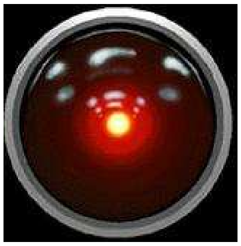


HCI/ComS 575X:
Computational Perception

Instructor: Alexander Stoytchev
http://www.cs.iastate.edu/~alex/classes/2007_Spring_575X/

Binary Image Processing

January 17, 2007

Readings

- Jain, Kasturi, and Schunck (1995). Machine Vision, ``Chapter 1: Introduction,'' McGraw-Hill, pp. 1-24.
- Jain, Kasturi, and Schunck (1995). Machine Vision, ``Chapter 2: Binary Image Processing,'' McGraw-Hill, pp. 25-72.

Reading for Next Lecture

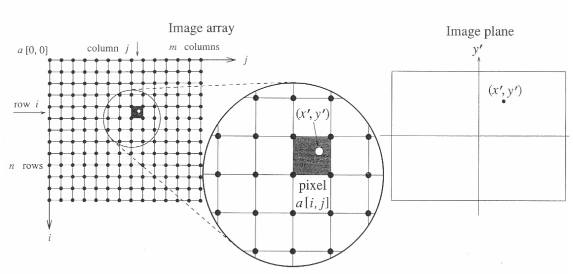
- Haralick and Shapiro (1993). Computer and Robot Vision, "Chapter 5: Mathematical Morphology," Addison-Wesley.

What is an image?

Intensity Levels

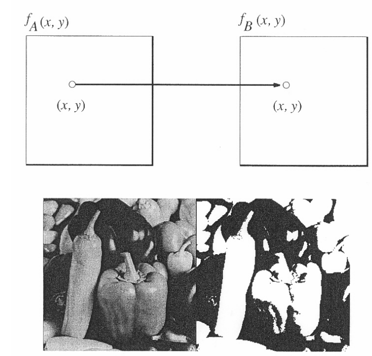
- 2
- 32
- 64
- 128
- 256 (8 bits)
- 512
- …
- 4096 (12 bits)

Image Plane v.s. Image Array



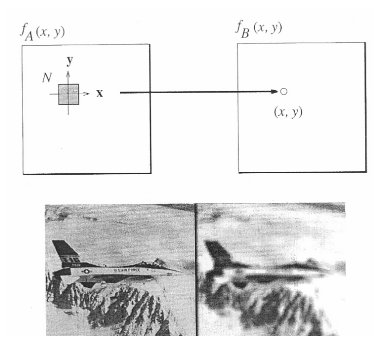
[Jain, Kasturi, and Schunck (1995). Machine Vision, Ch. 1]

Point Operations



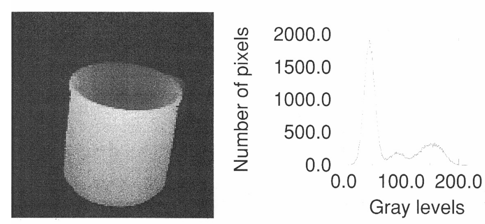
[Jain, Kasturi, and Schunck (1995). Machine Vision, Ch. 1]

Local Operations



[Jain, Kasturi, and Schunck (1995). Machine Vision, Ch. 1]

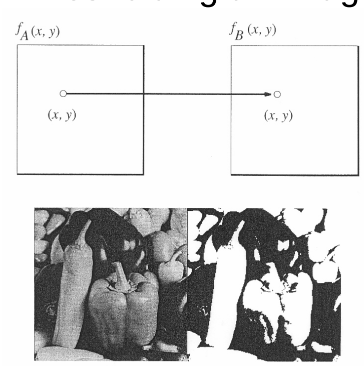
Global Operations



$$P = O_{\text{global}}\{f[i,j]\}$$

[Jain, Kasturi, and Schunck (1995). Machine Vision, Ch. 1]

Thresholding an Image



[Jain, Kasturi, and Schunck (1995). Machine Vision, Ch. 1]

Dark Image on a Light Background

$$F_T[i,j] = \begin{cases} 1 & \text{if } F[i,j] \le T \\ 0 & \text{otherwise.} \end{cases}$$

[Jain, Kasturi, and Schunck (1995). Machine Vision, Ch. 2]

Selecting a range of intensity values

$$F_T[i,j] = \begin{cases} 1 & \text{if } T_1 \le F[i,j] \le T_2 \\ 0 & \text{otherwise.} \end{cases}$$

[Jain, Kasturi, and Schunck (1995). Machine Vision, Ch. 2]

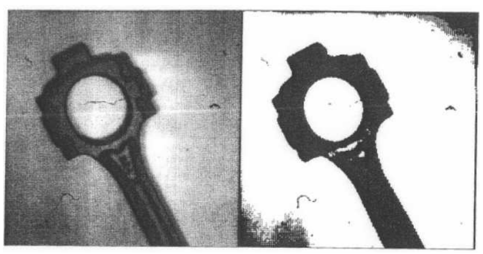
Generalized Thresholding

A general thresholding scheme in which the intensity levels for an object may come from several disjoint intervals may be represented as

$$F_T[i,j] = \begin{cases} 1 & \text{if } F[i,j] \in Z \\ 0 & \text{otherwise} \end{cases} \quad (2.4)$$

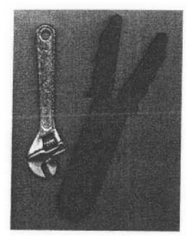
[Jain, Kasturi, and Schunck (1995). Machine Vision, Ch. 2]

Thresholding Example (1)



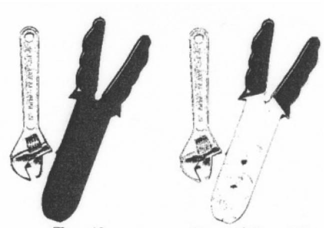
[Jain, Kasturi, and Schunck (1995). Machine Vision, Ch. 2]

Thresholding Example (2)



Original grayscale Image

[Jain, Kasturi, and Schunck (1995). Machine Vision, Ch. 2]



$T = 48$ $T_1 = 2 | T_2 = 48$

[Jain, Kasturi, and Schunck (1995). Machine Vision, Ch. 2]

$T = 21$ $T_1 = 135 | T_2 = 255$

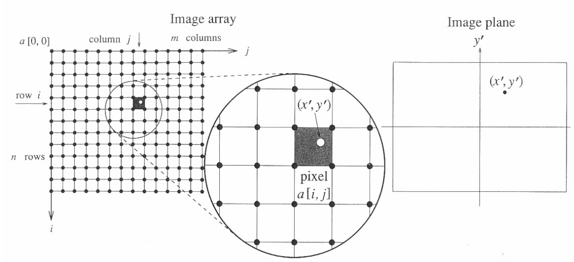
[Jain, Kasturi, and Schunck (1995). Machine Vision, Ch. 2]

Area of a Binary Image

$$A = \sum_{i=1}^{n} \sum_{j=1}^{m} B[i,j].$$

[Jain, Kasturi, and Schunck (1995). Machine Vision, Ch. 2]

This figure now becomes important



[Jain, Kasturi, and Schunck (1995). Machine Vision, Ch. 1]

Calculating the Position of an Object

$$\bar{x} \sum_{i=1}^{n} \sum_{j=1}^{m} B[i,j] = \sum_{i=1}^{n} \sum_{j=1}^{m} jB[i,j]$$

$$\bar{y} \sum_{i=1}^{n} \sum_{j=1}^{m} B[i,j] = \sum_{i=1}^{n} \sum_{j=1}^{m} iB[i,j]$$

[Jain, Kasturi, and Schunck (1995). Machine Vision, Ch. 2]

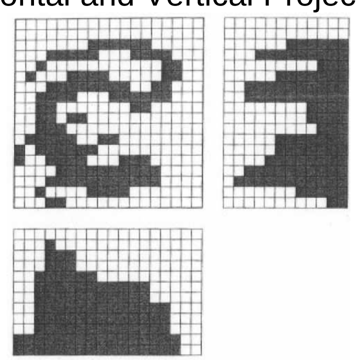
The center is given by

$$\bar{x} = \frac{\sum_{i=1}^{n} \sum_{j=1}^{m} jB[i,j]}{A}$$

$$\bar{y} = \frac{\sum_{i=1}^{n} \sum_{j=1}^{m} iB[i,j]}{A}.$$

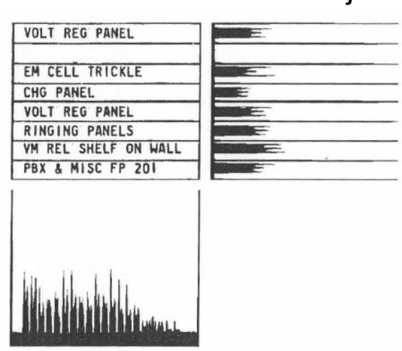
[Jain, Kasturi, and Schunck (1995). Machine Vision, Ch. 2]

Horizontal and Vertical Projections



[Jain, Kasturi, and Schunck (1995). Machine Vision, Ch. 2]

Horizontal and Vertical Projections



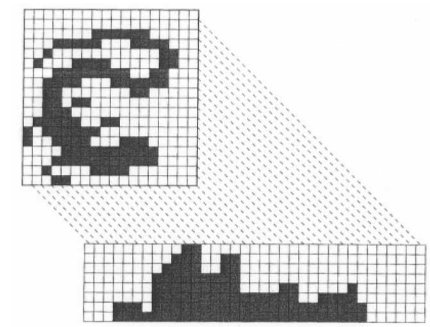
[Jain, Kasturi, and Schunck (1995). Machine Vision, Ch. 2]

Projection Formulas

$$H[i] = \sum_{j=1}^{m} B[i,j]$$
$$V[j] = \sum_{i=1}^{n} B[i,j].$$

[Jain, Kasturi, and Schunck (1995). Machine Vision, Ch. 2]

Diagonal Projection



[Jain, Kasturi, and Schunck (1995). Machine Vision, Ch. 2]

The area and the position can be computed form the H and V projections

$$A = \sum_{j=1}^{m} V[j] = \sum_{i=1}^{n} H[i]$$
$$\bar{y} = \frac{\sum_{i=1}^{n} i H[i]}{A}$$
$$\bar{x} = \frac{\sum_{j=1}^{m} j V[j]}{A}.$$

[Jain, Kasturi, and Schunck (1995). Machine Vision, Ch. 2]

Run-Length Encoding

Binary image:

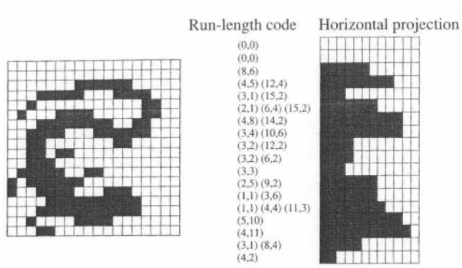
1	1	1	0	0	0	1	1	0	0	0	1	1	1	1	0	1	1	0	1	1	1
0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1	1	1	1
1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1

Start and length of 1 runs: $(1,3)$ $(7,2)$ $(12,4)$ $(17,2)$ $(20,3)$
$(5,13)$ $(19,4)$
$(1,3)$ $(17,6)$

Length of 1 and 0 runs: $3,3,2,3,4,1,2,1,3$
$0,4,13,1,4$
$3,13,6$

[Jain, Kasturi, and Schunck (1995). Machine Vision, Ch. 2]

Horizontal Projections Calculated from run-length code



[Jain, Kasturi, and Schunck (1995). Machine Vision, Ch. 2]

The area of an object can be obtained by summing the lengths of all 1 runs

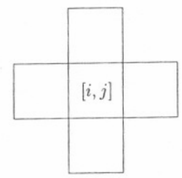
$$A = \sum_{i=0}^{n-1} \sum_{k=0}^{\left(\frac{m_i - 1}{2}\right)} r_{i,2k+1}$$

[Jain, Kasturi, and Schunck (1995). Machine Vision, Ch. 2]

Neighbors and Connectivity

4-Connected

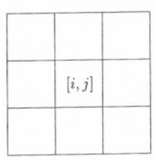
4-neighbors $[i+1,j]$, $[i-1,j]$, $[i,j+1]$, $[i,j-1]$



[Jain, Kasturi, and Schunck (1995). Machine Vision, Ch. 2]

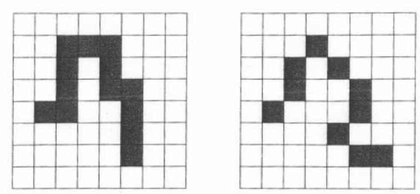
8-connected

8-neighbors $[i+1,j+1]$, $[i+1,j-1]$, $[i-1,j+1]$, $[i-1,j-1]$ plus all of the 4-neighbors



[Jain, Kasturi, and Schunck (1995). Machine Vision, Ch. 2]

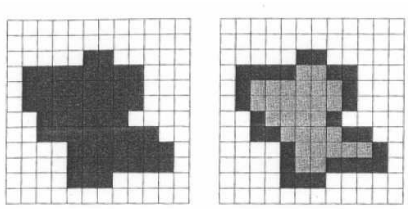
Examples of Paths



(a) 4-path (b) 8-path

[Jain, Kasturi, and Schunck (1995). Machine Vision, Ch. 2]

Boundary, Interior, and Background

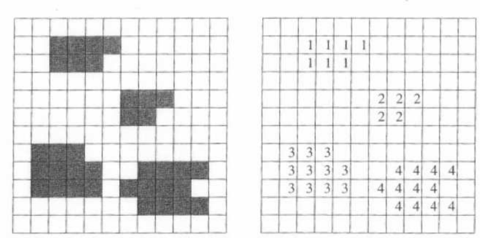


(a) Original image

(b) Boundary pixels; Interior pixels; Surrounds pixels

[Jain, Kasturi, and Schunck (1995). Machine Vision, Ch. 2]

An Image (a) and Its Connected Components (b)



(a) (b)

[Jain, Kasturi, and Schunck (1995). Machine Vision, Ch. 2]

Algorithm 2.1 Recursive Connected Components Algorithm

1. *Scan the image to find an unlabeled 1 pixel and assign it a new label L.*
2. *Recursively assign a label L to all its 1 neighbors.*
3. *Stop if there are no more unlabeled 1 pixels.*
4. *Go to step 1.*

[Jain, Kasturi, and Schunck (1995). Machine Vision, Ch. 2]

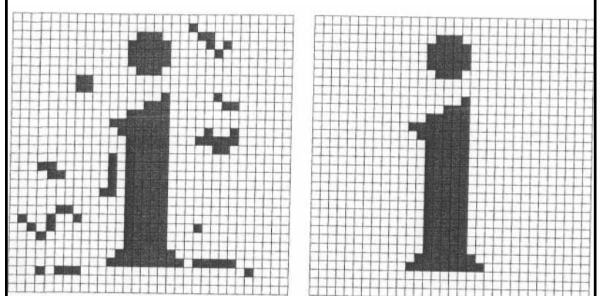
Algorithm 2.2 Sequential Connected Components Algorithm using 4-connectivity

1. *Scan the image left to right, top to bottom.*
2. *If the pixel is 1, then*
   - *(a) If only one of its upper and left neighbors has a label, then copy the label.*
   - *(b) If both have the same label, then copy the label.*
   - *(c) If both have different labels, then copy the upper's label and enter the labels in the equivalence table as equivalent labels.*
   - *(d) Otherwise assign a new label to this pixel and enter this label in the equivalence table.*
3. *If there are more pixels to consider, then go to step 2.*
4. *Find the lowest label for each equivalent set in the equivalence table.*
5. *Scan the picture. Replace each label by the lowest label in its equivalent set.*

[Jain, Kasturi, and Schunck (1995). Machine Vision, Ch. 2]

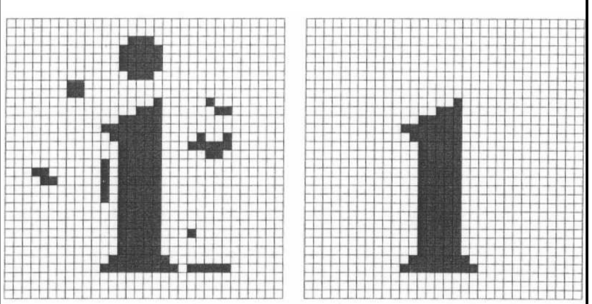
Thresholding by Size

Before and after a size filter (T=10)



[Jain, Kasturi, and Schunck (1995). Machine Vision, Ch. 2]

Before and after a size filter (T=25)



[Jain, Kasturi, and Schunck (1995). Machine Vision, Ch. 2]

Distance Metrics

Properties of a Good Distance Metrics

1. $d(p,q) \geq 0$ and $d(p,q) = 0$ iff $p = q$
2. $d(q,p) = d(p,q)$
3. $d(p,r) \leq d(p,q) + d(q,r)$

[Jain, Kasturi, and Schunck (1995). Machine Vision, Ch. 2]

Examples

Euclidean

$$d_{\text{Euclidean}}([i_1, j_1], [i_2, j_2]) = \sqrt{(i_1 - i_2)^2 + (j_1 - j_2)^2} \qquad (2.38)$$

City-block

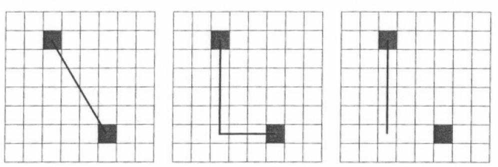
$$d_{\text{city}} = |i_1 - i_2| + |j_1 - j_2| \qquad (2.39)$$

Chessboard

$$d_{\text{chess}} = \max(|i_1 - i_2|, |j_1 - j_2|) \qquad (2.40)$$

[Jain, Kasturi, and Schunck (1995). Machine Vision, Ch. 2]

Examples (2)



(a) (b) (c)

Figure 2.17: Examples of (a) Euclidean, (b) city-block, and (c) chessboard distance measures.

[Jain, Kasturi, and Schunck (1995). Machine Vision, Ch. 2]

Euclidean Distance

			3			
	$\sqrt{8}$	$\sqrt{5}$	2	$\sqrt{5}$	$\sqrt{8}$	
	$\sqrt{5}$	$\sqrt{2}$	1	$\sqrt{2}$	$\sqrt{5}$	
3	2	1	0	1	2	3
	$\sqrt{5}$	$\sqrt{2}$	1	$\sqrt{2}$	$\sqrt{5}$	
	$\sqrt{8}$	$\sqrt{5}$	2	$\sqrt{5}$	$\sqrt{8}$	
			3			

[Jain, Kasturi, and Schunck (1995). Machine Vision, Ch. 2]

City-block Distance

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

[Jain, Kasturi, and Schunck (1995). Machine Vision, Ch. 2]

Chessboard distance

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

[Jain, Kasturi, and Schunck (1995). Machine Vision, Ch. 2]

Iterative Distance Transorms

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

Original

→

1	1	1	1	1	1
1	2	2	2	2	1
1	2	2	2	2	1
1	2	2	2	2	1
1	1	1	1	1	1

1-st iteration

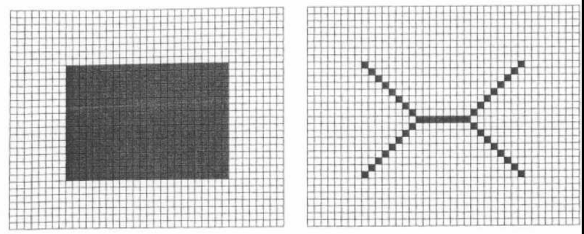
→

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

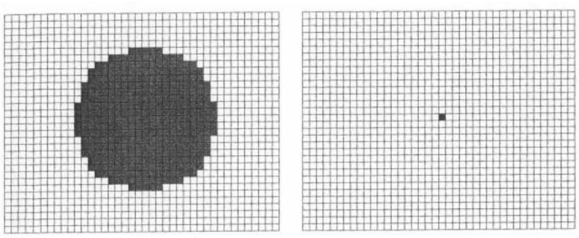
2-nd iteration

[Jain, Kasturi, and Schunck (1995). Machine Vision, Ch. 2]

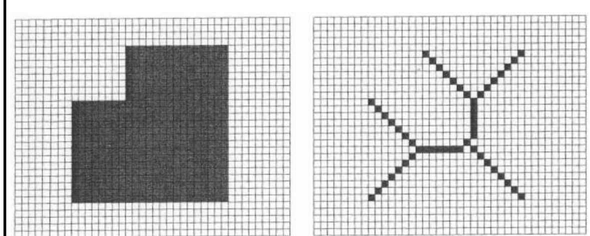
Medial Axis Example



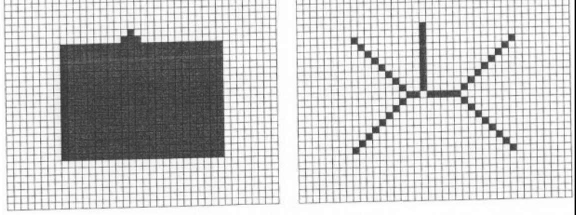
[Jain, Kasturi, and Schunck (1995). Machine Vision, Ch. 2]



[Jain, Kasturi, and Schunck (1995). Machine Vision, Ch. 2]

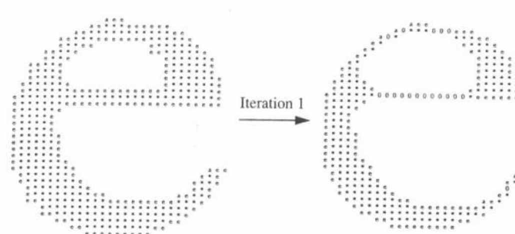


[Jain, Kasturi, and Schunck (1995). Machine Vision, Ch. 2]



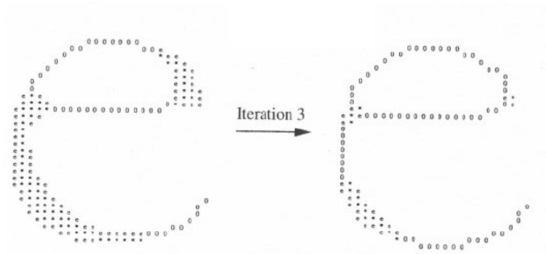
[Jain, Kasturi, and Schunck (1995). Machine Vision, Ch. 2]

Thinning



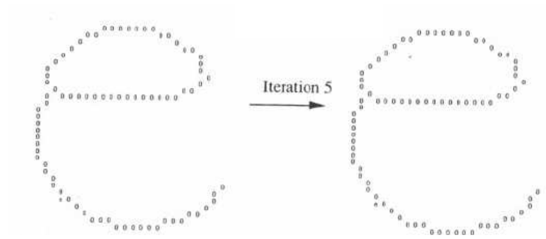
[Jain, Kasturi, and Schunck (1995). Machine Vision, Ch. 2]

Thinning

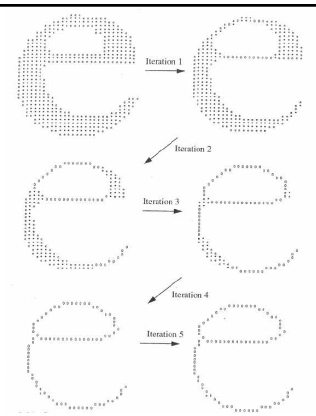


[Jain, Kasturi, and Schunck (1995). Machine Vision, Ch. 2]

Stopping Condition



[Jain, Kasturi, and Schunck (1995). Machine Vision, Ch. 2]



[Jain, Kasturi, and Schunck (1995). Machine Vision, Ch. 2]

Expanding and Shrining

- **Expanding:** change a pixel from 0 to 1 if any neighbors of the pixel are 1.
- **Shrinking:** change a pixel from 1 to 0 if any neighbors of the pixel are 0.

Expanding and Shrining

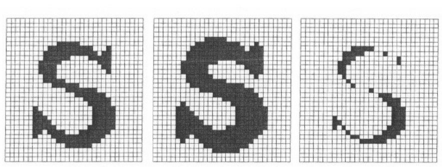


Figure 2.23: An example of expanding and shrinking operations on the letter "s." *Left*: The original image. *Middle*: Expanded image. *Right*: Shrunken image.

[Jain, Kasturi, and Schunck (1995). Machine Vision, Ch. 2]

Properties and Notation

$S^{(k)}$: S expanded k times
$S^{(-k)}$: S shrunk k times

the following properties hold:

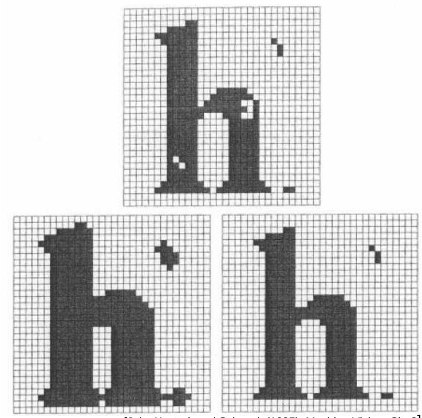
$$(S^m)^{-n} \neq (S^{-n})^m \neq S^{(m-n)}$$

$$S \subset (S^k)^{-k}$$
$$S \supset (S^{-k})^k$$

[Jain, Kasturi, and Schunck (1995). Machine Vision, Ch. 2]

Original

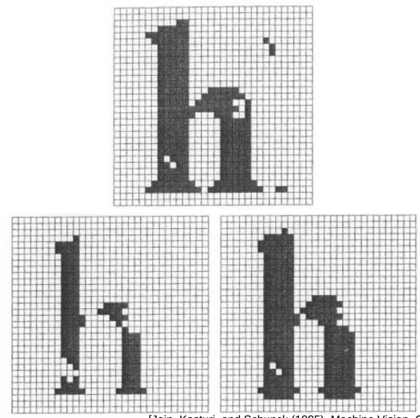
Expanding Followed By Shrinking



[Jain, Kasturi, and Schunck (1995). Machine Vision, Ch. 2]

Original

Shrinking Followed By Expanding



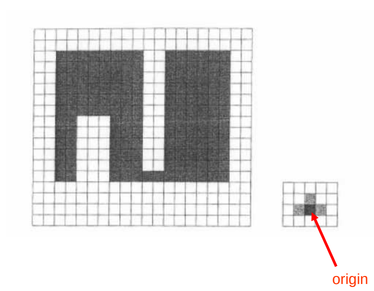
[Jain, Kasturi, and Schunck (1995). Machine Vision, Ch. 2]

Morphological Operators

- Intersection $A \cap B = \{p | p \in A \text{ and } p \in B\}.$
- Union $A \cup B = \{p | p \in A \text{ or } p \in B\}.$
- Complement $\overline{A} = \{p | p \in \Omega \text{ and } p \notin A\}.$

[Jain, Kasturi, and Schunck (1995). Machine Vision, Ch. 2]

Images and Structuring Elements



[Jain, Kasturi, and Schunck (1995). Machine Vision, Ch. 2]

Erosion

- Erosion of an image by a structuring element results in an image that gives all locations where the structuring element is contained in the image.

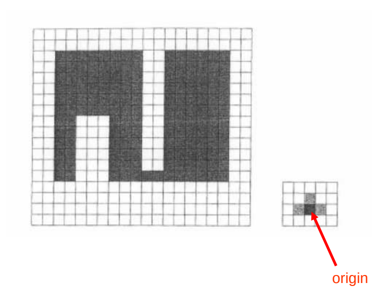
Dilation

- The union of the translations of the image A by the 1 pixels of the image B is called the dilation of A by B.

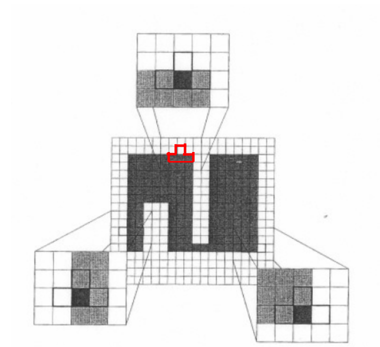
Notation

- Erosion $A \ominus B = \{p | B_p \subseteq A\}.$
- Dilation $A \oplus B = \bigcup_{b_i \in B} A_{b_i}$

Images and Structuring Elements

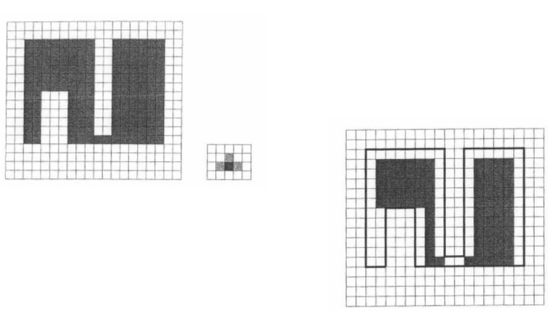


[Jain, Kasturi, and Schunck (1995). Machine Vision, Ch. 2]



[Jain, Kasturi, and Schunck (1995). Machine Vision, Ch. 2]

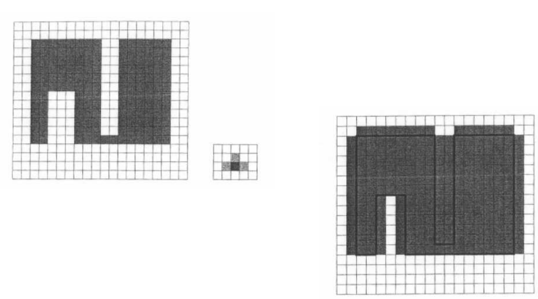
Erosion



[Jain, Kasturi, and Schunck (1995). Machine Vision, Ch. 2]

$A \ominus B = \{p | B_p \subseteq A\}$

Dilation



$A \oplus B = \bigcup_{b_i \in B} A_{b_i}$

[Jain, Kasturi, and Schunck (1995). Machine Vision, Ch. 2]

THE END