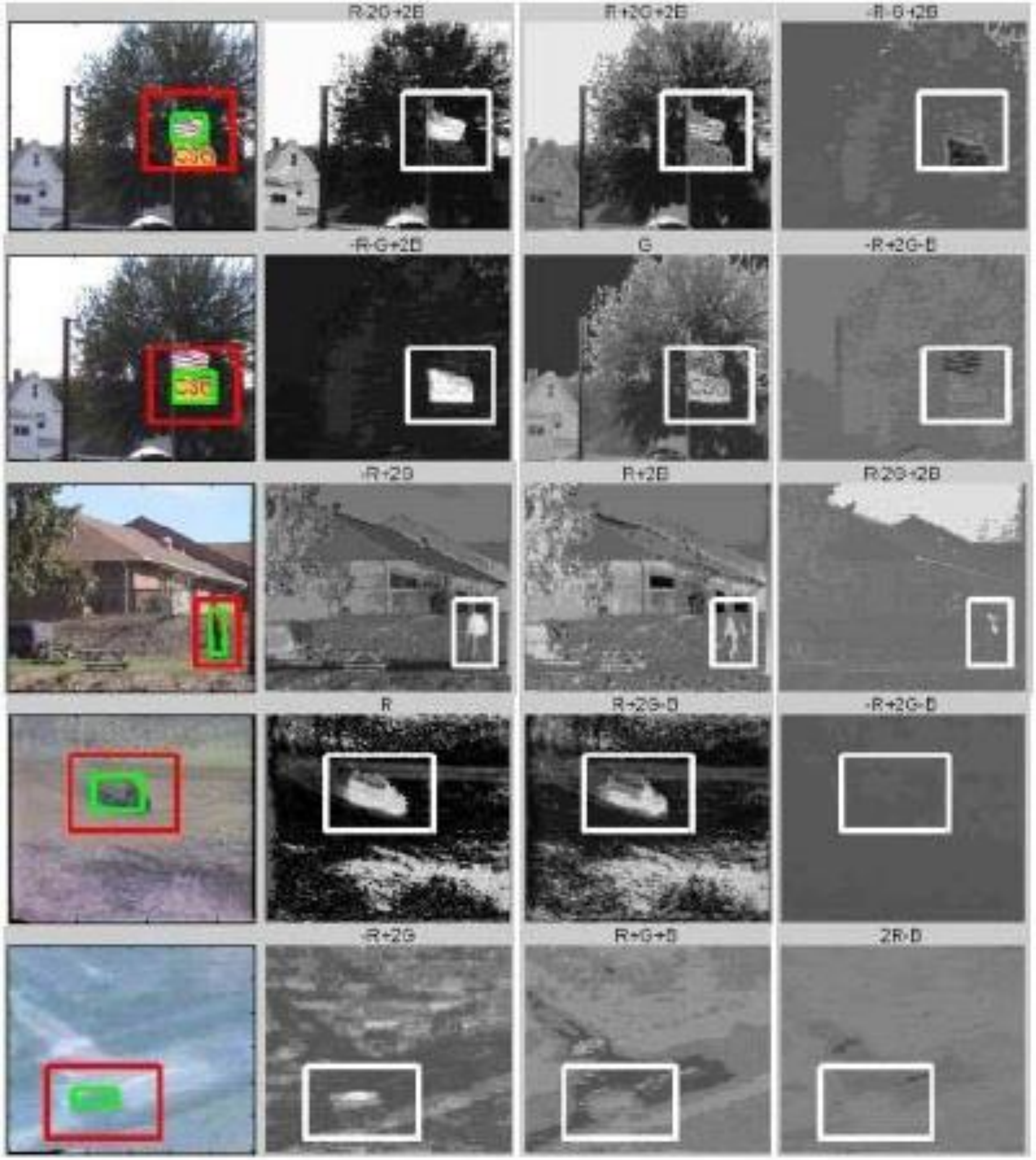


Fig. 1. (a) Input image. The likelihood ratio images of (b) the orientation histogram; (c) R; (d) G; (e) B; (f) integration of the best two features: orientation histogram and R; (g) H; (h) S; (i) r; and (j) g. Note that the likelihood image in (f) shows only the neighborhood of the target.

J. Wang and Y. Yagi, “Integrating color and shape-texture features for adaptive real-time object tracking,” *IEEE Trans. Image Processing*, vol. 17, pp. 235-240, 2008.

Feature Selection on Color Spaces



R.T. Collins, Y. Liu, “On-line selection of discriminative tracking features,” *IEEE ICCV* 2003.

Distance measures

- Definition

  three pixel p, q, z

  (1). $D(p,q) \geqq 0$ ($D(p,q) = 0$ iff $p = q$)

  (2). $D(p,q) = D(q,p)$, and

  (3). $D(p,z) \leqq D(p,q) + D(q,z)$

- Euclidean distance (x,y) to (s,t)

  $$[\,(x - s)^2 + (y - t)^2\,]^{1/2}$$

- D_4 distance

  $$|\,x - s\,| + |\,y - t\,|$$

- D_8 distance

  $$\max(\,|\,x - s\,|\,,\,|\,y - t\,|\,)$$

(0,0)	(1,0)	(2,0)
(0,1)	(1,1)	(2,1)
(0,2)	(1,2)	(2,2)

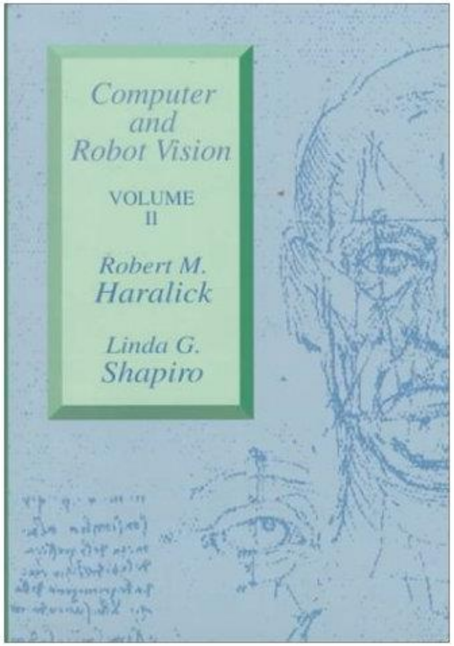
(0,0)	(1,0)	(2,0)
(0,1)	(1,1)	(2,1)
(0,2)	(1,2)	(2,2)

Binary Machine Vision

- binary value 1: considered part of object
- binary value 0: background pixel

- binary machine vision: generation and analysis of binary image
  - Thresholding
  - Segmentation

Computer and Robot Vision (Volume I&II)
Robert M. Haralick, Linda G. Shapiro



Thresholding

- $B(r,c)=1$ if $I(r,c) \geq T$
- $B(r,c)=0$ if $I(r,c) < T$

- r: row, c: column
- I: grayscale intensity, B: binary intensity
- T: intensity threshold

Thresholding

								5	8	9			
									9	8	9		
		5								7	9	8	
		3									6	9	9
		5				7							
	2					6							
	8						5			1	1		
3							6			1	1	1	
							7						
	4	3							3	2	6	2	
	2	4							3	8	4	3	
	5	9							7	2	3	9	

Figure 2.1 Original gray scale image. Pixels having no numbers have value of 0.

$$\xrightarrow{T=1}$$

								1	1	1			
									1	1	1		
		1								1	1	1	
		1									1	1	1
		1				1							
	1					1							
	1						1			1	1		
1							1			1	1	1	
							1						
	1	1							1	1	1	1	
	1	1							1	1	1	1	
	1	1							1	1	1	1	

Figure 2.2 Thresholded gray scale image. All pixels greater than 0 are marked with a binary 1.

Thresholding

- Implementation

$$I(r,c)=255, \quad I(r,c)\geq T$$

$$I(r,c)=0, \quad I(r,c)<T$$

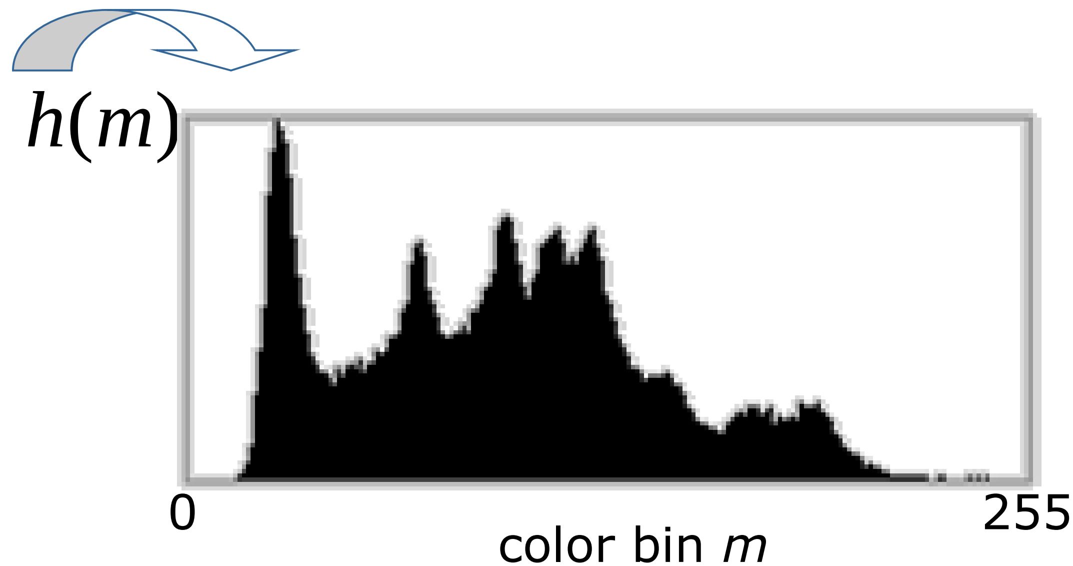
$\xrightarrow{T=128}$

$\xrightarrow{T=60}$

$\xrightarrow{T=200}$

Thresholding

- histogram $h(m) = \#\{(r,c) \mid I(r,c) = m\}$
  - *m* spans each gray level value e.g. 0 - 255
  - #: operator counts the number of elements in a set



Thresholding

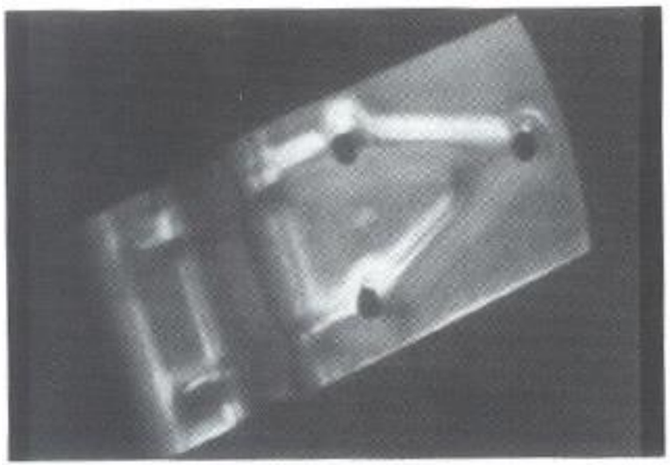


Figure 2.3 Image of a metal part.

$$h(m)$$

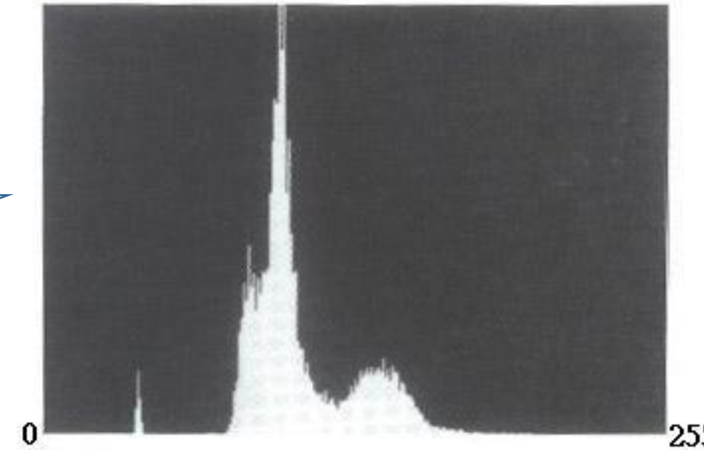


Figure 2.4 Histogram of the image of Fig. 2.3. The histogram shows two dominant modes. The small mode on the left tail is not significant.

$$T = 148$$

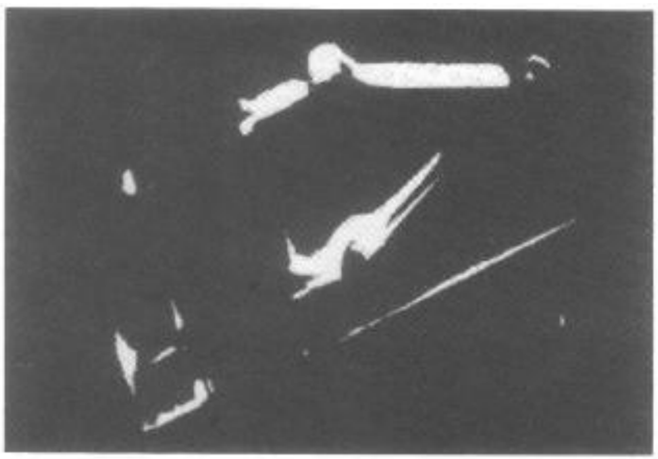
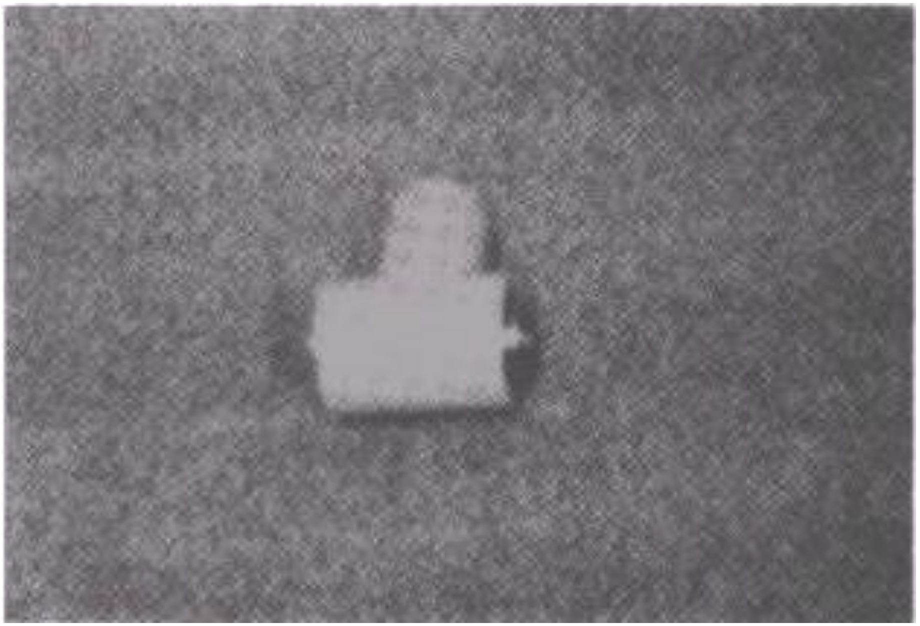


Figure 2.5 Metal-part image of Fig. 2.3 thresholded at gray level 148, which is in the valley between the two dominant modes.

Thresholding

$h(m)$



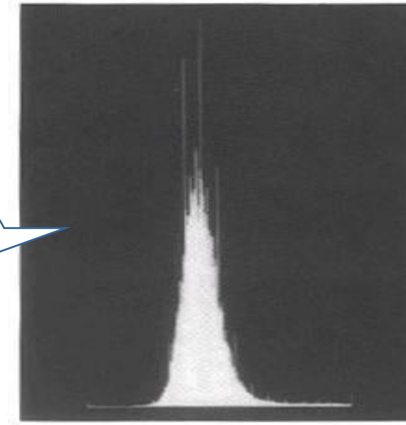


Figure 2.7 Histogram of the BNC T-connector image of Fig. 2.6.

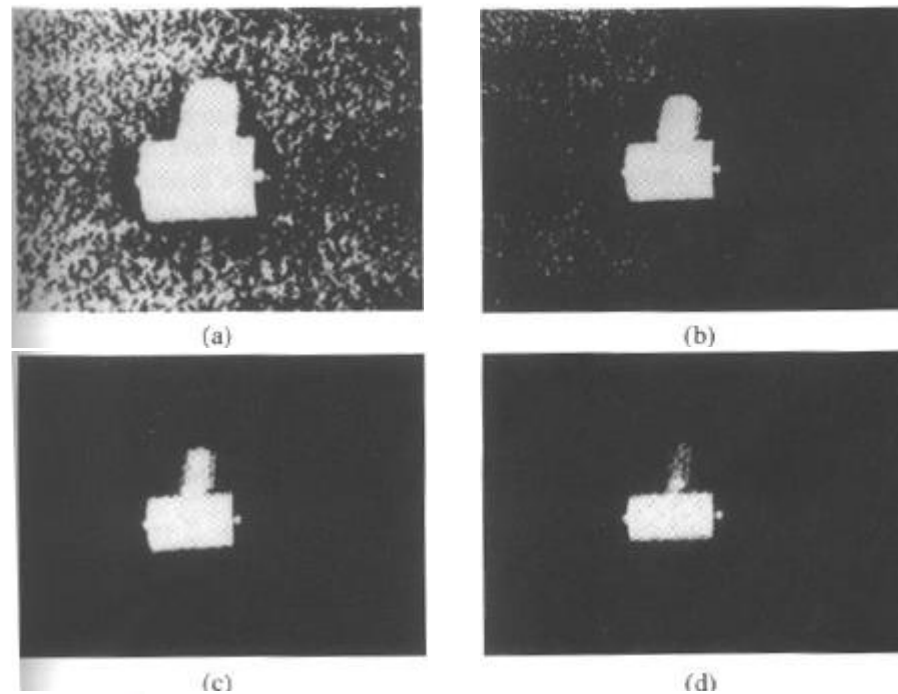


Figure 2.8 Image of the BNC T-connector thresholded at four levels: (a) 110, (b) 130, (c) 150, and (d) 170.

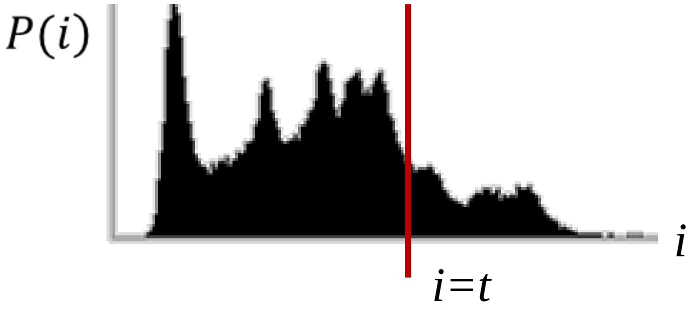
Minimizing Within-Group Variance

$P(1),\ldots,P(I)$: histogram probabilities of gray values $1\ldots I$

$$P(i)=\frac{\#\{(r,c)\,|\,Intensity(r,c)=i\}}{R\times C}$$

$R\times C$: the spatial domain of the image

Minimizing Within-Group Variance



Within-group variance σ_W^2: weighted sum of group variances

$$\sigma_W^2 = q_1(t)\sigma_1^2(t) + q_2(t)\sigma_2^2(t)$$

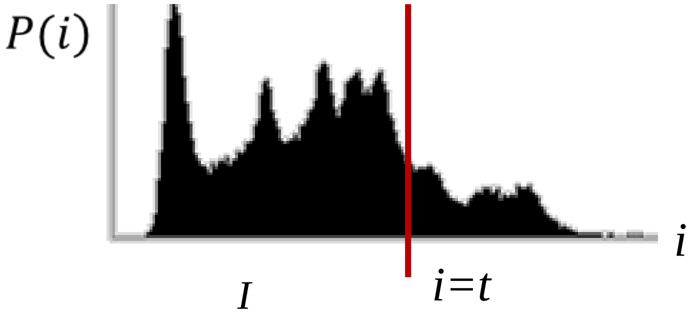
$q_{1(t)}$: probability for the group with values $\leq t$

$q_{2(t)}$: probability for the group with values $> t$

$\sigma_1^2(t)$: variance for the group with values $\leq t$

$\sigma_2^2(t)$: variance for the group with values $> t$

Minimizing Within-Group Variance



$$q_1(t) = \sum_{i=1}^{t} P(i) \qquad q_2(t) = \sum_{i=t+1}^{I} P(i)$$

$$\mu_1(t) = \sum_{i=1}^{t} iP(i) / q_1(t) \qquad \mu_2(t) = \sum_{i=t+1}^{I} iP(i) / q_2(t)$$

$$\sigma_1^2(t) = \sum_{i=1}^{t} [i - \mu_1(t)]^2 P(i) / q_1(t) \qquad \sigma_2^2(t) = \sum_{i=t+i}^{I} [i - \mu_2(t)]^2 P(i) / q_2(t)$$

Find t which minimizes $\sigma_W^2(t)$

Minimizing Within-Group Variance

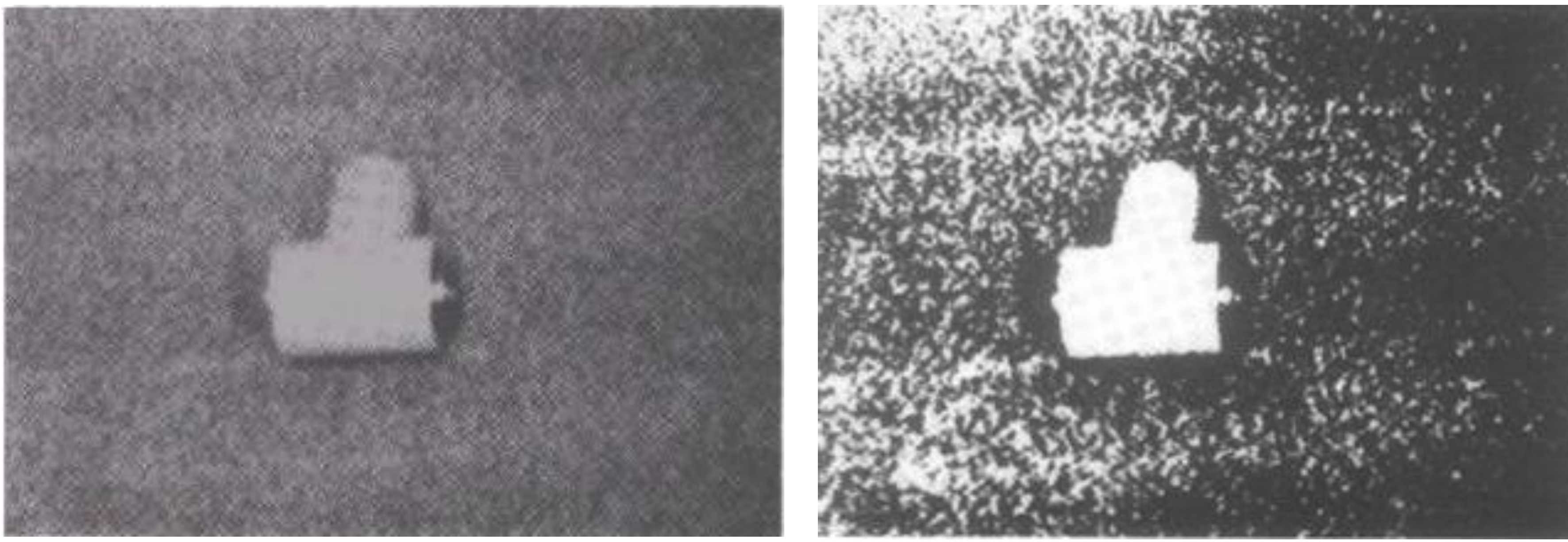


Figure 2.9 Binary image produced by thresholding the T-connector image of Fig. 2.6 with the Otsu threshold.

Minimizing Kullback Information Distance

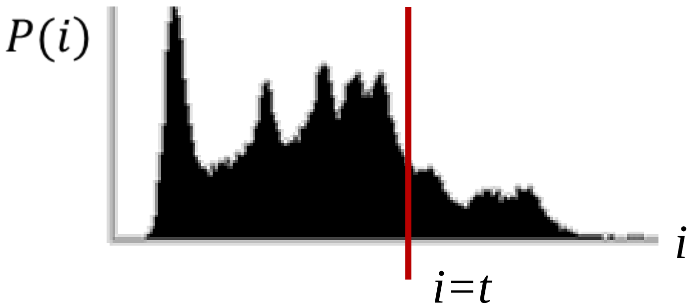
minimize the Kullback directed divergence J

- $f(i) \approx P(i)$

$$J = \sum_{i=1}^{I} P(i) \log\left[\frac{P(i)}{f(i)}\right]$$

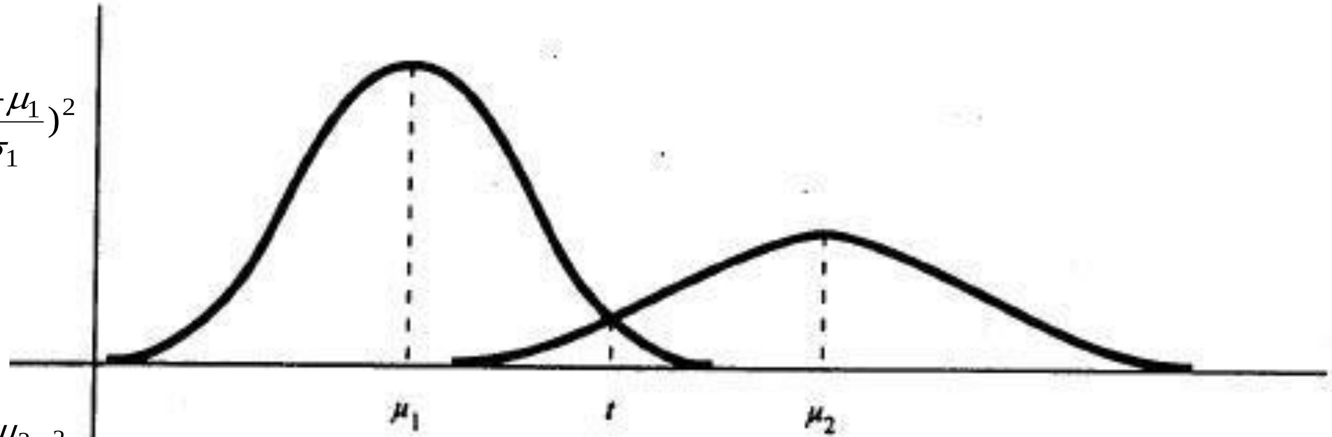
mixture distribution of the two Gaussians in histogram:

$$f(i) = \frac{q_1}{\sigma_1\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{i-\mu_1}{\sigma_1}\right)^2} + \frac{q_2}{\sigma_2\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{i-\mu_2}{\sigma_2}\right)^2}$$



Minimizing Kullback Information Distance

$$f_1(x) = \frac{q_1}{\sigma_1\sqrt{2\pi}} e^{-\frac{1}{2}(\frac{x-\mu_1}{\sigma_1})^2}$$



$$f_2(x) = \frac{q_2}{\sigma_2\sqrt{2\pi}} e^{-\frac{1}{2}(\frac{x-\mu_2}{\sigma_2})^2}$$

Figure 2.12 Mixture of two Gaussians. If an assignment is made to background pixel whenever $x < t$ and an assignment is made to foreground pixel whenever $x > t$, the probability of observing an $x < t$ and classifying it as background will be the area under f_1 to the left of t. The probability of observing an $x > t$ and classifying it as foreground will be the area under f_2 to the right of t.

μ : mean of distribution

Minimizing Kullback Information Distance

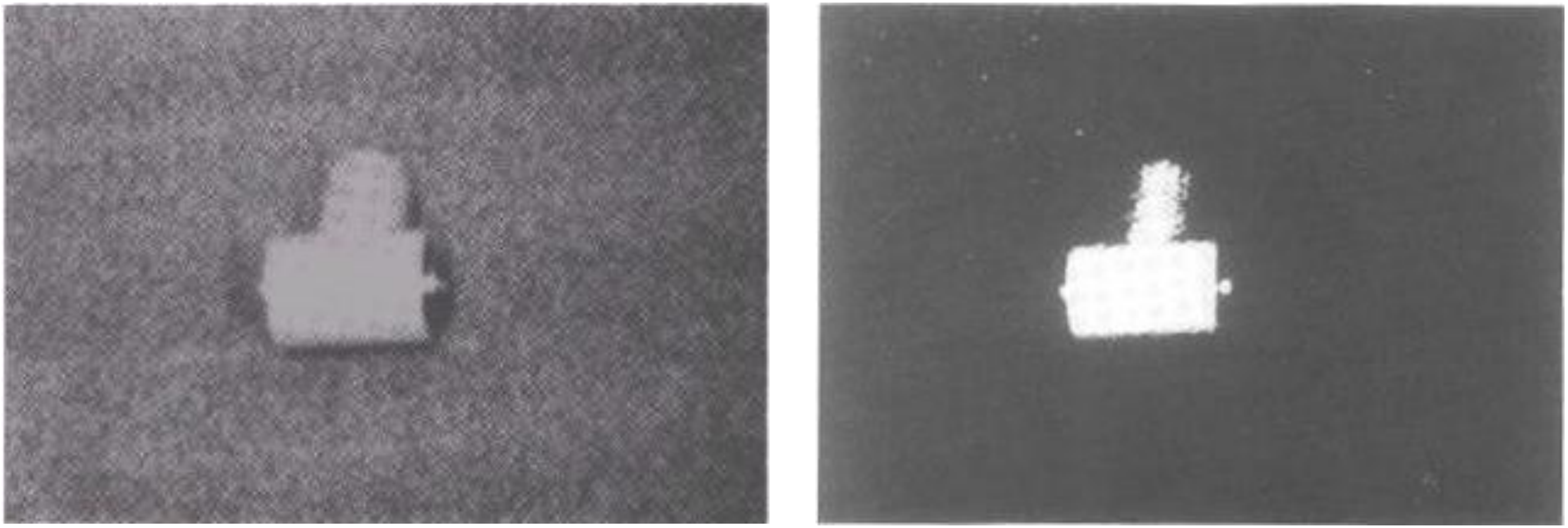


Figure 2.10 Binary image produced by thresholding the T-connector image of Fig. 2.6 with the Kittler-Illingworth threshold.

Minimizing Kullback Information Distance

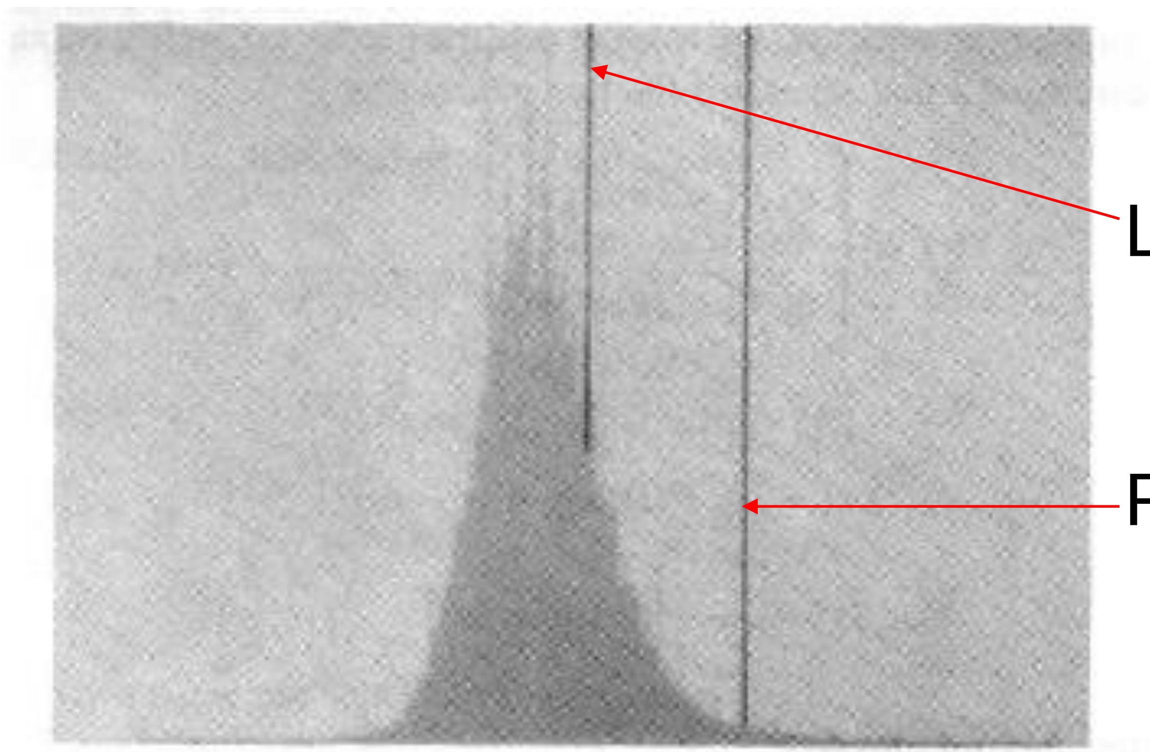


Figure 2.11 Histogram of the image of Fig. 2.6 showing where the Otsu and Kittler-Illingworth techniques choose the threshold value. The leftmost dark line is the Otsu threshold. The rightmost dark line is the Kittler-Illingworth threshold.

Left dark line:
Otsu

Right dark line:
Kittler-Illingworth

Signature Segmentation

- Signature: histogram of the nonzero pixels of the resulting masked image
- Signature: a projection
- Projections can be vertical, horizontal, diagonal, circular, radial…

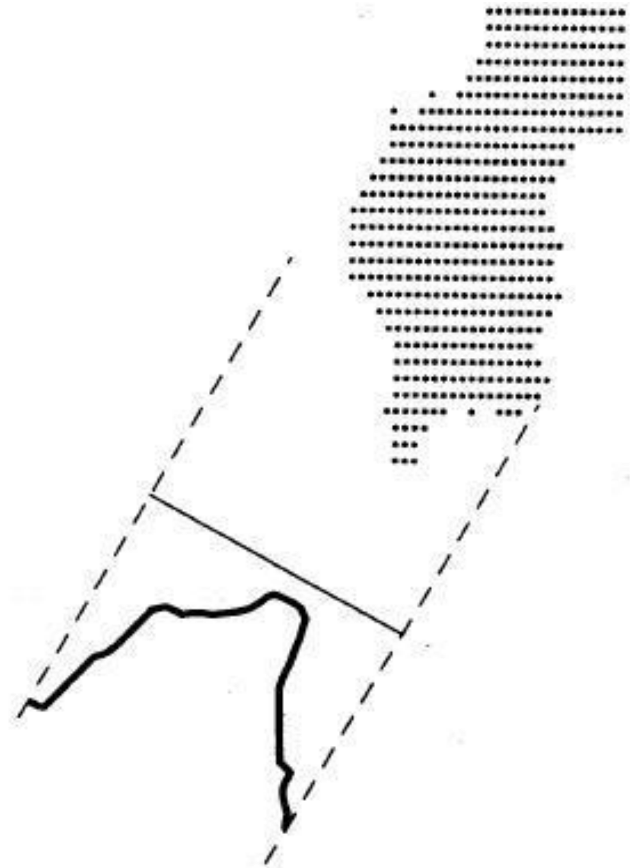


Figure 2.23 Diagonal projection of a shape. The direction of the projection is 45° counterclockwise from the column axis.

Signature Segmentation

- vertical projection of a segment:

  column c between s, t

- horizontal projection: row r between u, v

- vertical and horizontal projection define a rectangle

$$R = \{(r, c) \mid u \le r \le v \text{ and } s \le c \le t\}$$

Signature Segmentation

								1	1	1				3
									1	1	1			3
		1								1	1	1		4
		1									1	1	1	4
		1				1								2
	1					1								2
	1						1							2
1							1							2
								1						1
														0
	1	1							1	1	1	1		6
	1	1							1	1	1	1		6
	1	1							1	1	1	1		6
1	5	6	0	0	0	2	2	2	5	6	6	5	1	

Figure 2.25 Horizontal projection mask.

Signature Segmentation

	1	2	3
0			
0			
1			1
1			1
1			1
1		1	
1		1	
1	1		
0			

2	2	2	2	3	3	2	1	
		1	1	1				3
			1	1	1			3
				1	1	1		3
					1	1	1	3
1								1
1								1
	1							1
	1							1
		1						1

2		1	1
2		1	1
2		1	1
	0	3	3

			1	1	1	1		4
			1	1	1	1		4
			1	1	1	1		4
0	0	0	3	3	3	3	0	

Figure 2.27 Binary image segmented into regions on the basis of the segmentation of the initial vertical and horizontal projections. Also shown are the vertical and horizontal projections of each region.

Signature Segmentation

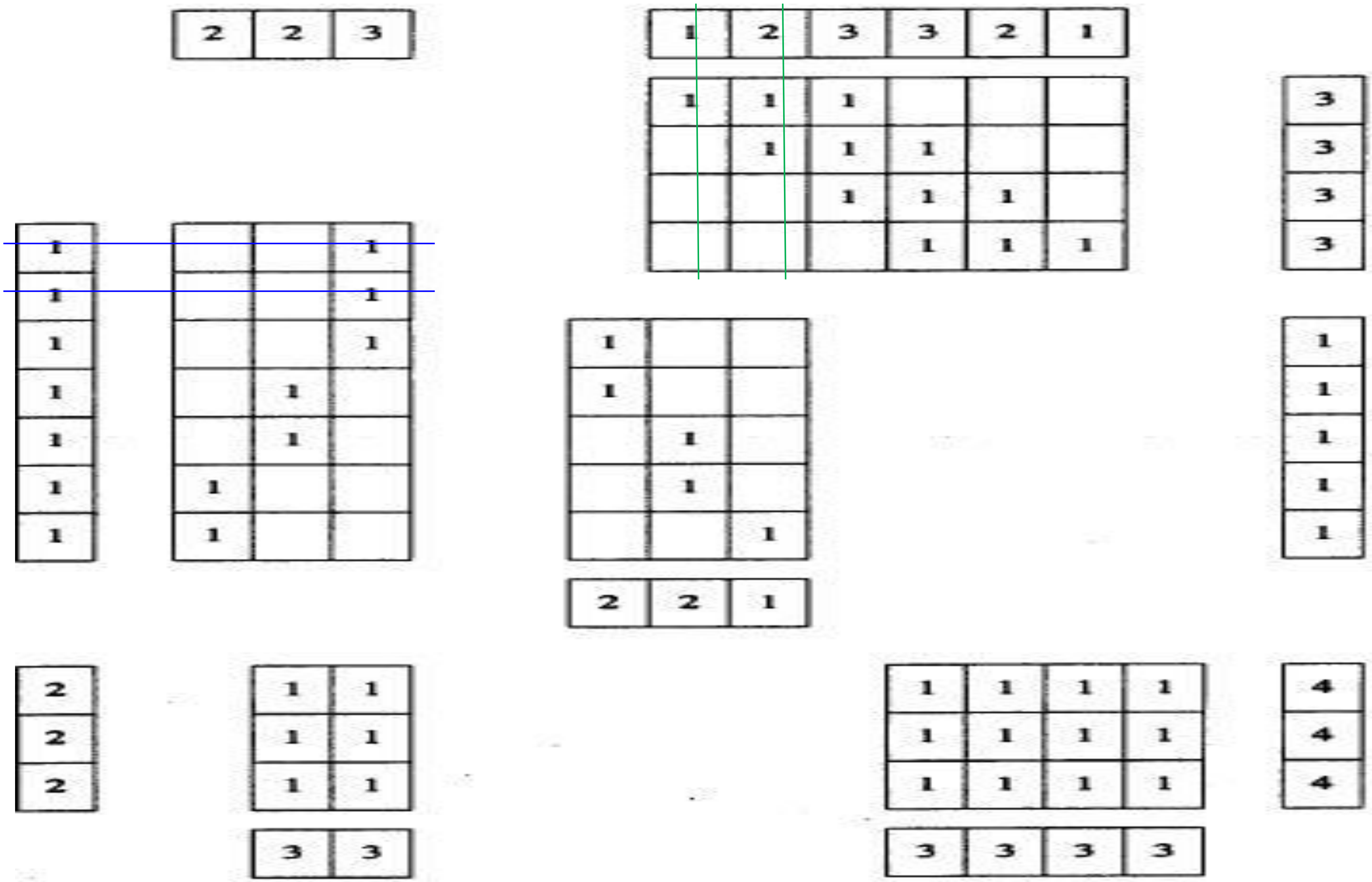
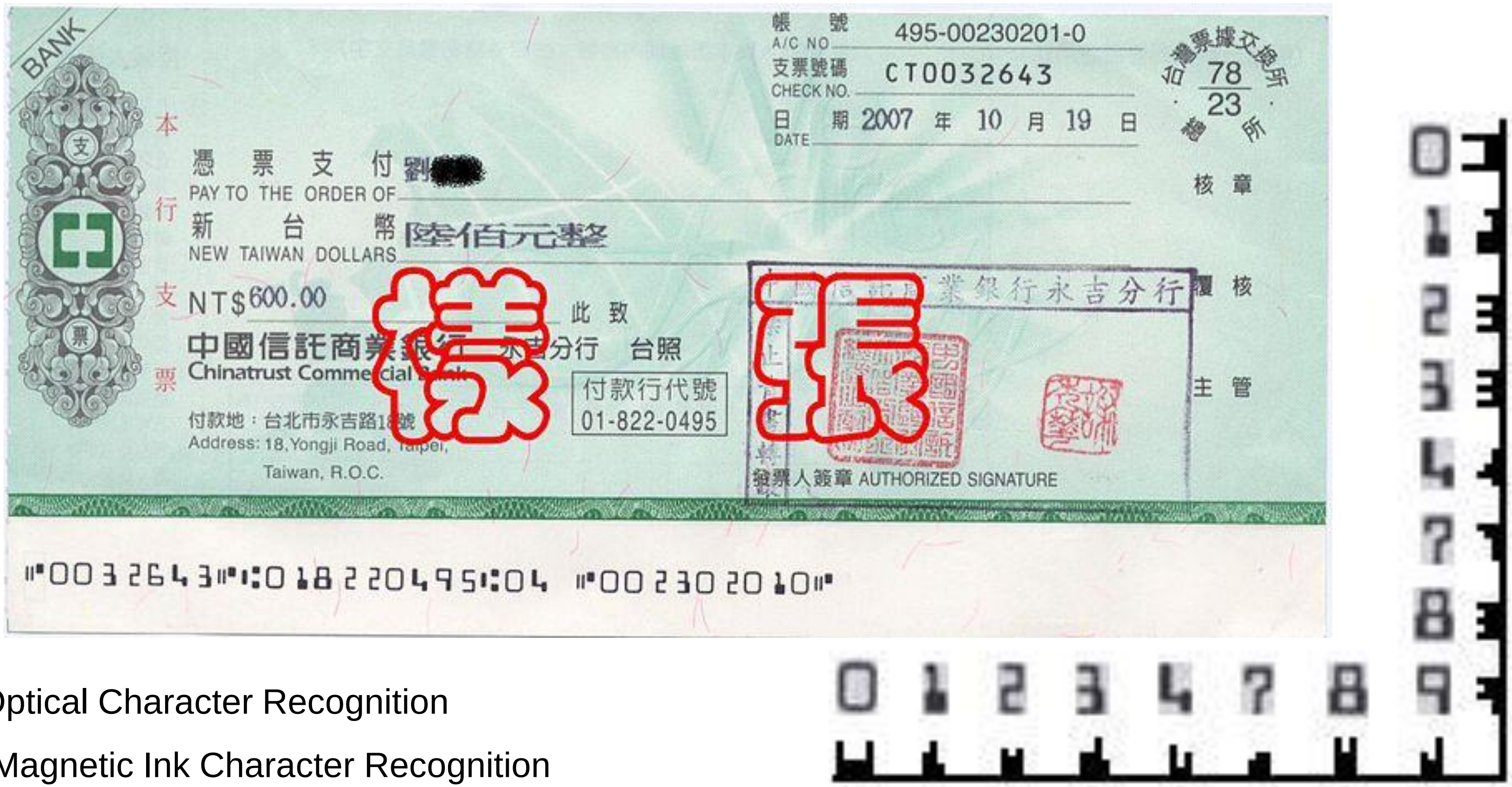


Figure 2.28 Binary image segmented into regions on the basis of the segmentation of Fig. 2.27. Also shown are the vertical and horizontal projections of each region.

Signature Segmentation

- Segment the vertical and horizontal projections
- Treat each rectangular subimage as the image



OCR: Optical Character Recognition

MICR: Magnetic Ink Character Recognition

Signature Segmentation

Diagonal projections:

- P_D : from upper left to lower right
- P_E : from upper right to lower left
- object area: sum of all the projections values in the segment

When

- components spaced away and relatively few, use signature segmentation

Signature Segmentation

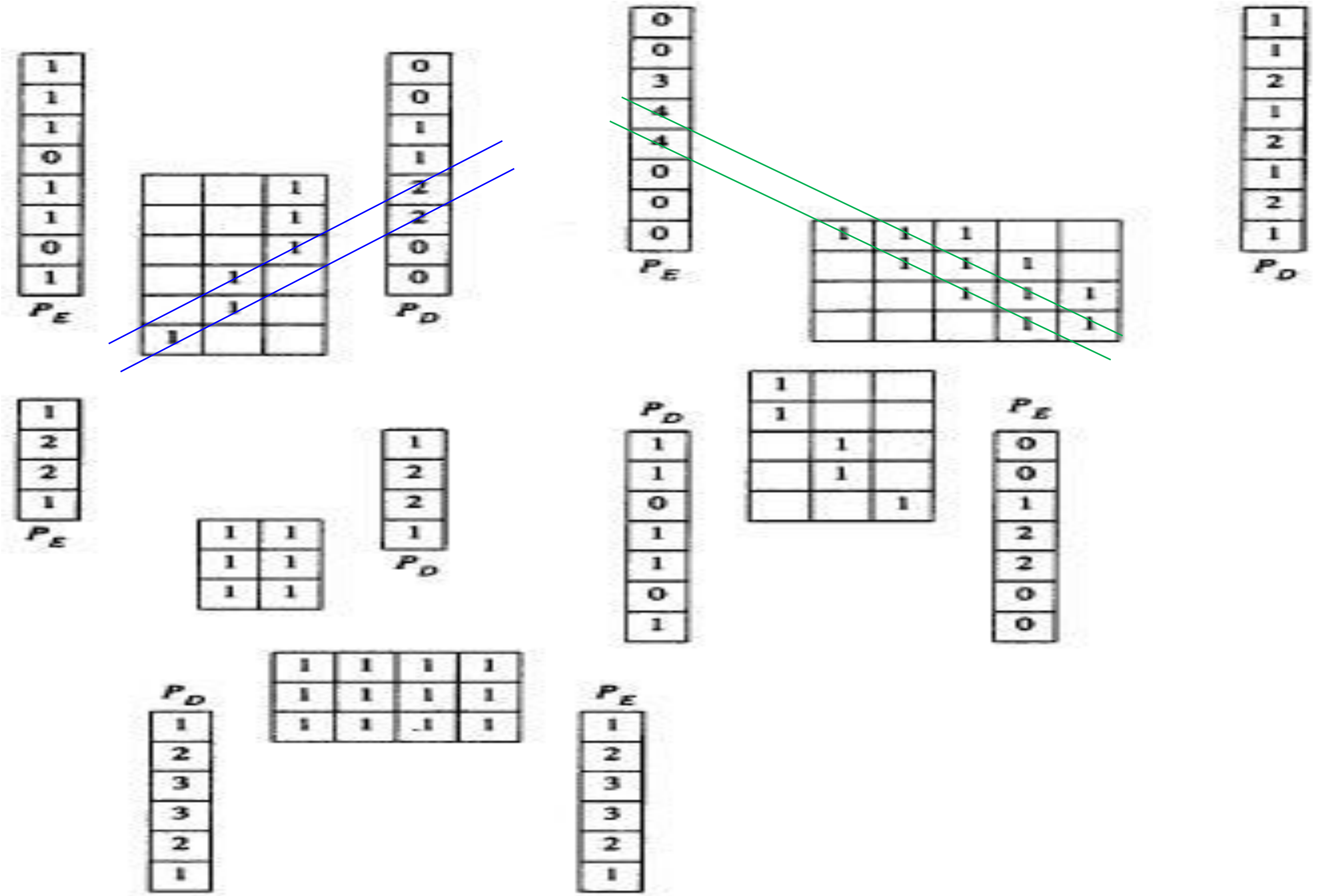


Figure 2.29 Diagonal projections P_D and P_E for each of the five image regions of Fig. 2.28. P_D is the diagonal projection taken 45° clockwise from the horizontal. P_E is the diagonal projection taken 135° clockwise from the horizontal.

Connected Components Labeling

- Connected components analysis of a binary image
  - connected components labeling of the binary-1 pixels
  - followed by property measurement of the component regions and decision making

- All pixels that have value binary-1 and are connected to each other by a path of pixels all with value binary-1 are given the same identifying label.

Connected Components Labeling

- Label
  - unique name or index of the region
  - identifier for a potential object region
- Connected components labeling: a grouping operation
- Pixel property: position, gray level or brightness level
- Region property: shape, bounding box, position, intensity statistics

Connected Components Operators

Two 1-pixels p and q belong to the same connected component C if there is a sequence of 1-pixels $(p_0, p_1, \ldots, p_n)$ of C where $p_0 = p, p_n = q$ and p_i is a neighbor of p_{i-1} for $i = 1, \ldots, n$

0	1	1	0	1	0	0
0	1	1	0	1	0	1
1	1	1	0	1	0	1
0	0	0	0	1	1	1
0	1	0	0	0	0	0
0	1	1	1	1	1	0
0	1	1	1	0	0	0

Figure 2.14 (a) Original image

0	1	1	0	2	0	0
0	1	1	0	2	0	2
1	1	1	0	2	0	2
0	0	0	0	2	2	2
0	3	0	0	0	0	0
0	3	3	3	3	3	0
0	3	3	3	0	0	0

(b) connected components

Connected Components Algorithms

- All the algorithms process a row of the image at a time
- All the algorithms assign new labels to the first pixel of each component
  - And propagate the label of a pixel to right or below neighbors

0	0	1	0	0	0	1
0	0	1	1	0	0	1
0	0	1	1	1	0	1
0	0	1	1	1	1	1

(a)

0	0	1	0	0	0	2
0	0	1	1	0	0	2
0	0	1	1	1	0	2
0	0	1	1	1	1	A

(b)

Figure 2.16 Propagation process. Label 1 has been propagated from the left to reach pixel A. Label 2 has been propagated down to reach pixel A. The connected components algorithm must assign a label to A and make labels 1 and 2 equivalent. Part (a) shows the original binary image, and (b) the partially processed image.

Connected Components Algorithms

- This process continues until the pixel marked "A" in row 4 encountered

0	0	1	0	0	0	1
0	0	1	1	0	0	1
0	0	1	1	1	0	1
0	0	1	1	1	1	1

(a)

0	0	1	0	0	0	2
0	0	1	1	0	0	2
0	0	1	1	1	0	2
0	0	1	1	1	1	A

(b)

Figure 2.16 Propagation process. Label 1 has been propagated from the left to reach pixel *A*. Label 2 has been propagated down to reach pixel *A*. The connected components algorithm must assign a label to *A* and make labels 1 and 2 equivalent. Part (a) shows the original binary image, and (b) the partially processed image.

Connected Components Algorithms

0	0	1	0	0	0	2
0	0	1	1	0	0	2
0	0	1	1	1	0	2
0	0	1	1	1	1	A

- The differences among the algorithms of three types:
  - What label should be assigned to pixel A?
  - How to keep track of the equivalence of two or more labels?
  - How to use the equivalence information to complete processing?
- An Iterative Algorithm
  - initialization of each pixel to a unique label
  - iteration of top-down followed by bottom-up passes
  - until no change

Connected Components Algorithms

	1	1		1	1	
	1	1		1	1	
	1	1	1	1	1	

(a)

	1	2		3	4	
	5	6		7	8	
	9	10	11	12	13	

(b)

	1	1		3	3	
	1	1		3	3	
	1	1	1	1	1	

(c)

	1	1		1	1	
	1	1		1	1	
	1	1	1	1	1	

(d)

Figure 2.17 Iterative algorithm for connected components labeling. Part (a) shows the original binary image; (b) the results after initialization of each 1-pixel to a unique label; (c) the results after the first top-down pass, in which the value of each nonzero pixel is replaced by the minimum value of its nonzero neighbors in a recursive manner going from left to right and top to bottom; and (d) the results after the first bottom-up pass.

The Classical Algorithm

- Makes two passes but requires a large global table for equivalences
  - performs label propagation as above
  - when two different labels propagate to the same pixel, the <u>smaller label propagates</u> and <u>equivalence entered into table</u>

- Equivalence classes are found by transitive closure

- Second pass performs a translation

The Classical Algorithm

																					1
																					1
								1	1	1	1	1	1								1
								1	1												1
				1	1	1	1	1	1												1
							1	1	1			1									1
							1	1	1			1									1
							1	1	1	1	1	1	1								1
													1								1
													1								1
													1	1							1
									1			1	1	1							1
		1			1	1		1	1			1	1	1							1
1	1	1			1	1		1					1	1							1
		1			1	1		1			1	1	1								1
		1			1	1		1				1	1								1
		1			1	1		1					1								1
1	1	1	1	1	1	1	1	1	1				1								1
1													1								1
1													1								1
1					1	1	1	1	1	1	1	1	1								1
1																					1
1																					1
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Figure 2.18 Classical connected components labeling algorithm: (a) shows the initial binary image

The Classical Algorithm

																		1	1	1	1
																					1
								2	2	2	2	2	2								1
								2	2												1
				3	3	3	3	2	2												1
							3	2	2			4									1
							3	2	2			4									1
							3	2	2	2	2	2	2								1
													2								1
													2								1
													2	2							1
									5			6	2	2							1
		7			8	8		9	5			6	2	2							1
10	10	7			8	8		9					2	2							1
		7			8	8		9			11	11	2								1
		7			8	8		9				11	2								1
		7			8	8		9					2								1
12	12	7	7	7	7	7	7	7	7				2								1
12													2								1
12													2								1
12					13	13	13	13	13	13	13	13	2								1
12																					1
12																					1
12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	1

Figure 2.18 Classical connected components labeling algorithm: (b) the labeling after the first top-down pass of the algorithm. The equivalence classes found are 1: { 1,12,7,8,9,10,5 } and 2: { 2,3,4,6,11,13 }

A Space-Efficient Two-Pass Algorithm

- The Classical Algorithm
  - main problem: global equivalence table may be too large for memory

- A space-efficient two-pass algorithm
  - Uses a local equivalence table
  - Small table stores only equivalences from current and preceding lines
  - Maximum number of equivalences = image width
  - Relabel each line with equivalence labels when equivalence detected

A Space-Efficient Two-Pass Algorithm

																		1	1	1	1
																					1
								2	2	2	2	2	2								1
								2	2												1
				2	2	2	2	2	2												1
							2	2	2			4									1
							2	2	2			4									1
							2	2	2	2	2	2	2								1
													2								1
													2								1
													2	2							1
									5			2	2	2							1
		7			8	8		5	5			2	2	2							1
7	7	7			8	8		5					2	2							1
		7			8	8		5			2	2	2								1
		7			8	8		5				2	2								1
		7			8	8		5					2								1
5	5	5	5	5	5	5	5	5	5				2								1
5													2								1
5													2								1
5					2	2	2	2	2	2	2	2	2								1
5																					1
5																					1
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Figure 2.19 Results after the top-down pass of the local table method on the binary image of Fig. 2.18(a). Note that on the lines where equivalences were detected, the pixels have different labels from those they had after pass 1 of the classical algorithm. For example, on line 5 the four leading 3s were changed to 2s on the second scan of that line, after the equivalence of labels 2 and 3 was detected. The bottom up pass will now propagate the label 1 to all pixels of the single connected component.

An Efficient Run-Length Implementation of the Local Table Method

- Run-length encoding
  - Transmits lengths of runs of zeros and ones

Example: $01111000110000 \rightarrow [1,4,3,2,4]$

run 1

run 2

	1	2	3	4	5
1	1	1		1	1
2	1	1			1
3	1	1	1		1
4					
5		1	1	1	1

(a)

run 7

	ROW_START	ROW_END
1	1	2
2	3	4
3	5	6
4	0	0
5	7	7

(b)

run number

run 1

run 2

	1	2	3	4	5
1	1	1		1	1
2	1	1			1
3	1	1	1		1
4					
5		1	1	1	1

(a)

run 7

	ROW_START	ROW_END
1	1	2
2	3	4
3	5	6
4	0	0
5	7	7

(b)

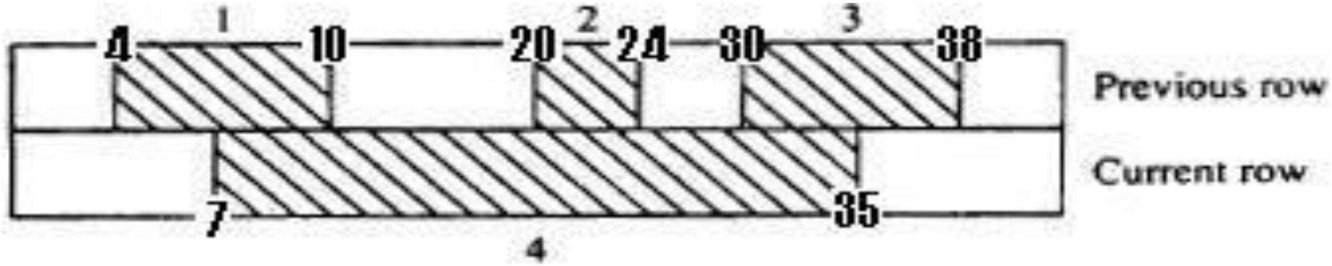
run number

run number

	ROW	START_COL	END_COL	PERM_LABEL
1	1	1	2	0
2	1	4	5	0
3	2	1	2	0
4	2	5	5	0
5	3	1	3	0
6	3	5	5	0
7	5	2	5	0

(c)

Figure 2.20 Binary image (a) and its run-length encoding (b) and (c). Each run of 1-pixels is encoded by its row (ROW) and the columns of its starting and ending pixels (START_COL and END_COL). In addition, for each row of the image, ROW_START points to the first run of the row and ROW_END points to the last run of the row. The PERM_LABEL field will hold the component label of the run; it is initialized to zero.



	ROW	START_COL	END_COL	PERM_LABEL
1	1	4	10	1
2	1	20	24	2
3	1	30	38	3
4	2	7	35	1

	LABEL	NEXT
1	1	3
2	1	3
3	1	0

	EQ_CLASS
1	1

Figure 2.21 Data structures used for keeping track of equivalence classes. In this example, run 4 has PERM_LABEL 1, which is an index into the LABEL array, that gives the equivalence class label for each possible PERM_LABEL value. In the example, PERM_LABELS 1,2, and 3 have all been determined to be equivalent, so LABEL(1), LABEL(2), and LABEL(3) all contain the equivalence class label, which is 1. Furthermore, the equivalence class label is an index into the EQ_CLASS array that contains pointers to the beginnings of the equivalence classes that are linked lists in the LABEL/NEXT structure. In this example there is only one equivalence class, class 1, and three elements of the LABEL/NEXT array are linked together to form this class.

Binary Machine Vision: Region Analysis

- Regions

  produced by connected components labeling operator

- Region properties

  to store as a measurement vector input to classifier

  - Region intensity histogram: gray level values for all pixels
  - Mean gray level value: summary statistics of regions intensity

Region Properties

- Bounding rectangle: smallest rectangle circumscribes the region

- Area:

$$A = \sum_{(r,c)\in R} 1$$

- Centroid:

$$\bar{r} = \frac{1}{A} \sum_{(r,c)\in R} r$$

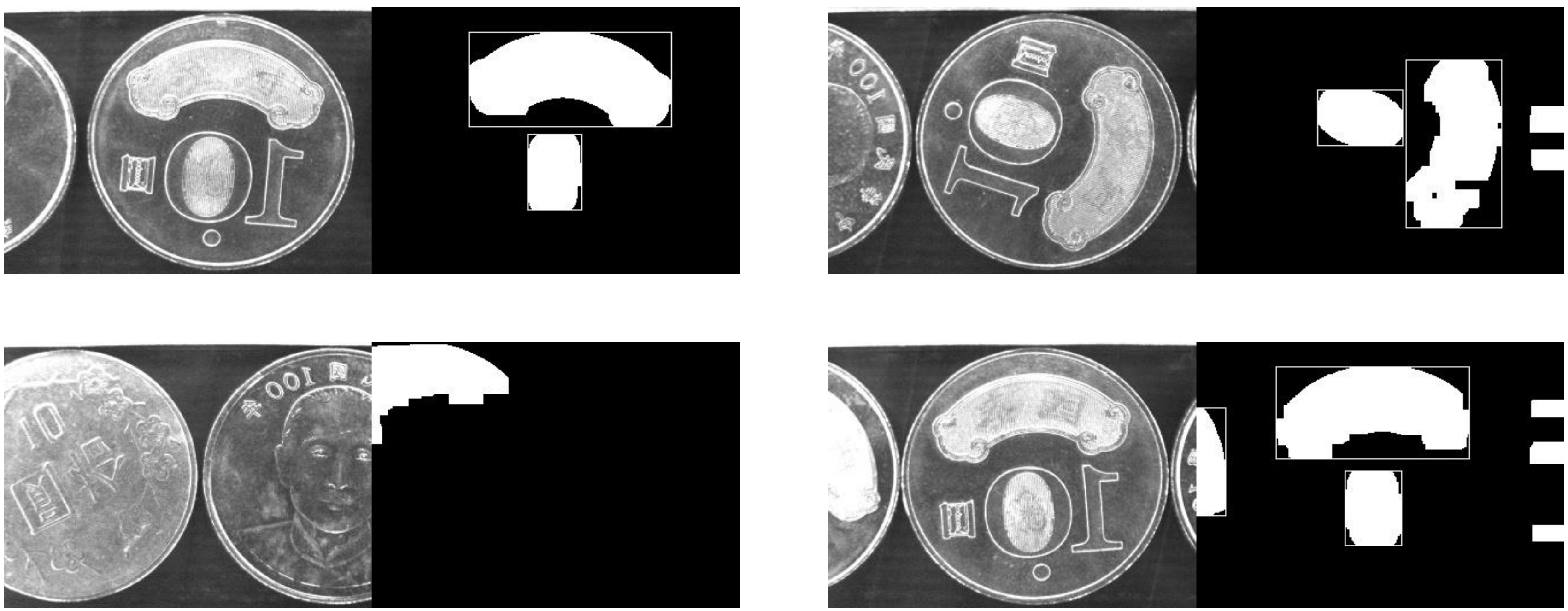
$$\bar{c} = \frac{1}{A} \sum_{(r,c)\in R} c$$

0	0	0	0	0	0	0	0
0	0	0	0	0	0	1	0
0	0	1	1	1	1	1	0
0	0	1	1	1	0	0	0
0	0	1	1	1	0	0	0
0	1	1	1	1	1	0	0
0	1	1	0	0	1	1	0
0	0	0	0	0	0	0	0

A=21
r=3.476
c=4.095

Application

- Recognition by
  - Area
  - Bounding box, width/height ratio
  - Centroid position



Extremal Points

- Eight distinct extremal points

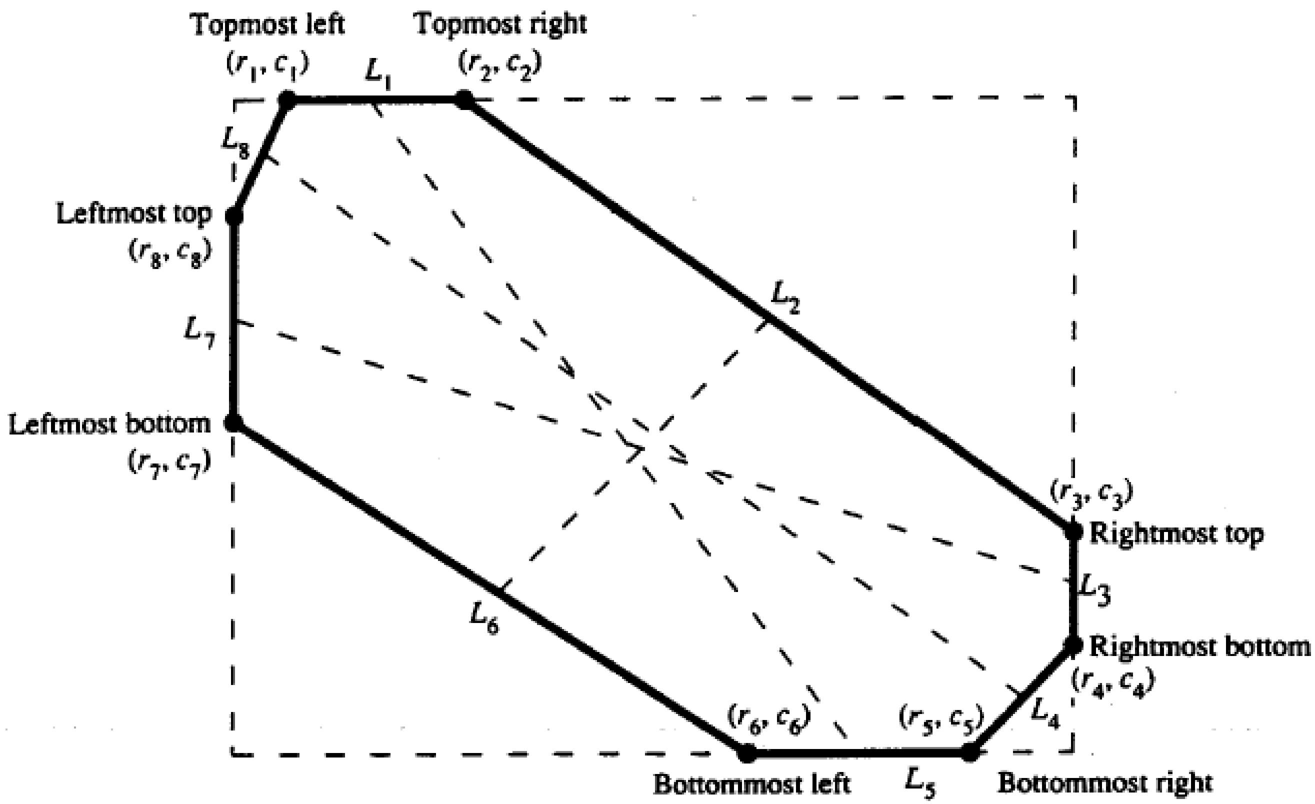


Figure 3.1 The eight extremal points a region can have and the normally oriented bounding rectangle that encloses the region. The interior dotted lines pair together opposite sides.

$$r_1 = r_2 = rmin$$
$$c_1 = \min\{c|(rmin, c) \in R\}$$
$$c_2 = \max\{c|(rmin, c) \in R\}$$
$$r_3 = \min\{r|(r, cmax) \in R\}$$
$$r_4 = \max\{r|(r, cmax) \in R\}$$
$$c_3 = c_4 = cmax$$

$$r_5 = r_6 = rmax$$
$$c_5 = \max\{c|(rmax, c) \in R\}$$
$$c_6 = \min\{c|(rmax, c) \in R\}$$
$$r_7 = \max\{r|(r, cmin) \in R\}$$
$$r_8 = \min\{r|(r, cmin) \in R\}$$
$$c_7 = c_8 = cmin$$

Extremal Points

- Different extremal points may be coincident

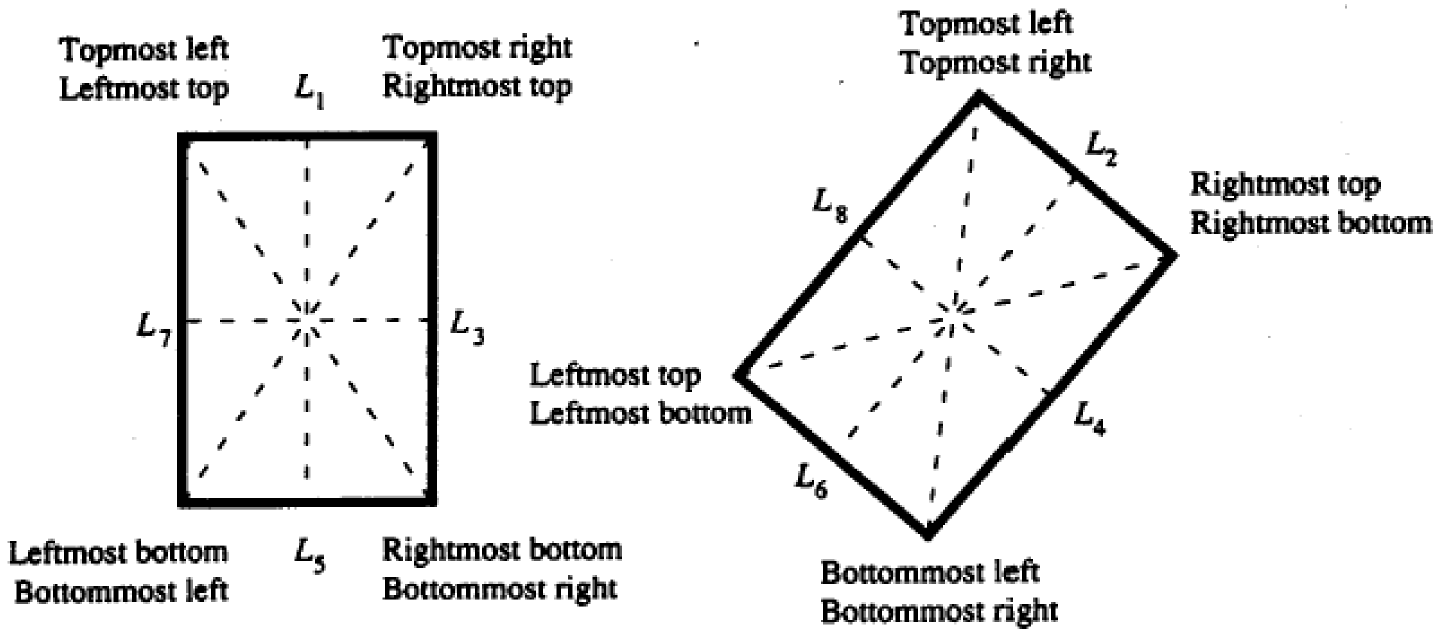


Figure 3.2 Two regions in which the extremal points are not unique and in which they pair differently. The interior dotted lines pair together opposite sides. Because some extremal points are coincident, some opposite sides have zero length.

Extremal Points

- Axes paired: M_1 with M_3 and M_2 with M_4
  - Distance calculation by Euclidean distance

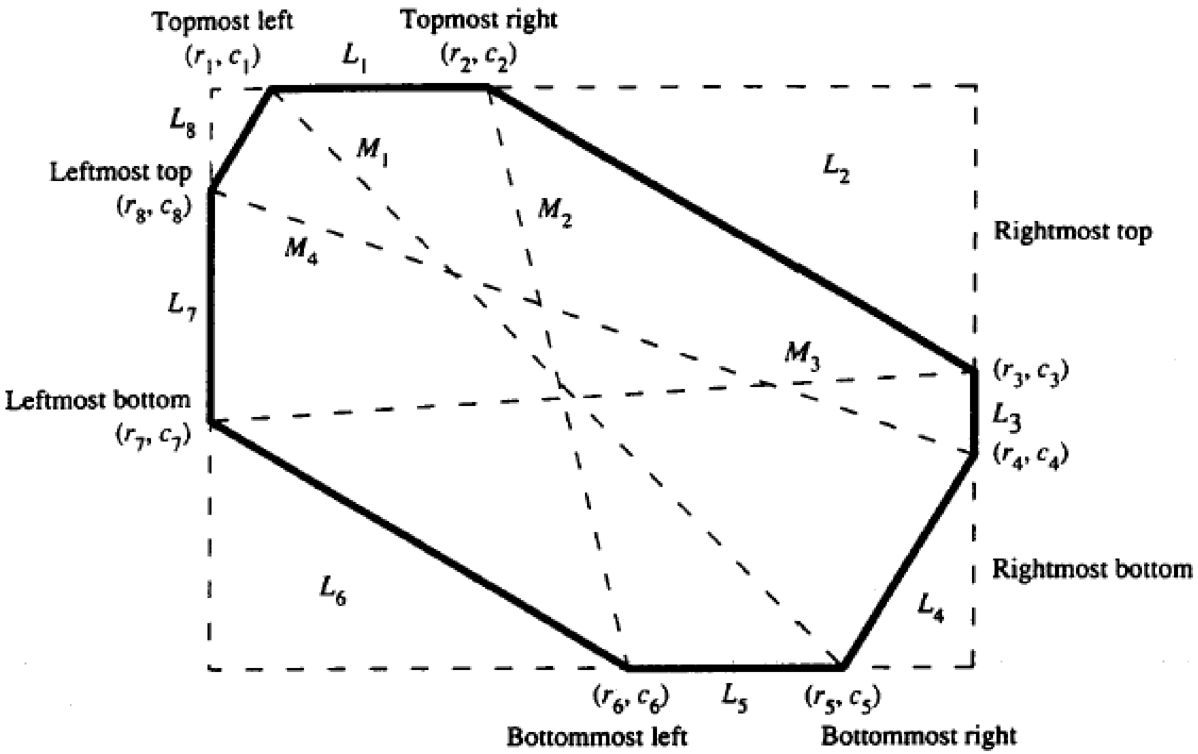
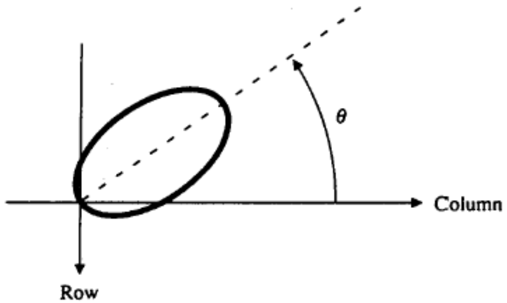


Figure 3.4 Orientation convention for the axes. The orientation angle of an axis is measured counterclockwise from the column axis.

- Orientation



$$\phi_1 = \tan^{-1} \frac{r_1 - r_5}{-(c_1 - c_5)}$$

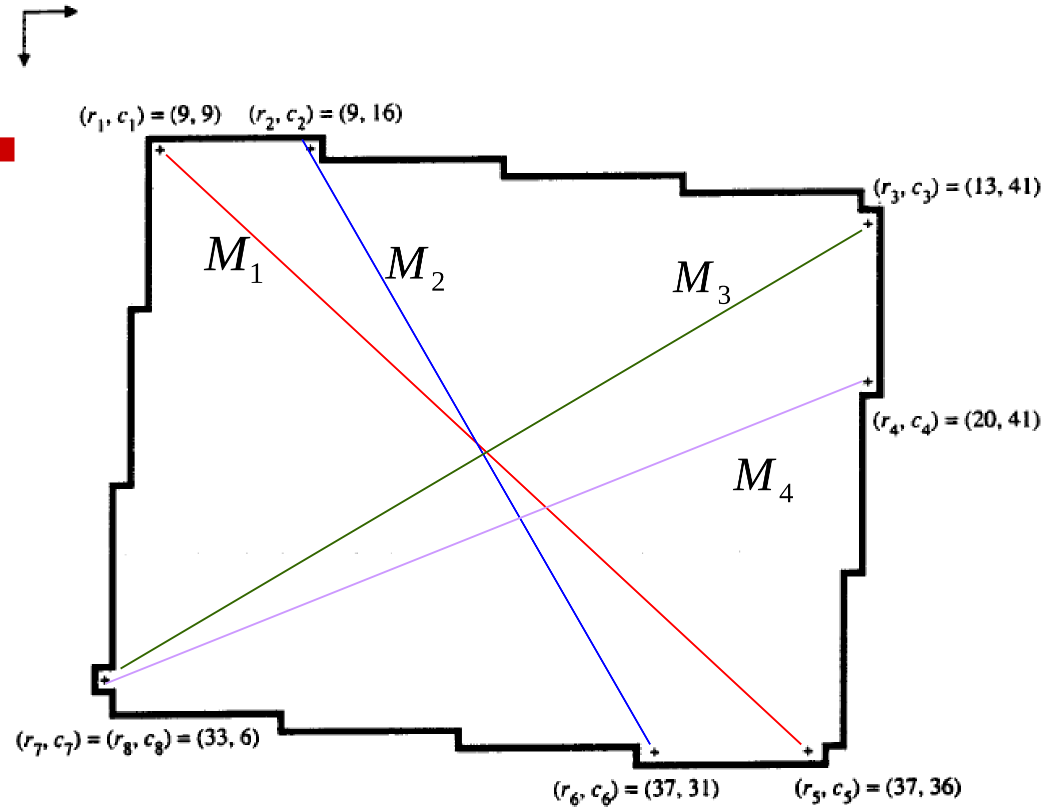
$$\phi_2 = \tan^{-1} \frac{r_2 - r_6}{-(c_2 - c_6)}$$

$$\phi_3 = \tan^{-1} \frac{r_3 - r_7}{-(c_3 - c_7)}$$

$$\phi_4 = \tan^{-1} \frac{r_4 - r_8}{-(c_4 - c_8)}$$

Extremal Points

- Example



$$M_1 = \sqrt{(9-37)^2 + (9-38)^2} + 1.12 = 41.43$$

$$M_2 = \sqrt{(9-37)^2 + (16-31)^2} + 1.12 = 32.88$$

$$M_3 = \sqrt{(13-33)^2 + (41-6)^2} + 1.12 = 41.43$$

$$M_4 = \sqrt{(20-33)^2 + (41-6)^2} + 1.12 = 38.46$$

$$M_{(1)} = M_1$$

$$M_{m(1)} - M_3$$

$$\phi_{(1)} = \tan^{-1} \frac{r_1 - r_5}{-(c_1 - c_5)} = \tan^{-1} \frac{9-37}{-(9-38)} \quad \tan^{-1} \frac{-28}{29} = -43.99°$$

$$\phi_{m(1)} = \tan^{-1} \frac{r_3 - r_7}{-(c_3 - c_7)} = \tan^{-1} \frac{13-33}{-(41-6)} \quad \tan^{-1} \frac{-20}{-35} = -150.26°$$

$$\theta_R = \frac{\phi_{(1)} + \phi_{m(1)}}{2} + 90° = -7.13°$$

Spatial Moments

- Second-order row moment

$$\mu_{rr} = \frac{1}{A} \sum_{(r,c)\in R} (r - \bar{r})^2$$

- Second-order mixed moment

$$\mu_{rc} = \frac{1}{A} \sum_{(r,c)\in R} (r - \bar{r})(c - \bar{c})$$

- Second-order column moment

$$\mu_{cc} = \frac{1}{A} \sum_{(r,c)\in R} (c - \bar{c})^2$$

Mixed Spatial Gray Level Moments

- Region properties: position, extent, shape, gray level properties

- Second-order mixed gray level spatial moments

$$\mu_{rg} = \frac{1}{A} \sum_{(r,c) \in R} (r - \bar{r})[I(r,c) - \mu]$$

$$\mu_{cg} = \frac{1}{A} \sum_{(r,c) \in R} (c - \bar{c})[I(r,c) - \mu]$$

Signature Properties

- Signature analysis: important because of easy, fast implementation

- Vertical projection $P_V(c) = \#\{r|(r,c) \in R\}$

- Horizontal projection $P_H(r) = \#\{c|(r,c) \in R\}$

- Diagonal projection from lower left to upper right $P_D(d) = \#\{(r,c) \in R | r + c = d\}$

- Diagonal projection from upper left to lower right $P_E(e) = \#\{(r,c) \in R | r - c = e\}$

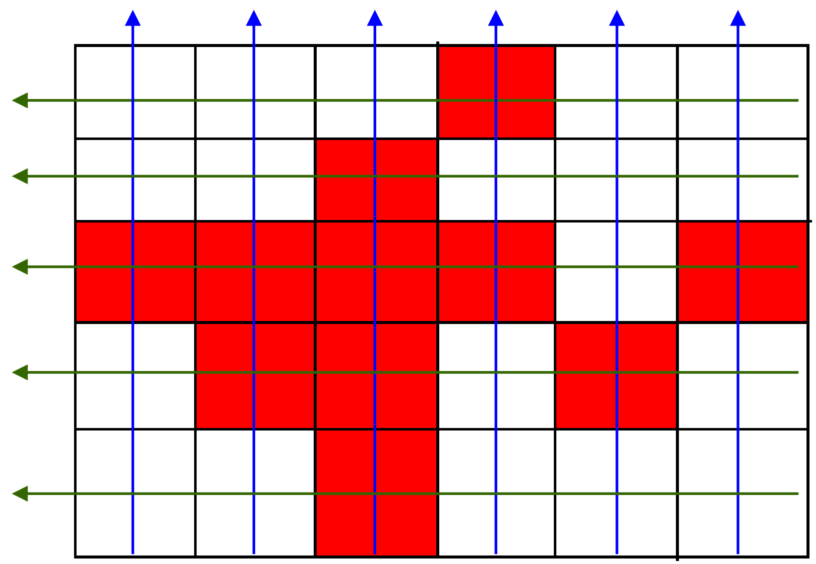
Signature Properties

- Example

$$P_V(c) = \#\{r|(r,c) \in R\}$$

$$P_V(0)=1, P_V(1)=2, P_V(2)=4, P_V(3)=2, P_V(4)=1, P_V(5)=1$$

$$P_H(0)=1$$
$$P_H(1)=1$$
$$P_H(2)=5$$
$$P_H(3)=3$$
$$P_H(4)=1$$



$$P_H(r) = \#\{c|(r,c) \in R\}$$

  - Area

$$A = \sum_{(r,c)\in R} 1 = \sum_r \sum_{\{c|(r,c)\in R\}} 1 = \sum_r P_H(r)$$

Signature Properties

- rmin: top row of bounding rectangle

$$rmin = \min\{r|(r,c) \in R\} = \min\{r|P_H(r) \neq 0\}$$

- rmax; bottom row of bounding rectangle

$$rmax = \max\{r|(r,c) \in R\} = \max\{r|P_H(r) \neq 0\}$$

- cmin: leftmost column of bounding rectangle

$$cmin = \min\{c|(r,c) \in R\} = \min\{c|P_V(c) \neq 0\}$$

- cmax: rightmost column of bounding rectangle

$$cmax = \max\{c|(r,c) \in R\} = \max\{c|P_V(c) \neq 0\}$$

Signature Properties

- Row centroid

$$\bar{r} = \frac{1}{A} \sum_{(r,c)\in R} r = \frac{1}{A} \sum_{r} \sum_{\{c|(r,c)\in R\}} r$$

$$= \frac{1}{A} \sum_{r} r \sum_{\{c|(r,c)\in R\}} 1 = \frac{1}{A} \sum_{r} r P_H(r)$$

- Column centroid

$$\bar{c} = \frac{1}{A} \sum_{(r,c)\in R} c = \frac{1}{A} \sum_{c} \sum_{\{r|(r,c)\in R\}} c$$

$$= \frac{1}{A} \sum_{c} c \sum_{\{r|(r,c)\in R\}} 1 = \frac{1}{A} \sum_{c} c P_V(c)$$

- Diagonal centroid

$$\bar{d} = \frac{1}{A} \sum_{d} d P_D(d)$$

- Another diagonal centroid

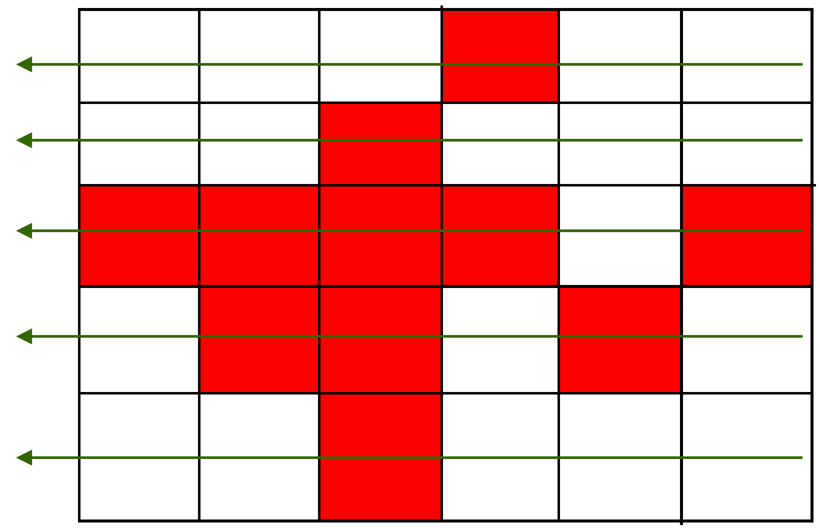
$$\bar{e} = \frac{1}{A} \sum_{e} e P_E(e)$$

Signature Properties

- Second row moment from horizontal projection

$$\mu_{rr} = \frac{1}{A}\sum_r (r-\bar{r})^2 P_H(r) \qquad P_H(r) = \#\{c \mid (r,c) \in R\}$$

$$(0-\bar{r})^2 * P_H(0) = (0-\bar{r})^2 * 1$$
$$(1-\bar{r})^2 * P_H(1) = (1-\bar{r})^2 * 1$$
$$(2-\bar{r})^2 * P_H(2) = (2-\bar{r})^2 * 5$$
$$(3-\bar{r})^2 * P_H(3) = (3-\bar{r})^2 * 3$$
$$(4-\bar{r})^2 * P_H(4) = (4-\bar{r})^2 * 1$$



- Second column moment from vertical projection

$$\mu_{cc} = \frac{1}{A}\sum_c (c-\bar{c})^2 P_V(c)$$

- Second diagonal moment $\mu_{dd} = \frac{1}{A}\sum_d (d-\bar{d})^2 P_D(d)$