Computer Vision ECE5470

Anthony P. Reeves
School of Electrical and Computer Engineering
Cornell University

Lecture 5: Binary Image Representation and Processing

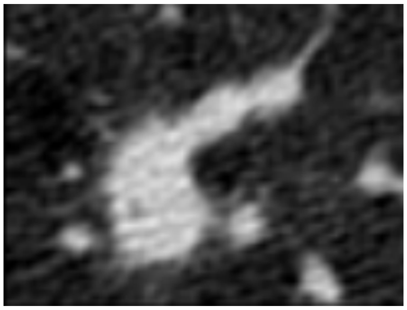
Image Analysis

- Image processing (filtering)
- Segmentation (thresholding etc.)
  - binary image filtering
  - binary shape representation
- Feature Description
- Classification

Topics

- Binary Image Algorithms
  - Region description using moments
  - The Medial Axis Transform
  - The Distance Transform
  - Thinning
- Morphological Filtering
  - Convolution review
  - Erosion and Dilation
  - Opening and Closing

2D Segmentation



5 mm

2D Image

2D Thresholding

Region Description

- How big is it ?
- Where is it?
- How is it oriented?
- What is its shape?
- Other characteristics?

Moments Summary

The second order moments of a binary image region provide:

1. The location of the COM of the region
2. The size of the region
3. The orientation of the region

(4. The width and height of the region)

How to represent shape?

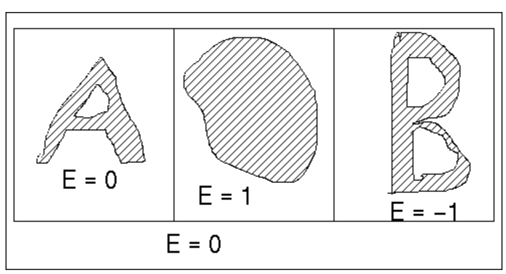
- High order moments of a region
- Trace the boundary of a region
  what about holes?

- Ad hoc measures
  - Area A
  - Perimeter P
  - Compactness C

$$C = \frac{P^2}{A}$$

Topological Invariance

- Euler Number E = C - H
  where C is the number of components and H is the number of holes

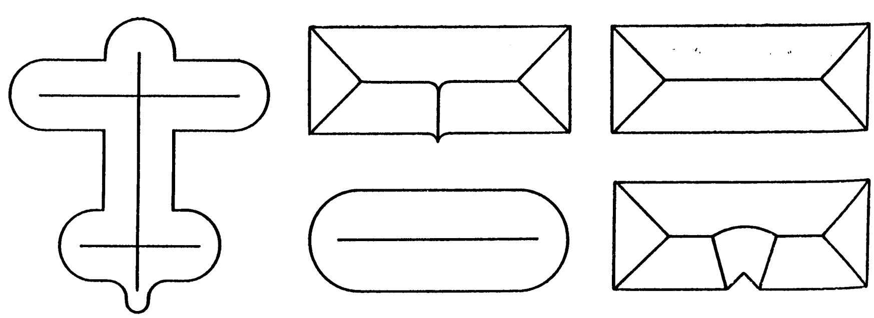


- Invariant to size, locally computable
- extends to higher dimensions

Medial Axis Transform (MAT)

- Transform a segment to a 1-pixel wide “skeleton” representation
- Definition: The MAT is the locus of the points that are a maximum distance from a regions boundary.
- Concept: compact representation of a regions shape
- Problem: small boundary changes can cause large changes to the MAT

MAT Examples



The Distance Transform

- Consider a set of pixels in a region S
  Define: d([i,j],S) is the distance from pixel i,j to region S.
- Problem: which distance metric? [Euclidean, chess board, city block]
- Define: The **distance transform** is the distance from each pixel to the background

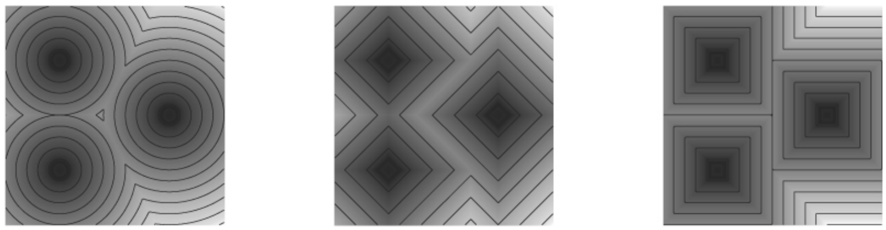
Example Distance Transform

	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				

0	0	0	0	0	0	0	0	0	0	0	0
0	1	1	1	0	0	0	0	0	0	0	0
0	1	2	1	0	0	0	0	0	1	0	0
0	1	1	1	0	0	0	0	1	2	1	0
0	0	0	1	1	1	0	0	1	1	1	0
0	0	0	1	2	2	1	0	0	0	0	0
0	0	0	1	2	3	2	1	0	0	0	0
0	1	1	2	3	4	3	2	1	0	0	0
0	1	2	3	4	4	4	3	2	1	0	0
0	1	2	2	3	3	3	3	2	1	0	0
0	0	1	1	2	2	2	2	1	0	0	0
0	0	0	0	1	1	1	1	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0

- Distance transform (4-connected)

Example Distance Transform



(a) (b) (c)

Figure 2.11: Three distances used often in distance transform calculations—the input consists of three isolated 'ones'. Output distance is visualized as intensity; lighter values denote higher distances. Contour plots are superimposed for better visualization. (a) Euclidean distance D_E. (b) City block distance D_4. (c) Chessboard distance D_8. © *Cengage Learning 2015.*

MAT

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

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

- Select maxima for medial axis
- This MAT is not connected

Reconstruction from MAT

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

	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				

- Must use the same distance metric

Medial Axis Transform

- A skeleton is a set of pairs $\{\underline{x}, d_s(\underline{x}, B)\}$ where $\{\underline{x}, B\}$ is the distance from $\underline{x}$ to the boundary

- Reconstruction: The region may be reconstructed by computing the union of “disks” of radius $d_s(\underline{x}, B)$ centered on $\underline{x}$

Thinning

Thinning is the process of peeling off boundary layers of pixels (maintaining connectivity) until a 1-pixel wide representation is achieved. Thinning approximates the MAT skeleton

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			

More Thinning

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		

- Result is only an approximation of the MAT
- Original image cannot be reconstructed

Medial Axis Transform: Summary

- Represents a 2D shape by a “center line” skeleton.
  - Skeleton may not be connected
  - Original binary shape can be reconstructed
  - Small surface “nicks” can cause problems
  - Useful for long thin regions (such as vessels, not very useful for “blob” objects
- Thinning results in a binary approximation to the MAT skeleton

Morphological (geometric) Binary Image Filtering

Binary Image Filtering: Morphological (geometric) Processing

Local filtering on binary images is performed with "opening" and "closing" operations that are related to the convolution operation for gray-level images

Convolution (Review)

$$h(x,y) = f(x,y) \otimes g(x,y)$$

is defined by:

$$h(x,y) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(x',y')g(x-x',y-y')dx'dy'$$

For a discrete function

$$h[i,j] = f[i,j] \otimes g[i,j]$$

$$h[i,j] = \sum_{k=1}^{n}\sum_{l=1}^{m} f[k,l]g[i-k,j-l]$$

No need for causality (kernel is usually centered on pixel)

Image Convolution

- Usually $h[i,j]$ and $f[i,j]$ are images - same size
- Border elements of $h[i,j]$ may not be valid.
- $g[i,j]$ - the *convolution mask* is a small $n \times m$ matrix.

convolution, correlation:

$$h[i,j] = \sum_{k=-\frac{n}{2}}^{\frac{n}{2}} \sum_{l=-\frac{m}{2}}^{\frac{m}{2}} f[i-k, j-l]g[k,l]$$

Morphological Filtering (Dilation)

$$h[i,j] = f[i,j] \oplus g[i,j]$$

is defined by:

$$h[i,j] = \bigvee_{k=-\frac{n}{2}}^{\frac{n}{2}} \bigvee_{l=-\frac{m}{2}}^{\frac{m}{2}} f[i-k, j-l]g[k,l]$$

where $g[i,j]$ - the *structuring element* is a small $n \times m$ binary valued matrix.
Dilation may also be expressed by:

$$h[i,j] = \bigvee_{[l,k]\in\{g[l,k]\}} f[i-k, j-l]g[k,l]$$

$$h[i,j] = \bigvee_{[l,k]\in\{g[l,k]\}} f[i-k, j-l]$$

Morphological Filtering (Erosion)

Erosion:

$$h[i,j] = f[i,j] \ominus g[i,j]$$

is defined by:

$$h[i,j] = \bigwedge_{[l,k]\in\{g[l,k]\}} f[i-k, j-l]$$

Erosion gives all locations where a structuring element fits within a region.

Expanding and Shrinking

- Expanding: change a pixel from 0 to 1 if any of its (8) neighbors are 1

  Dilate with g =
  $$\begin{array}{|ccc|}\hline 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \\ \hline\end{array}$$

- Shrinking: Change a pixel from 1 to 0 if any if its (8) neighbors are 0

  Erode with g =
  $$\begin{array}{|ccc|}\hline 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \\ \hline\end{array}$$

Structuring Elements

- The location of the coordinate origin of the structuring element is important

$$\boxed{\begin{matrix} 0 & 0 & 1 \\ 0 & 0 & 0 \end{matrix}}$$

What is the effect of dilating with the above element?

- For odd dimensioned elements assume the origin is in the center

$$\boxed{\begin{matrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{matrix}}$$

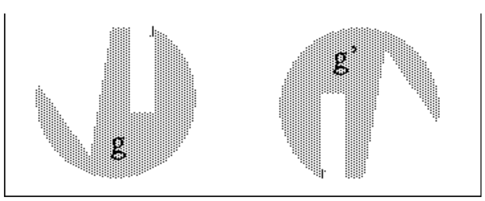
Erosion

Erosion is the dilation of the background with the reflected structuring element.

$$f \Theta g = \overline{\bar{f} \oplus g'}$$

g' is the geometric complement (reflection)

$$g' = \{-p \mid p \in g\}$$

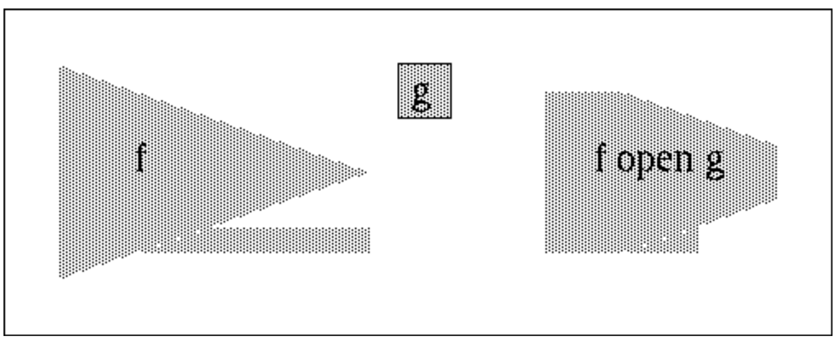


Morphological Operations

- Erosion $f \ominus g$
- Dilation $f \oplus g$
- Opening $f \ominus g \oplus g$
- Closing $f \oplus g \ominus g$
- Reflection g'

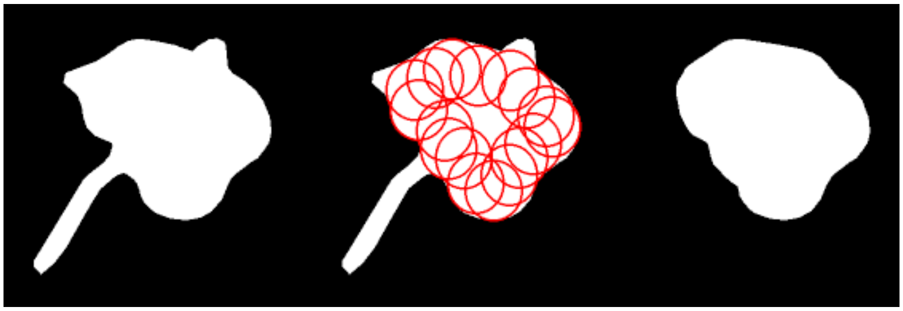
Opening $f \ominus g \oplus g$

- Remove any parts of f that g will not fit inside f.

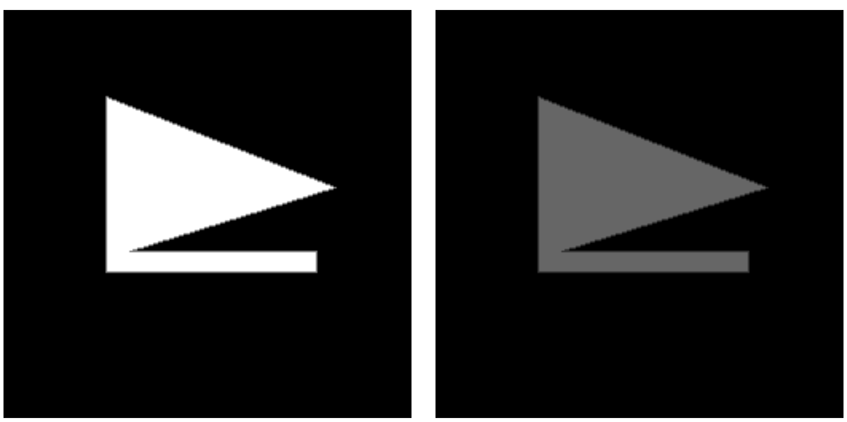


Basic Morphological (geometric) Filtering

2D morphological opening example

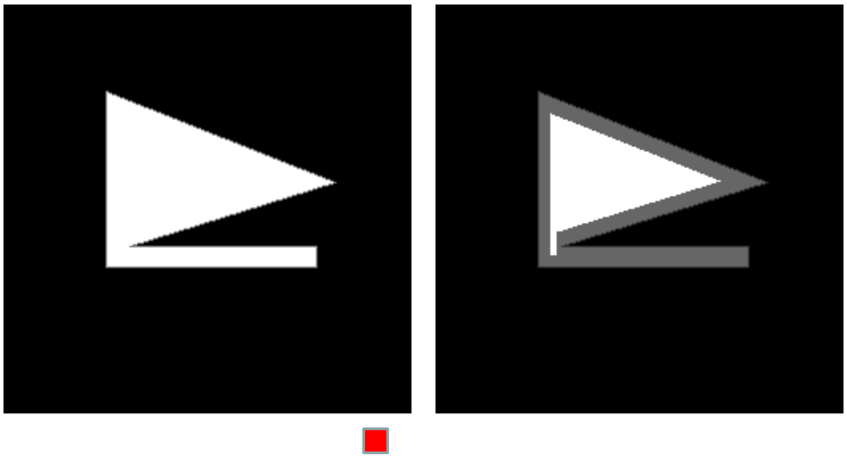


Opening $f \ominus g \oplus g$

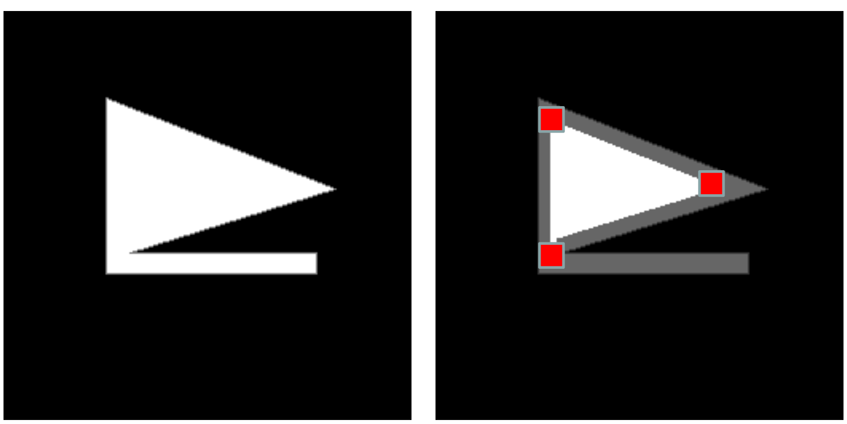


Structuring element (g)

Opening $f \ominus g \oplus g$

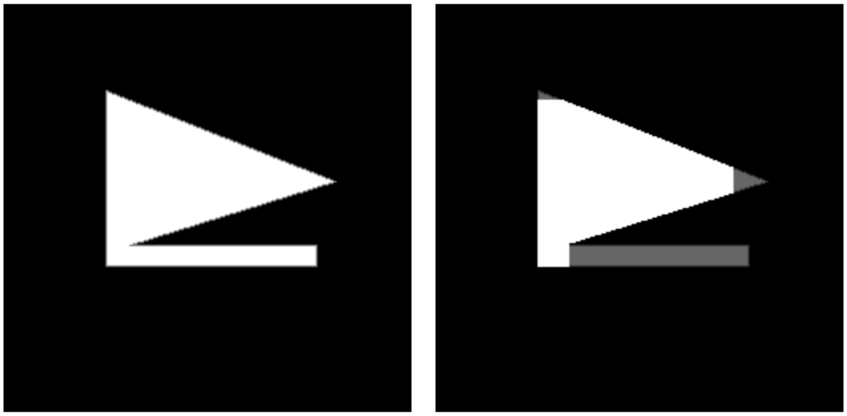


Opening $f \ominus g \oplus g$

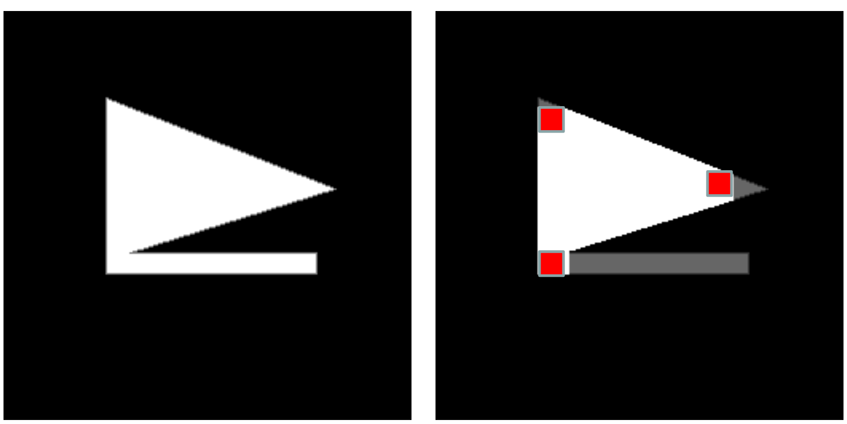


Cornell University
Vision and Image Analysis Group
ECE 5470
5

Opening $f \ominus g \oplus g$



Opening $f \ominus g \oplus g$



Opening: wherever the structuring element will fit inside the object

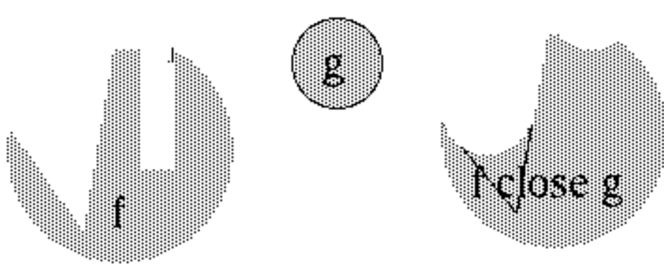
Opening $f \ominus g \oplus g$

Commands used to create the example

```
vgenim s=256 of=im                          ; #create a blank image
##   draw shape using vdview from vview     ; #draw a shape on the image
varend -f im.1.vxa of=im2 fc=255 bc=255     ; #convert annotation to binary image
vmorph -e im2 s=17,17 t=c of=im3            ; #erode with 17 x 17 square
vmorph -d im3 s=17,17 t=c of=im4            ; #dilate result with 17. x 17 square
vpix tf=0.4 im2 of=im5                      ; #create dim input image for visualization
vxport im2 of=im2.png                       ; #create png version for powerpoint
vop -or im5 im3 | vxport of=im6.png         ; #create erosion visualization
vop -or im5 im4 | vxport of=im7.png         ; #create dilation visualization
```


Closing $f \oplus g \ominus g$

- Fill any parts of a region that g' will not fit into.

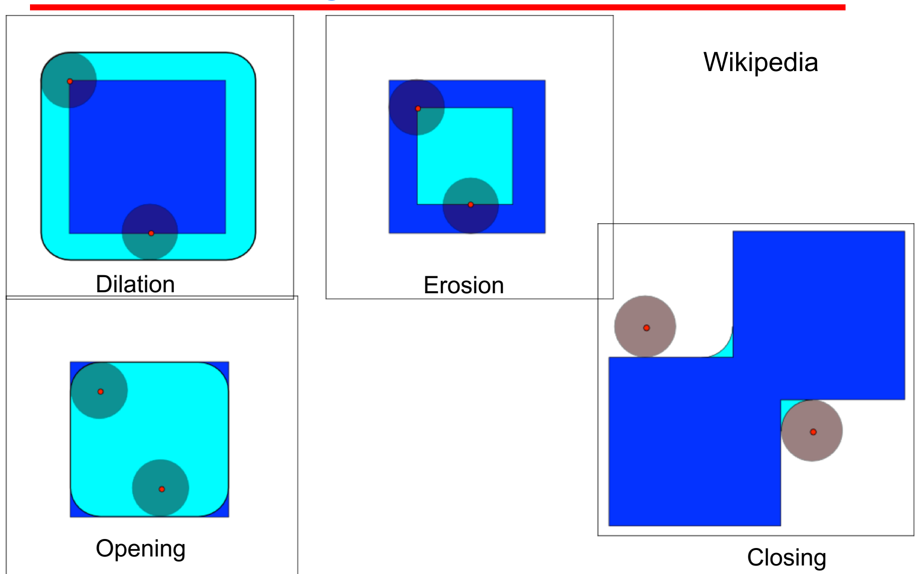


Morphological Filtering (Summary)

Fundamental “shape” image filtering

- Erosion and dilation are the basic primitives but opening and closing are frequently used for filtering.
- Complement is the Geometric inverse
- For general (isotropic) filtering the usual filter kernel is a circle; in practice a square may be a reasonable approximation.
- Other shaped kernels are appropriate for specific tasks
- [Also may be defined for grey level images: in most cases a box filter is used which implies max or min filtering]

Morphological Filtering Examples



Morphological Filtering Notation

Example: Dilation   See “notation” in *Lecture Notes* on course website

- pixel transformation approach (Sonka V3 pg 661, eq 13.9):

$$A \oplus B = \{p \in \mathcal{E}^2 : p = a + b, a \in A, b \in B\}$$

- whole image approach (Sonka V3 pg 662, eq 13.12):

$$\mathrm{A} \oplus B = \bigcup_{b \in B} A_b$$

- *convolution* approach (lecture notes):

$$a[i,j] = \bigcup_{[l,k] \in B} a[i-l, j-k]$$

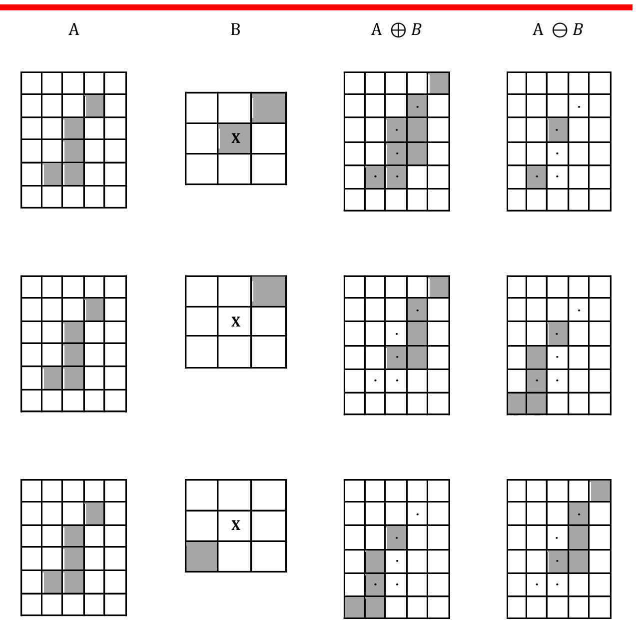
The convolution approach **always** uses a mirrored kernel and gives **different** results for asymmetric kernels; to get the same results one would use the following equation:

$$a[i,j] = \bigcup_{[l,k] \in B} a[i+l, j+k]$$

Morphological Filtering Notation

Examples:
For Prelims

‘**x**’ marks the origin (reference point) of the kernel.



Binary Image Filtering (Summary)

Task specific algorithms for binary image processing may be constructed with a combination of the Morphological binary operations Dilation, Erosion, and Reflection combined with the regular binary operations AND, OR, and NOT.

There are often many opportunities for improving the efficiency of implementations (similar to the methods used for optimizing conventional linear filters).

Binary Image Analysis (Summary)

- Connected Component Analysis
- Moments (basic size and location)
- Medial Axis Transform or Boundary Representation
- Binary Image Filtering (Morphological Filtering)