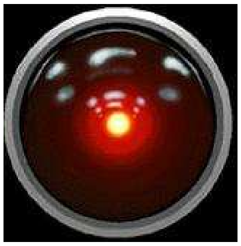


HCI/ComS 575X:
Computational Perception

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

Mathematical Morphology

January 22, 2007

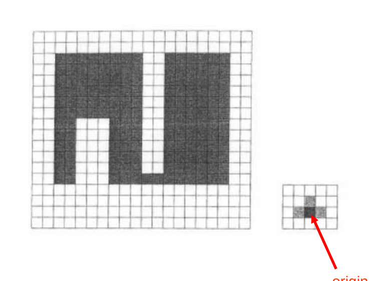
Reading for Today's Lecture

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

Reading for Next Lecture

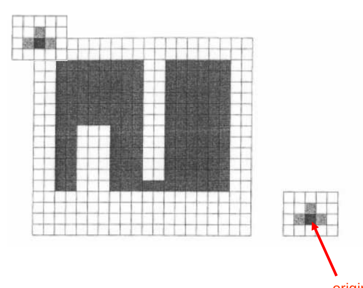
- Jain, Kasturi, and Schunck (1995). Machine Vision, ``Chapter 4: Image Filtering," McGraw-Hill, pp. 112-139.

Images and Structuring Elements



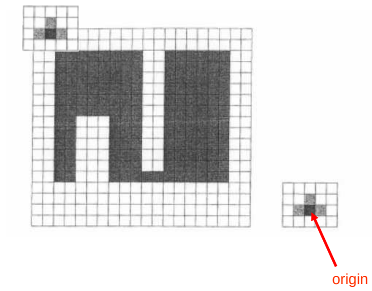
[Jain, Kasturi, and Schunck (1995). Machine Vision, Ch. 2]

Images and Structuring Elements



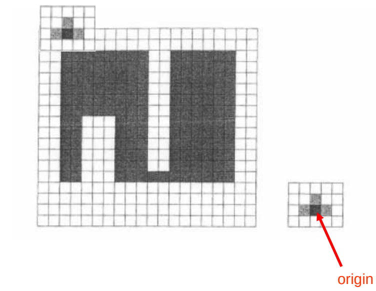
[Jain, Kasturi, and Schunck (1995). Machine Vision, Ch. 2]

Images and Structuring Elements



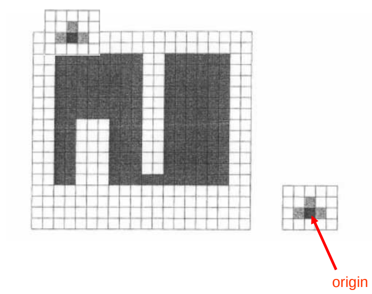
[Jain, Kasturi, and Schunck (1995). Machine Vision, Ch. 2]

Images and Structuring Elements



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

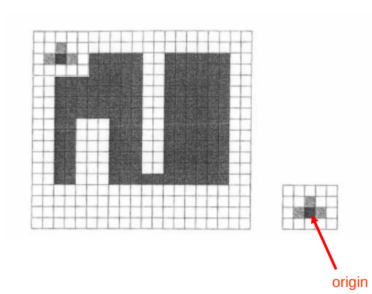
Images and Structuring Elements



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

- And so on …. until we reach the first pixel of the larger image

Images and Structuring Elements

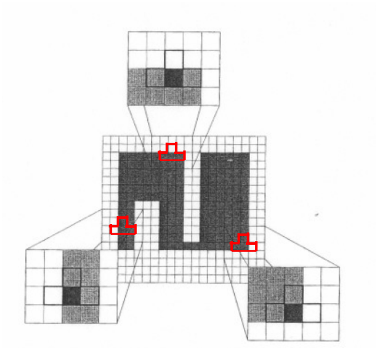


[Jain, Kasturi, and Schunck (1995). Machine Vision, Ch. 2]

Erosion

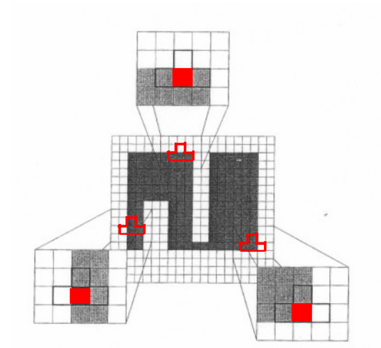
- Erosion of an image by a structuring element results in an image that gives all locations where the structuring element is contained in the image.

Examples of Pixels that will be deleted



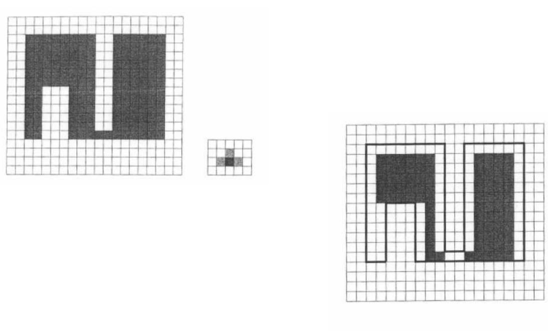
[Jain, Kasturi, and Schunck (1995). Machine Vision, Ch. 2]

Examples of Pixels that will be deleted



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

Erosion Results

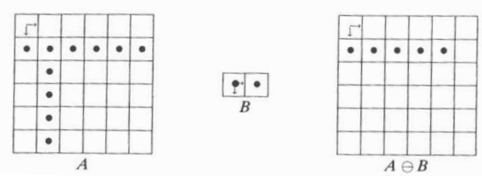


[Jain, Kasturi, and Schunck (1995). Machine Vision, Ch. 2]

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

Another Example

$A = \{(1,0), (1,1), (1,2), (1,3), (1,4), (1,5),$
$(2,1), (3,1), (4,1), (5,1),\}$
$B = \{(0,0), (0,1)\}$
$A \ominus B = \{(1,0), (1,1), (1,2), (1,3), (1,4)\}$



[Haralick and Shapiro (1993). "Computer and Robot Vision," Ch. 5]

Yet Another Example

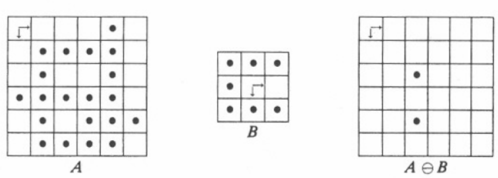


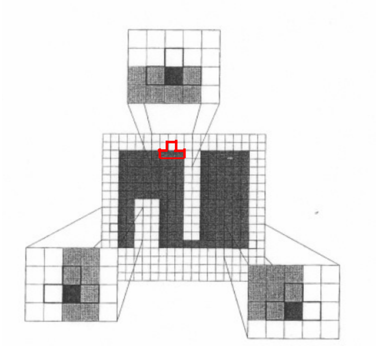
Figure 5.5 Erosion of a set A by a structuring element B that does not contain the origin. As a result, no point of the erosion is guaranteed to be in common with A. However, some translations of $A \ominus B$ are contained in A.

[Haralick and Shapiro (1993). "Computer and Robot Vision," Ch. 5]

Dilation

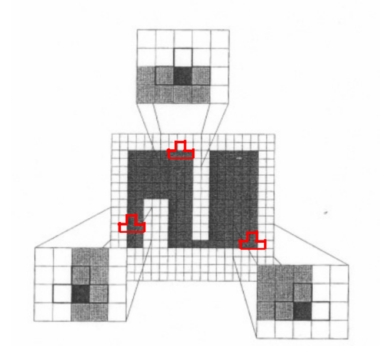
- The union of the translations of the image A by the 1 pixels of the image B is called the dilation of A by B.

Dilation Example



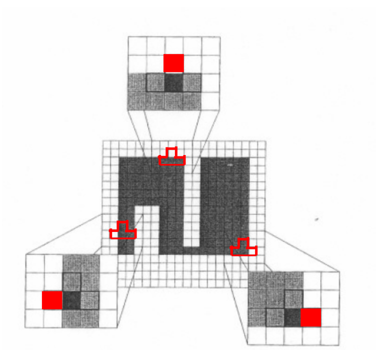
[Jain, Kasturi, and Schunck (1995). Machine Vision, Ch. 2]

Examples of pixels that will be added



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

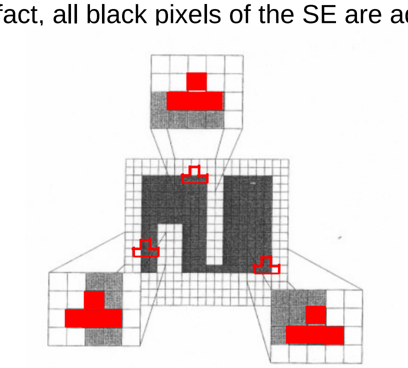
Examples of pixels that will be added



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

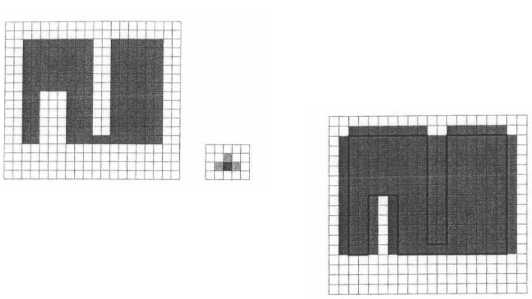
Examples of pixels that will be added

(in fact, all black pixels of the SE are added)



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

Dilation Results

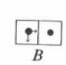


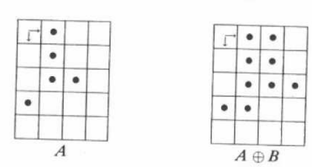
$$A \oplus B = \bigcup_{b_i \in B} A_{b_i}$$

$$A_t = \{c \in E^N \mid c = a + t \text{ for some } a \in A\}$$

$$A = \{(0,1),(1,1),(2,1),(2,2),(3,0)\}$$

$$B = \{(0,0),(0,1)\}$$





$$A \oplus B = \{(0,1),\ (1,1),\ (2,1),\ (3,0)\ (0,2),\ (1,2),\ (2,2),\ (2,3),\ (3,1)\}$$

[Haralick and Shapiro (1993). "Computer and Robot Vision," Ch. 5]

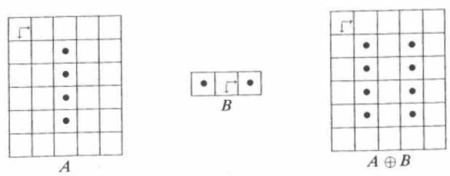


Figure 5.2 Set A dilated by a structuring element B that does not contain the origin. As a result, the dilated set is not even guaranteed to have a single point in common with A. However, there are always translations of $A \oplus B$ that can contain A.

[Haralick and Shapiro (1993). "Computer and Robot Vision," Ch. 5]

Notation

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

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

Real Examples

Segmenting the Body and Handle

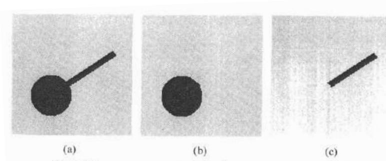


Figure 5.16 Extraction of the body and handle of a shape F by opening with L for the body and taking the residue of the opening for the handle; (a) F, (b) $F \circ L$, and (c) $F - F \circ L$.

[Haralick and Shapiro (1993). "Computer and Robot Vision," Ch. 5]

Horizontal & Vertical Lines

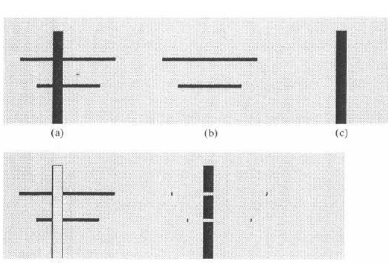
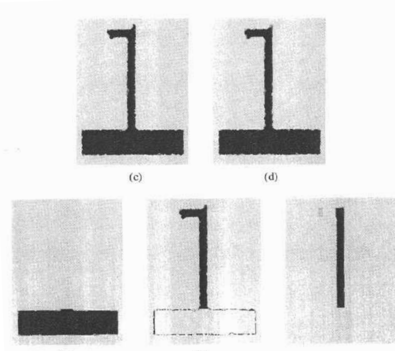


Figure 5.17 Extraction of the trunk and arms of a shape F by opening with vertically and horizontally oriented structuring elements.

[Haralick and Shapiro (1993). "Computer and Robot Vision," Ch. 5]



[Haralick and Shapiro (1993). "Computer and Robot Vision," Ch. 5]

Circles & Lines

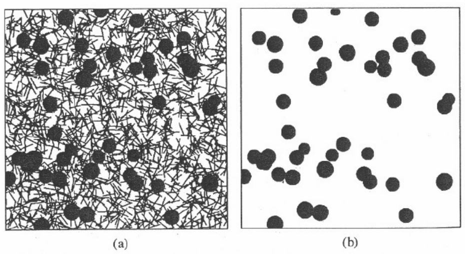


Figure 5.19 (a) A binary image. (b) Opening of (a) with a disk structuring element.

[Haralick and Shapiro (1993). "Computer and Robot Vision," Ch. 5]

Clustering

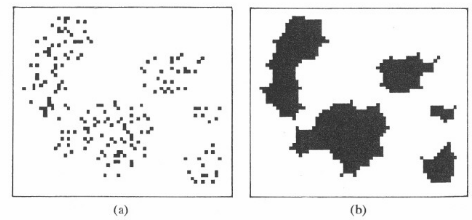
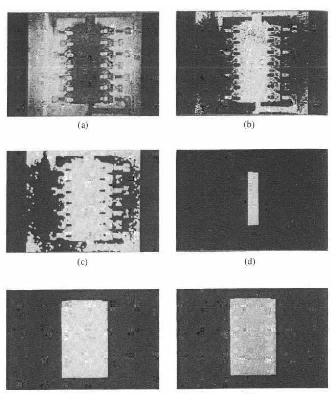


Figure 5.23 (a) A binary image with five clusters of points. Points within each cluster satisfy the partition property with distance ρ_0, and the clusters are farther from each other than $2\rho_0$ pixels. (b) The image of (a) closed by a disk with a radius just greater than $2\rho_0$.

[Haralick and Shapiro (1993). "Computer and Robot Vision," Ch. 5]

Chip



[Haralick and Shapiro (1993). "Computer and Robot Vision," Ch. 5]

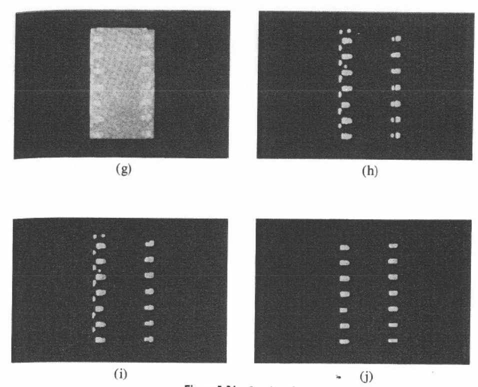


Figure 5.24 *Continued.*

[Haralick and Shapiro (1993). "Computer and Robot Vision," Ch. 5]

Algebraic Relations

$$(A \oplus B) \oplus C = A \oplus (B \oplus C)$$
$$(A \cup B) \oplus C = (A \oplus C) \cup (B \oplus C)$$
$$A \oplus B = \bigcup_{b \in B} A_b$$
$$A \subseteq B \Rightarrow A \oplus C \subseteq B \oplus C$$
$$(A \cap B) \oplus C \subseteq (A \oplus C) \cap (B \oplus C)$$
$$A \oplus (B \cup C) = (A \oplus B) \cup (A \oplus C)$$
$$(A \oplus B)^c = A^c \ominus \check{B}$$
$$A \oplus B_t = (A \oplus B)_t$$
$$A \oplus B = B \oplus A$$
$$(A \ominus B) \ominus C = A \ominus (B \oplus C)$$
$$(A \cap B) \ominus C = (A \ominus C) \cap (B \ominus C)$$
$$A \ominus B = \bigcap_{b \in B} A_{-b}$$
$$A \subseteq B \Rightarrow A \ominus C \subseteq B \ominus C$$
$$(A \cup B) \ominus C \supseteq (A \ominus C) \cup (B \ominus C)$$
$$A \ominus (B \cap C) \supseteq (A \ominus B) \cup (A \ominus C)$$
$$A \ominus (B \cup C) = (A \ominus B) \cap (A \ominus C)$$
$$A \ominus B_t = (A \ominus B)_{-t}$$

[Haralick and Shapiro (1993). "Computer and Robot Vision," Ch. 5]

Matlab Demos
(The code is posted on
the Class Web)

THE END