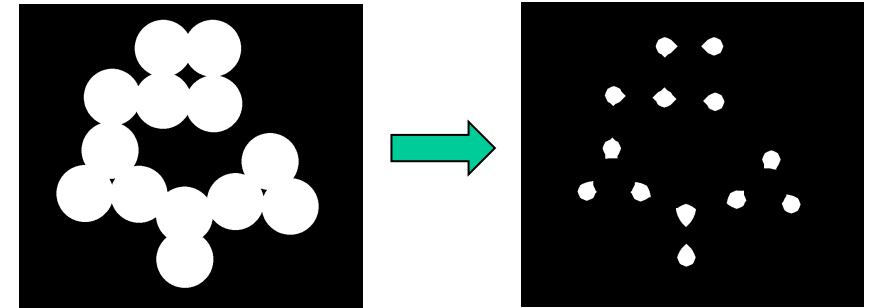


# Morphological Image Processing

- Binary dilation and erosion
- Set interpretation
- Opening, closing, morphological edge detectors
- Hit-miss filter
- Morphological filters for gray-level images
- Cascading dilations and erosions
- Rank filters, median filters, majority filters



## INTEREST-POINT DETECTION

Feature extraction typically starts by finding the salient interest points in the image. For robust image matching, we desire interest points to be repeatable under perspective transformations (or, at least, scale changes, rotation, and translation) and real-world lighting variations. An example of feature extraction is illustrated in Figure 3. To achieve scale invariance, interest points are typically computed at multiple scales using an image pyramid [15]. To achieve rotation invariance, the patch around each interest point is canonically oriented in the direction of the dominant gradient. Illumination changes are compensated by normalizing the mean and standard deviation of the pixels of the gray values within each patch [16].

# Binary image processing

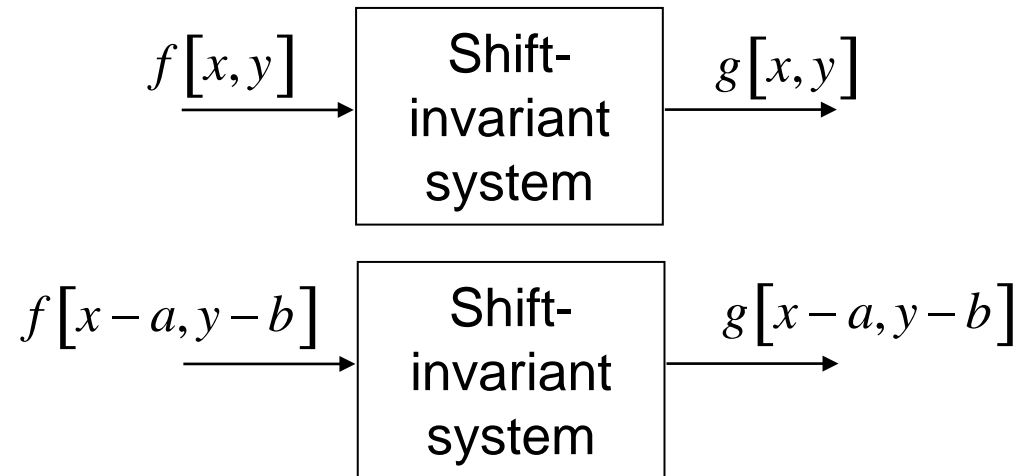
- Binary images are common
  - Intermediate abstraction in a gray-scale/color image analysis system
    - Thresholding/segmentation
    - Presence/absence of some image property
  - Text and line graphics, document image processing
- Representation of individual pixels as 0 or 1, convention:
  - foreground, object = 1 (white)
  - background = 0 (black)
- Processing by logical functions is fast and simple
- Shift-invariant logical operations on binary images:  
*“morphological” image processing*
- Morphological image processing has been generalized to gray-level images via level sets

# Shift-invariance

- Assume that digital images  $f[x,y]$  and  $g[x,y]$  have infinite support

$$[x,y] \in \{\dots, -2, -1, 0, 1, 2, \dots\} \times \{\dots, -2, -1, 0, 1, 2, \dots\}$$

... then, for all integers  $a$  and  $b$



- Shift-invariance does **not** imply linearity (or vice versa).

# Structuring element

- Neighborhood “window” operator

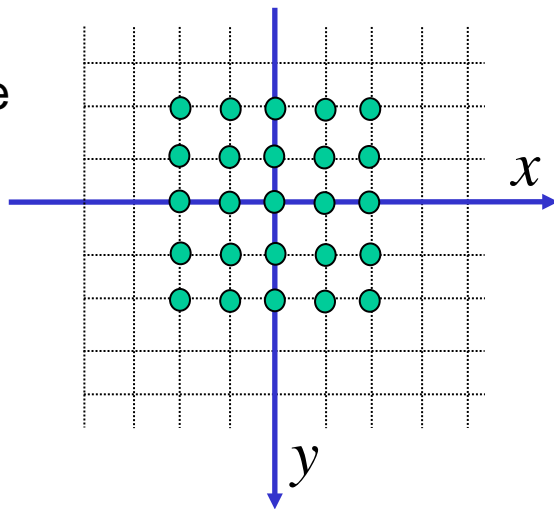
$$W \{ f[x, y] \} = \left\{ f[x - x', y - y'] : [x', y'] \in \Pi_{xy} \right\}$$

“structuring element”

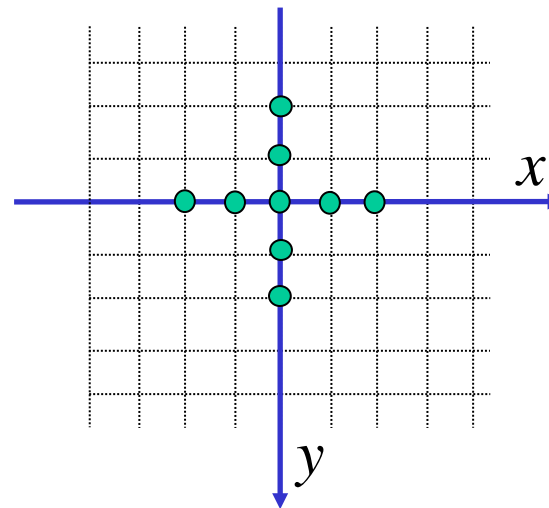


- Example structuring elements  $\Pi_{xy}$ :

5x5 square



cross



# Structuring element

- Neighborhood “window” operator

$$W^{(-)} \{ f[x, y] \} = \{ f[x - x', y - y'] : [x', y'] \in \Pi_{xy} \}$$

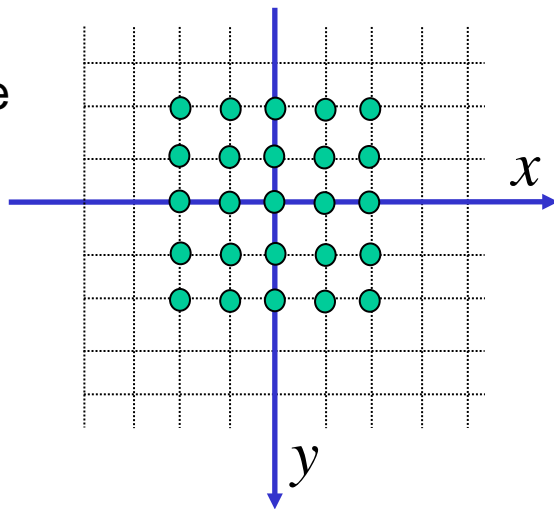
$$W^{(+)} \{ f[x, y] \} = \{ f[x + x', y + y'] : [x', y'] \in \Pi_{xy} \}$$

“structuring element”

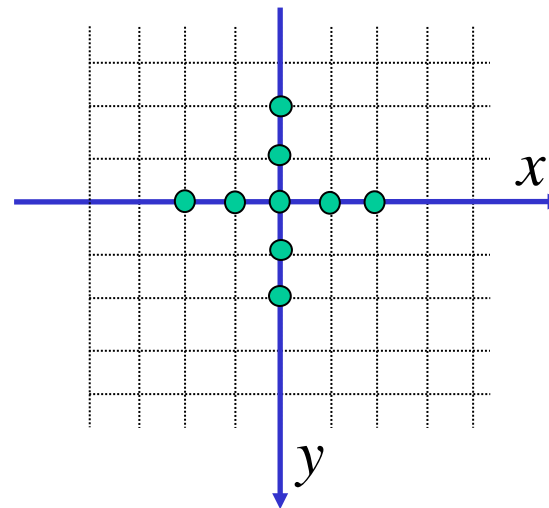


- Example structuring elements  $\Pi_{xy}$ :

5x5 square

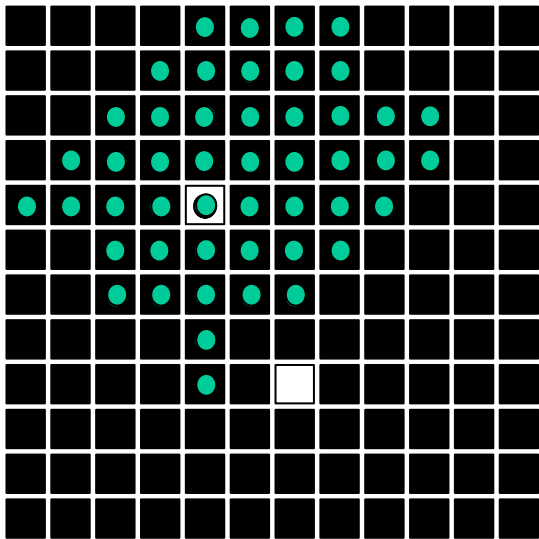


cross

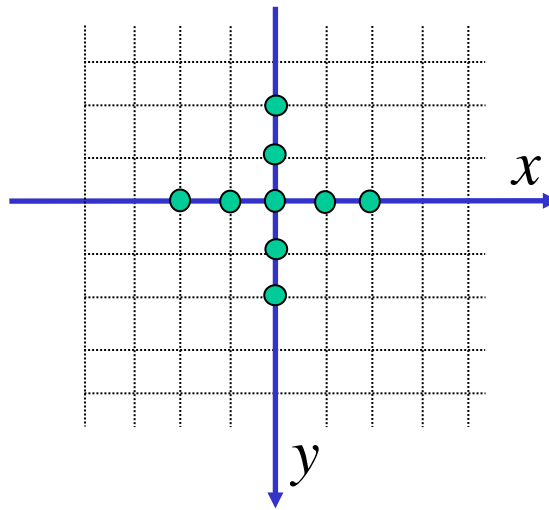


# Binary dilation (expanding foreground)

$$g[x, y] = OR \left[ W^{(-)} \left\{ f[x, y] \right\} \right] := \text{dilate}(f, W)$$



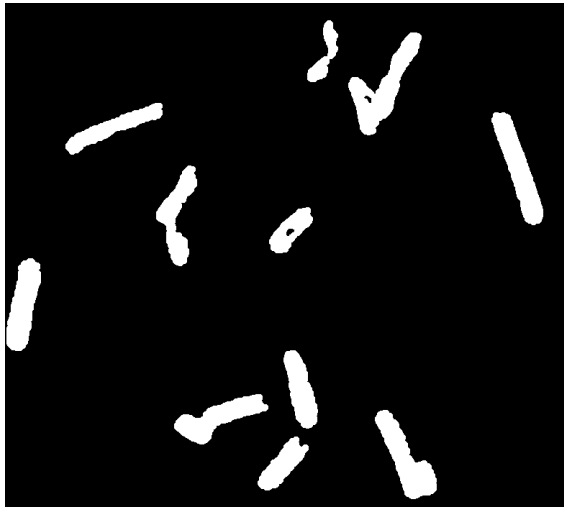
$f[x, y]$



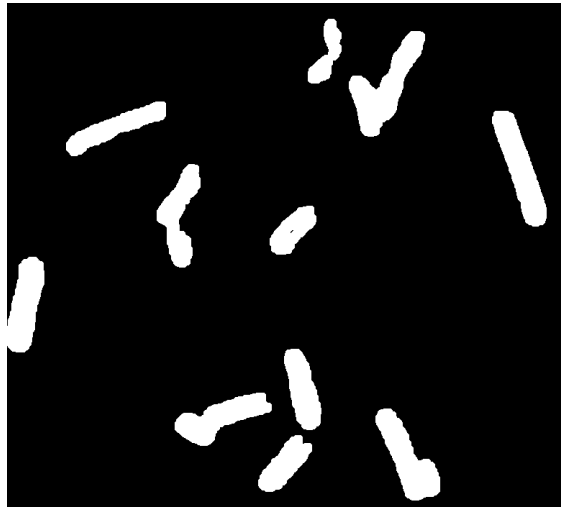
$\Pi_{xy}$

# Binary dilation with square structuring element

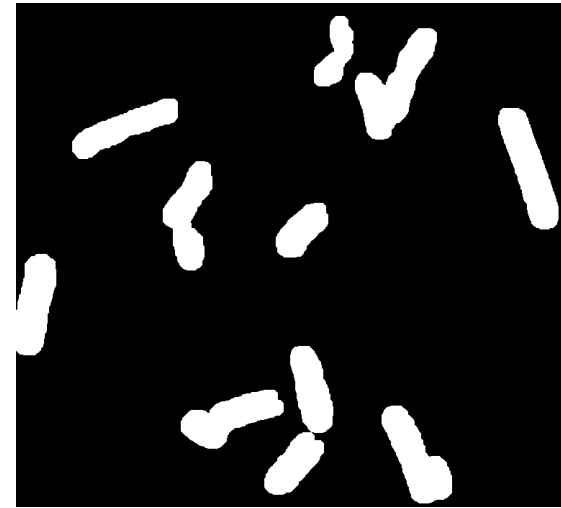
$$g[x, y] = OR \left[ W^{(-)} \left\{ f[x, y] \right\} \right] := dilate(f, W)$$



Original (701x781)



dilation with  
3x3 structuring element



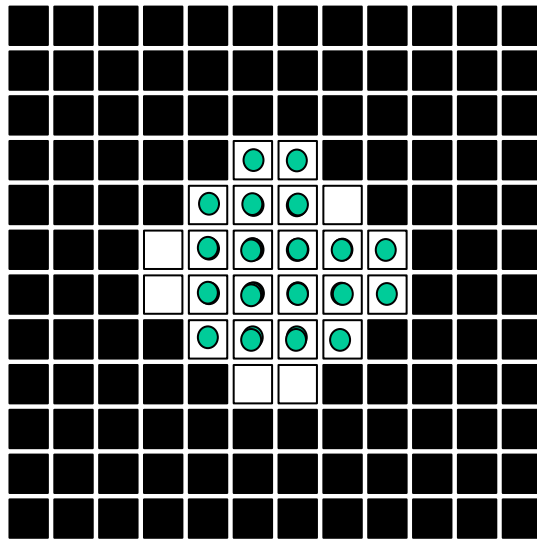
dilation with  
7x7 structuring element

- Expands the size of 1-valued objects
- Smooths object boundaries
- Closes holes and gaps

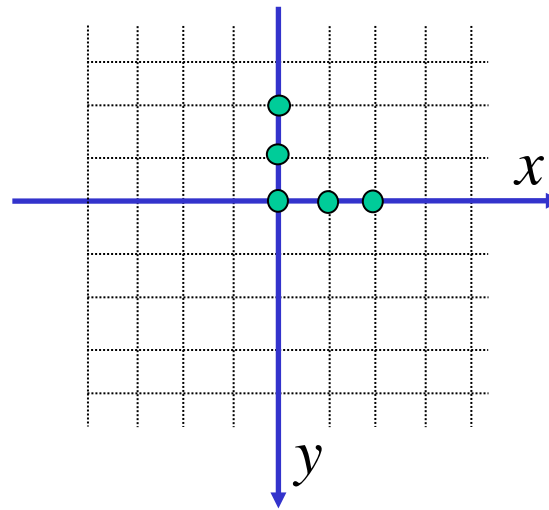


# Binary erosion (shrinking foreground)

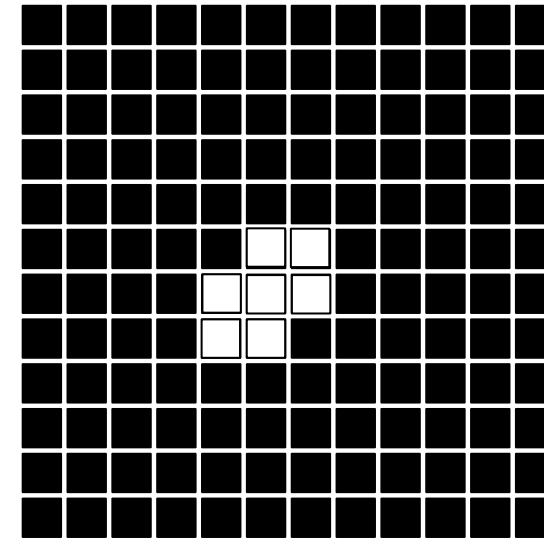
$$g[x, y] = \text{AND} \left[ W^{(+)} \left\{ f[x, y] \right\} \right] := \text{erode}(f, W)$$



$f[x, y]$



$\Pi_{xy}$

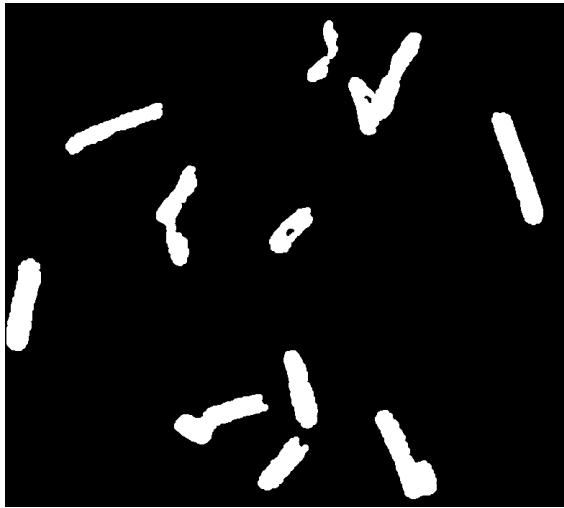


$g[x, y]$

- attention: Matlab function *imerode()* does not flip the window (*imdilate()* does)!

# Binary erosion with square structuring element

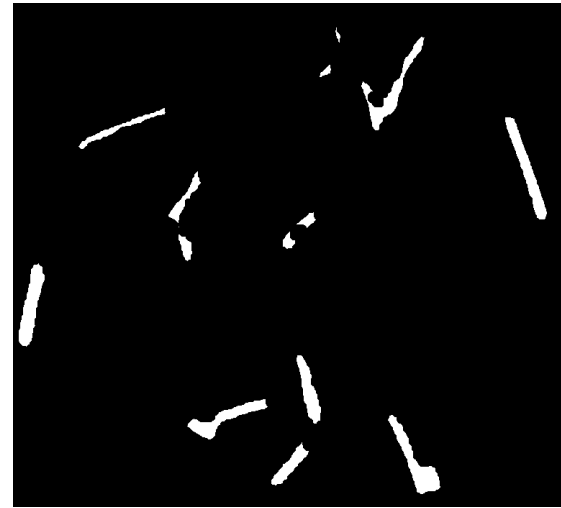
$$g[x, y] = \text{AND} \left[ W^{(+)} \left\{ f[x, y] \right\} \right] := \text{erode}(f, W)$$



Original (701x781)



erosion with  
3x3 structuring element



erosion with  
7x7 structuring element

- Shrinks the size of 1-valued objects
- Smooths object boundaries
- Removes peninsulas, fingers, and small objects



# Relationship between dilation and erosion

- Duality: erosion is dilation of the background

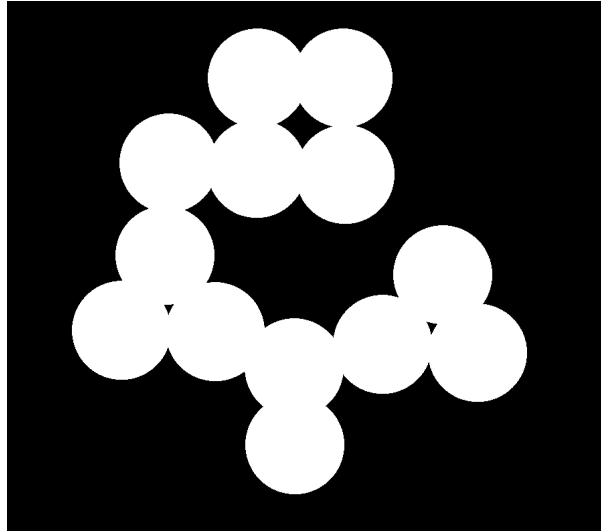
$$\textit{dilate}(f, W) = \textit{NOT}[\textit{erode}(\textit{NOT}[f], W)]$$

$$\textit{erode}(f, W) = \textit{NOT}[\textit{dilate}(\textit{NOT}[f], W)]$$

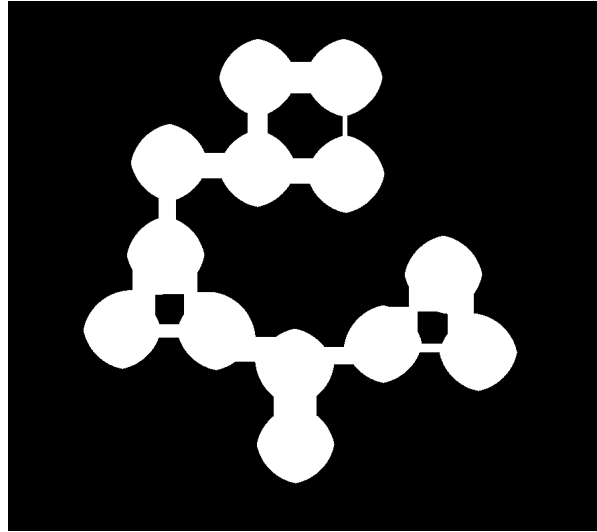
- But: erosion is not the inverse of dilation

$$\begin{aligned} f[x, y] &\neq \textit{erode}(\textit{dilate}(f, W), W) \\ &\neq \textit{dilate}(\textit{erode}(f, W), W) \end{aligned}$$

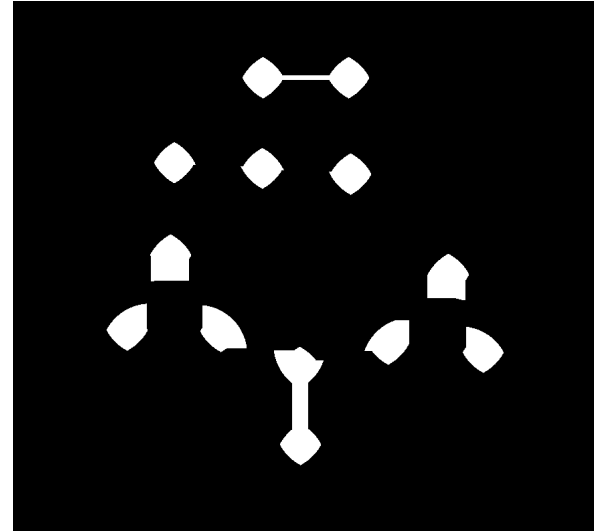
# Example: blob separation/detection by erosion



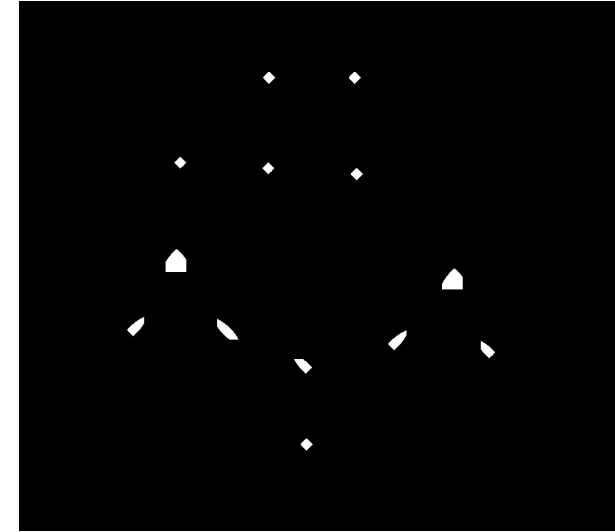
Original binary image  
*Circles* (792x892)



Erosion by 30x30  
structuring element



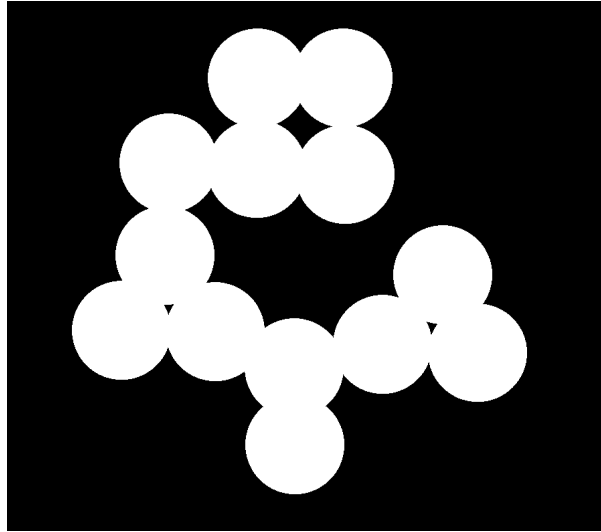
Erosion by 70x70  
structuring element



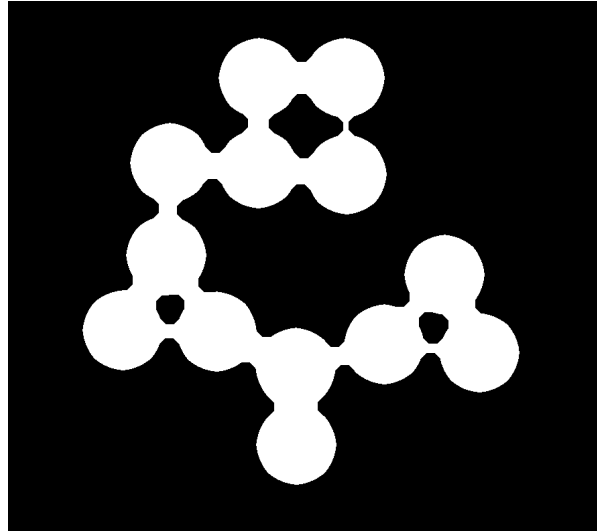
Erosion by 96x96  
structuring element



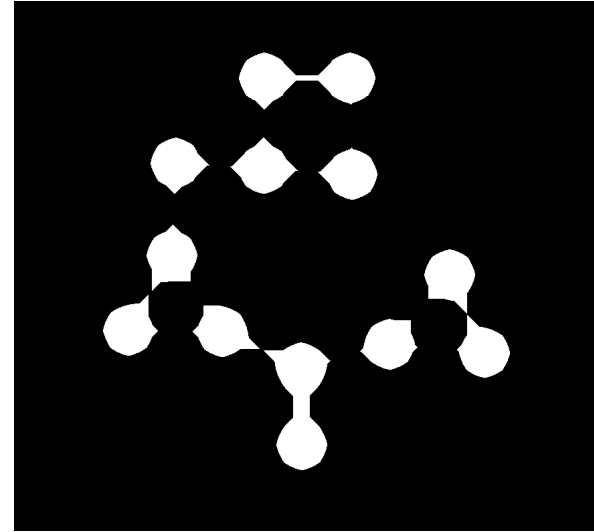
# Example: blob separation/detection by erosion



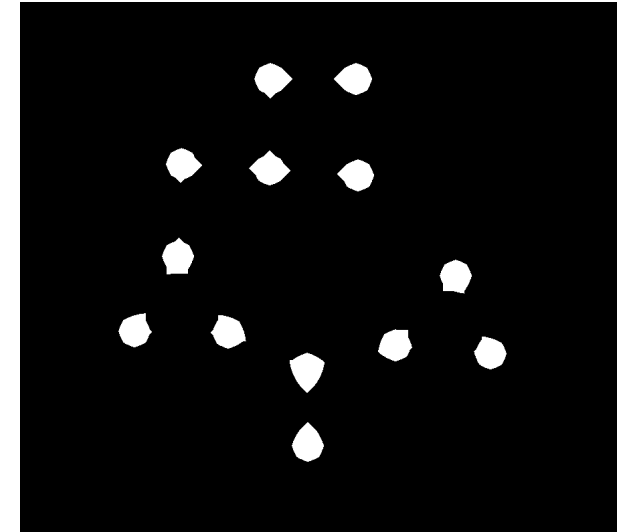
Original binary image  
*Circles* (792x892)



Erosion by disk-shaped  
structuring element  
Diameter=15



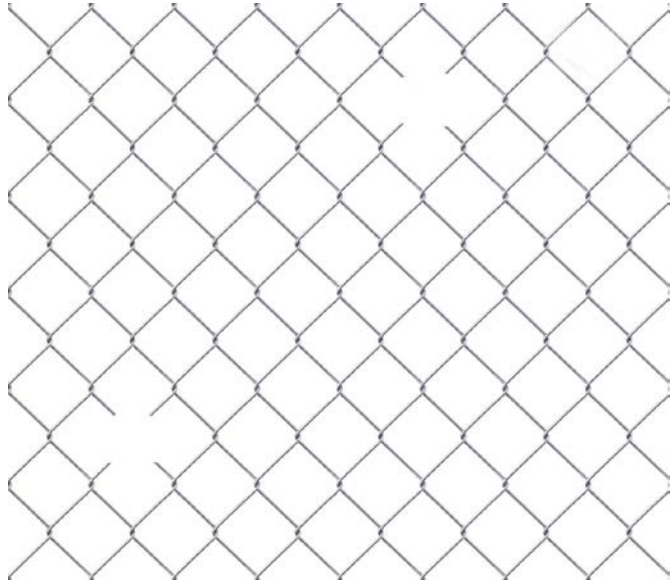
Erosion by disk-shaped  
structuring element  
Diameter=35



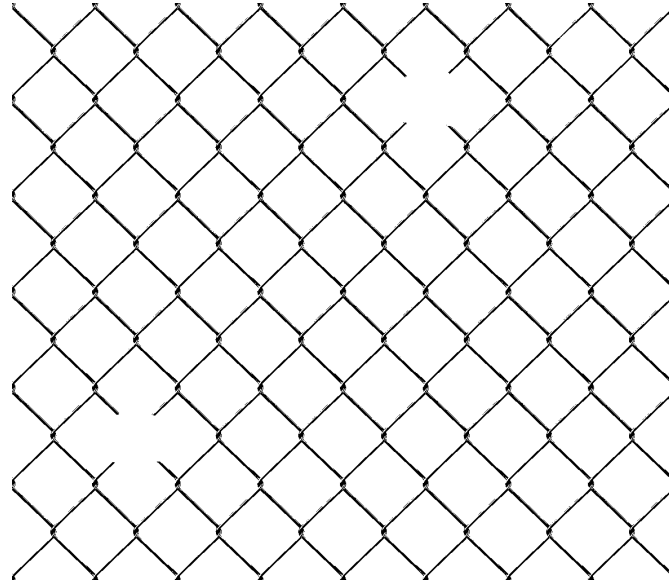
Erosion by disk-shaped  
structuring element  
Diameter=48



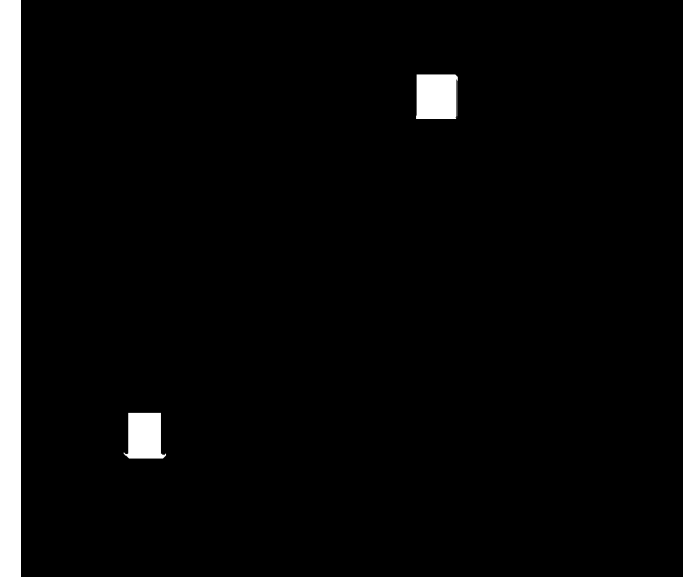
# Example: chain link fence hole detection



Original grayscale image  
*Fence* (1023 x 1173)



*Fence* thresholded  
using Otsu's method



Erosion with 151x151  
"cross" structuring element



# Set-theoretic interpretation

- Set of object pixels

$$F \equiv \left\{ (x, y) : f(x, y) = 1 \right\}$$

*Continuous (x,y).  
Works for discrete [x,y]  
in the same way.*

- Background: complement of foreground set

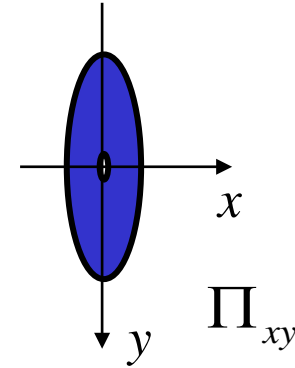
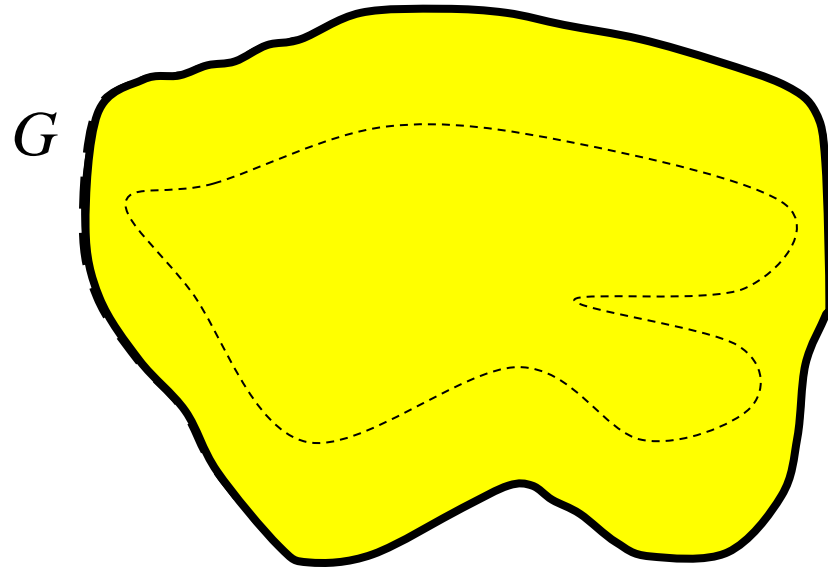
$$F^c \equiv \left\{ (x, y) : f(x, y) = 0 \right\}$$

- Dilation is union of shifted sets (Minkowski set addition)

*Commutative and associative!*

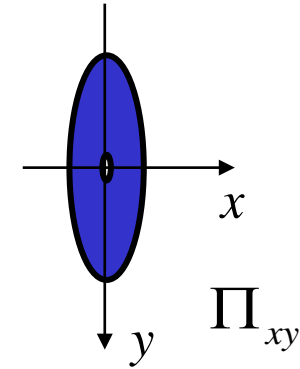
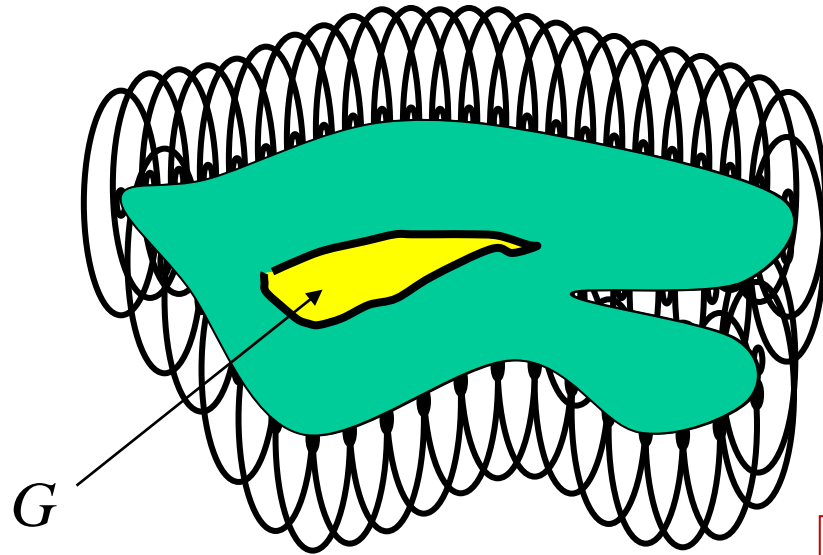
$$G = \bigsqcup_{(x,y) \in F} \mathit{shift}_{xy} \left( \Pi_{rotated} \right)$$

# Set-theoretic interpretation: dilation



$$G = \bigsqcup_{(x,y) \in F} \mathit{shift}_{xy}(\Pi_{rotated})$$

# Set-theoretic interpretation: erosion



*Not commutative!*  
*Not associative!*

intersection of shifted sets (Minkowski set subtraction)

$$G = \bigcap_{(x,y) \in F^c} \text{shift}_{xy} \left( \text{NOT}(\Pi_{\text{rotated}}) \right)$$

↑  
infinite support!

# Opening and closing

- Goal: smoothing without size change

- Open filter

$$\mathit{open}(f, W) = \mathit{dilate}(\mathit{erode}(f, W), W)$$

- Close filter

$$\mathit{close}(f, W) = \mathit{erode}(\mathit{dilate}(f, W), W)$$

- Open filter and close filter are biased

- Open filter removes small 1-regions
- Close filter removes small 0-regions
- Bias is often desired for enhancement or detection!

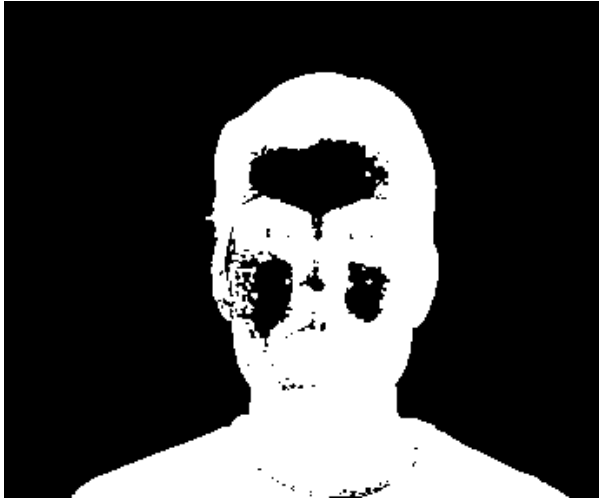
- Unbiased size-preserving smoothers

$$\mathit{close} - \mathit{open}(f, W) = \mathit{close}(\mathit{open}(f, W), W)$$

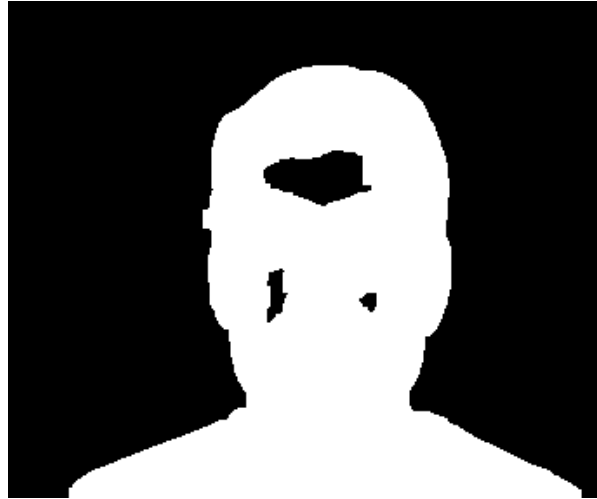
$$\mathit{open} - \mathit{close}(f, W) = \mathit{open}(\mathit{close}(f, W), W)$$

- *close-open* and *open-close* are duals, but not inverses of each other.

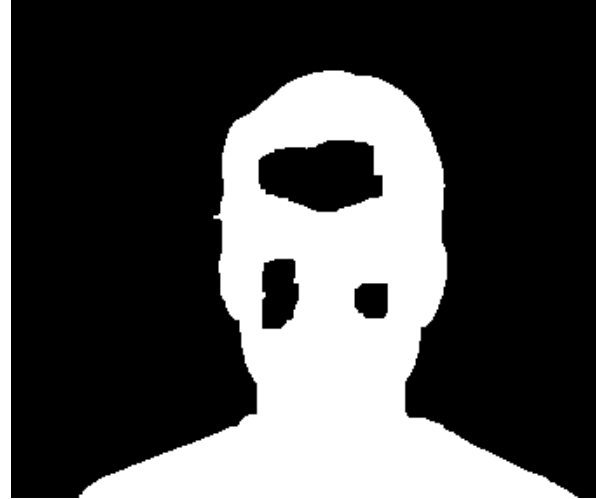
# Small hole removal by closing



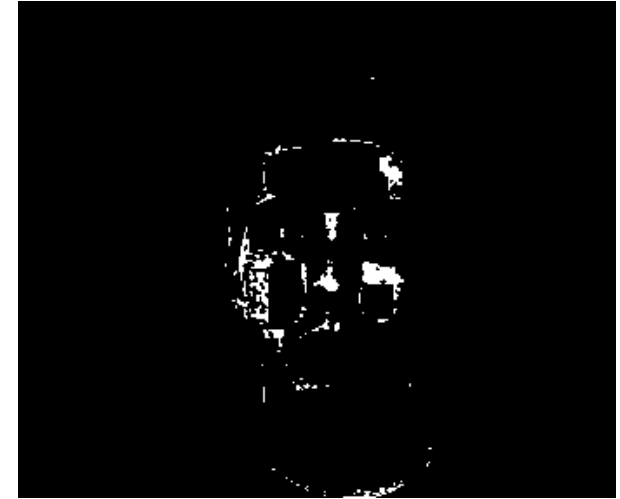
Original binary mask



Dilation  
10x10



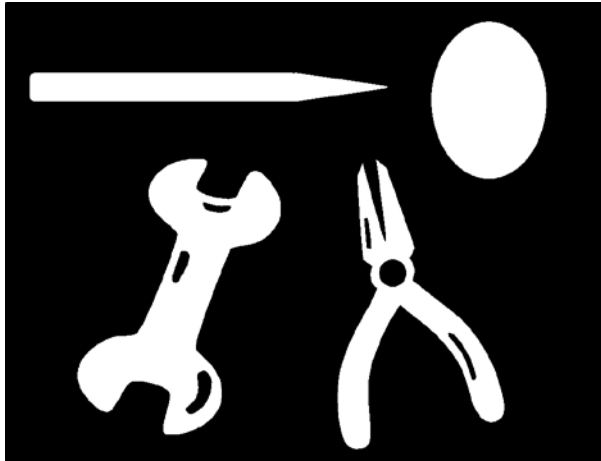
Closing 10x10



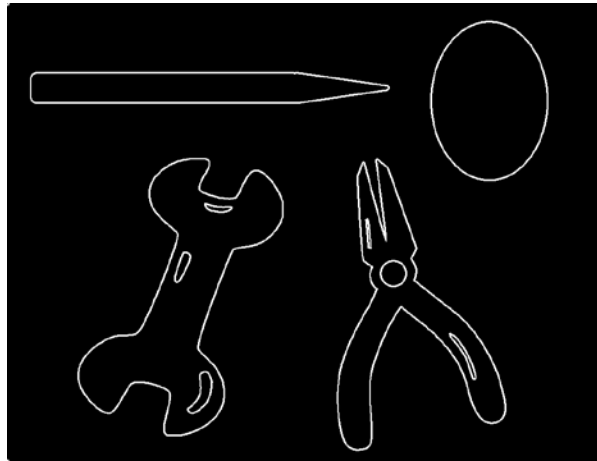
Difference to original mask



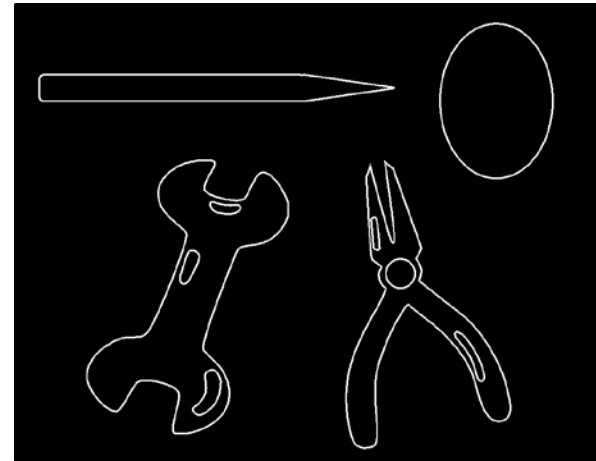
# Morphological edge detectors



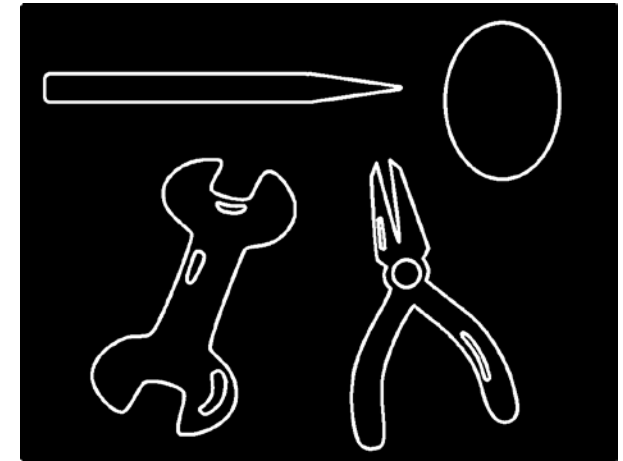
$f[x, y]$



$dilate(f, W) \neq f$



$erode(f, W) \neq f$



$dilate(f, W) \neq erode(f, W)$



# Recognition by erosion

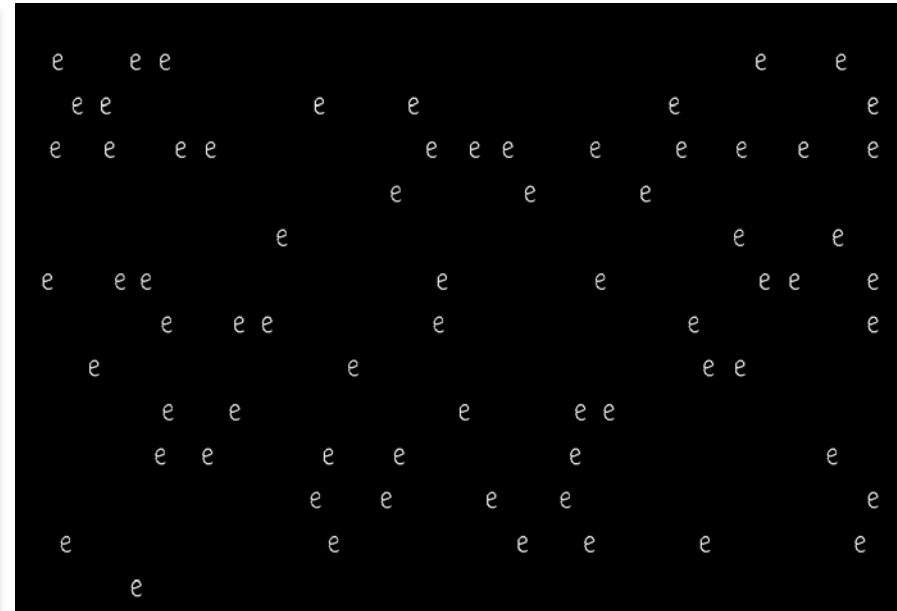
Binary image  $f$

$$\text{open}(\text{NOT}[f], W) = \text{dilate}(\text{erode}(\text{NOT}[f], W), W)$$

## INTEREST-POINT DETECTION

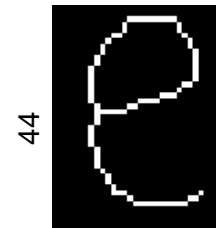
Feature extraction typically starts by finding the salient interest points in the image. For robust image matching, we desire interest points to be repeatable under perspective transformations (or, at least, scale changes, rotation, and translation) and real-world lighting variations. An example of feature extraction is illustrated in Figure 3. To achieve scale invariance, interest points are typically computed at multiple scales using an image pyramid [15]. To achieve rotation invariance, the patch around each interest point is canonically oriented in the direction of the dominant gradient. Illumination changes are compensated by normalizing the mean and standard deviation of the pixels of the gray values within each patch [16].

1400



2000

Structuring  
element  $W$



44

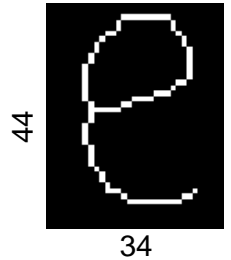
34

# Recognition by erosion

## INTEREST-POINT DETECTION

Feature extraction typically starts by finding the salient interest points in the image. For robust image matching, we desire interest points to be repeatable under perspective transformations (or, at least, scale changes, rotation, and translation) and real-world lighting variations. An example of feature extraction is illustrated in Figure 3. To achieve scale invariance, interest points are typically computed at multiple scales using an image pyramid [15]. To achieve rotation invariance, the patch around each interest point is canonically oriented in the direction of the dominant gradient. Illumination changes are compensated by normalizing the mean and standard deviation of the pixels of the gray values within each patch [16].

Structuring  
element  $W$



# Recognition by erosion

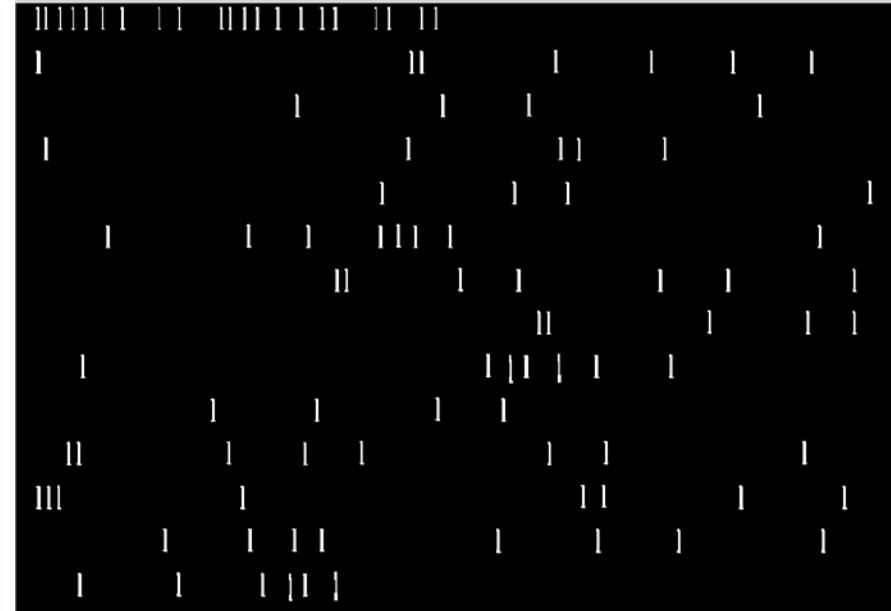
Binary image  $f$

$$\text{open}(\text{NOT}[f], W) = \text{dilate}(\text{erode}(\text{NOT}[f], W), W)$$

INTEREST-POINT DETECTION  
Feature extraction typically starts by finding the salient interest points in the image. For robust image matching, we desire interest points to be repeatable under perspective transformations (or, at least, scale changes, rotation, and translation) and real-world lighting variations. An example of feature extraction is illustrated in Figure 3. To achieve scale invariance, interest points are typically computed at multiple scales using an image pyramid [15]. To achieve rotation invariance, the patch around each interest point is canonically oriented in the direction of the dominant gradient. Illumination changes are compensated by normalizing the mean and standard deviation of the pixels of the gray values within each patch [16].

1400

2000



Structuring element  $W$



# Recognition by erosion

## INTEREST-POINT DETECTION

Feature extraction typically starts by finding the salient interest points in the image. For robust image matching, we desire interest points to be repeatable under perspective transformations (or, at least, scale changes, rotation, and translation) and real-world lighting variations. An example of feature extraction is illustrated in Figure 3. To achieve scale invariance, interest points are typically computed at multiple scales using an image pyramid [15]. To achieve rotation invariance, the patch around each interest point is canonically oriented in the direction of the dominant gradient. Illumination changes are compensated by normalizing the mean and standard deviation of the pixels of the gray values within each patch [16].

Structuring  
element  $W$



# Hit-miss filter

Binary image  $f$

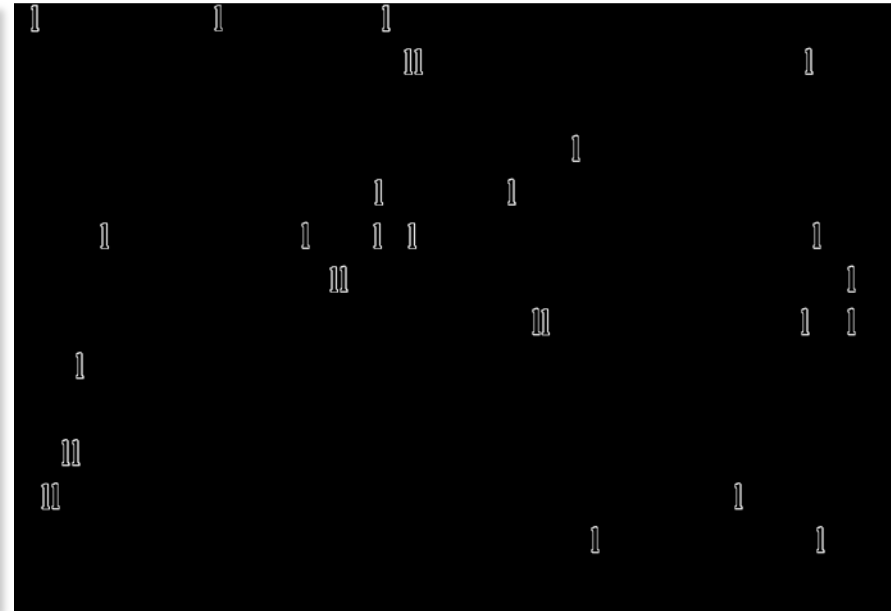
$$\text{dilate}\left(\text{erode}\left(\text{NOT}[f], V\right) \& \text{erode}\left(f, W\right), W\right)$$

1400

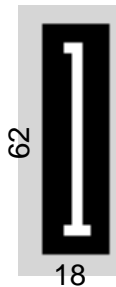
## INTEREST-POINT DETECTION

Feature extraction typically starts by finding the salient interest points in the image. For robust image matching, we desire interest points to be repeatable under perspective transformations (or, at least, scale changes, rotation, and translation) and real-world lighting variations. An example of feature extraction is illustrated in Figure 3. To achieve scale invariance, interest points are typically computed at multiple scales using an image pyramid [15]. To achieve rotation invariance, the patch around each interest point is canonically oriented in the direction of the dominant gradient. Illumination changes are compensated by normalizing the mean and standard deviation of the pixels of the gray values within each patch [16].

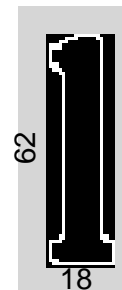
2000



Structuring  
element  $V$



Structuring  
element  $W$

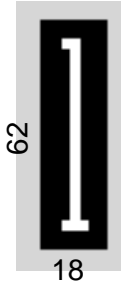


# Hit-miss filter

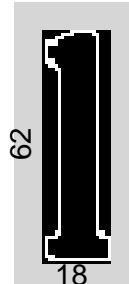
## INTEREST-POINT DETECTION

Feature extraction typically starts by finding the salient interest points in the image. For robust image matching, we desire interest points to be repeatable under perspective transformations (or, at least, scale changes, rotation, and translation) and real-world lighting variations. An example of feature extraction is illustrated in Figure 3. To achieve scale invariance, interest points are typically computed at multiple scales using an image pyramid [15]. To achieve rotation invariance, the patch around each interest point is canonically oriented in the direction of the dominant gradient. Illumination changes are compensated by normalizing the mean and standard deviation of the pixels of the gray values within each patch [16].

Structuring element  $V$



Structuring element  $W$



# Morphological filters for gray-level images

- Threshold sets of a gray-level image  $f[x, y]$

$$T_{\theta}(f[x, y]) = \{[x, y] : f[x, y] \geq \theta\}, \quad -\infty < \theta < +\infty$$

- Reconstruction of original image from threshold sets

$$f[x, y] = \sup \left\{ \theta : [x, y] \in T_{\theta}(f[x, y]) \right\}$$

- Idea of morphological operators for multi-level (or continuous-amplitude) signals

- Decompose into threshold sets
- Apply binary morphological operator to each threshold set
- Reconstruct via supremum (maximum) operation
- Gray-level operators thus obtained: **flat operators**  
→ *Flat morphological operators and thresholding are commutative*

# Dilation/erosion for gray-level images

- Explicit decomposition into threshold sets not required in practice
- Flat dilation operator: local maximum over window  $W$

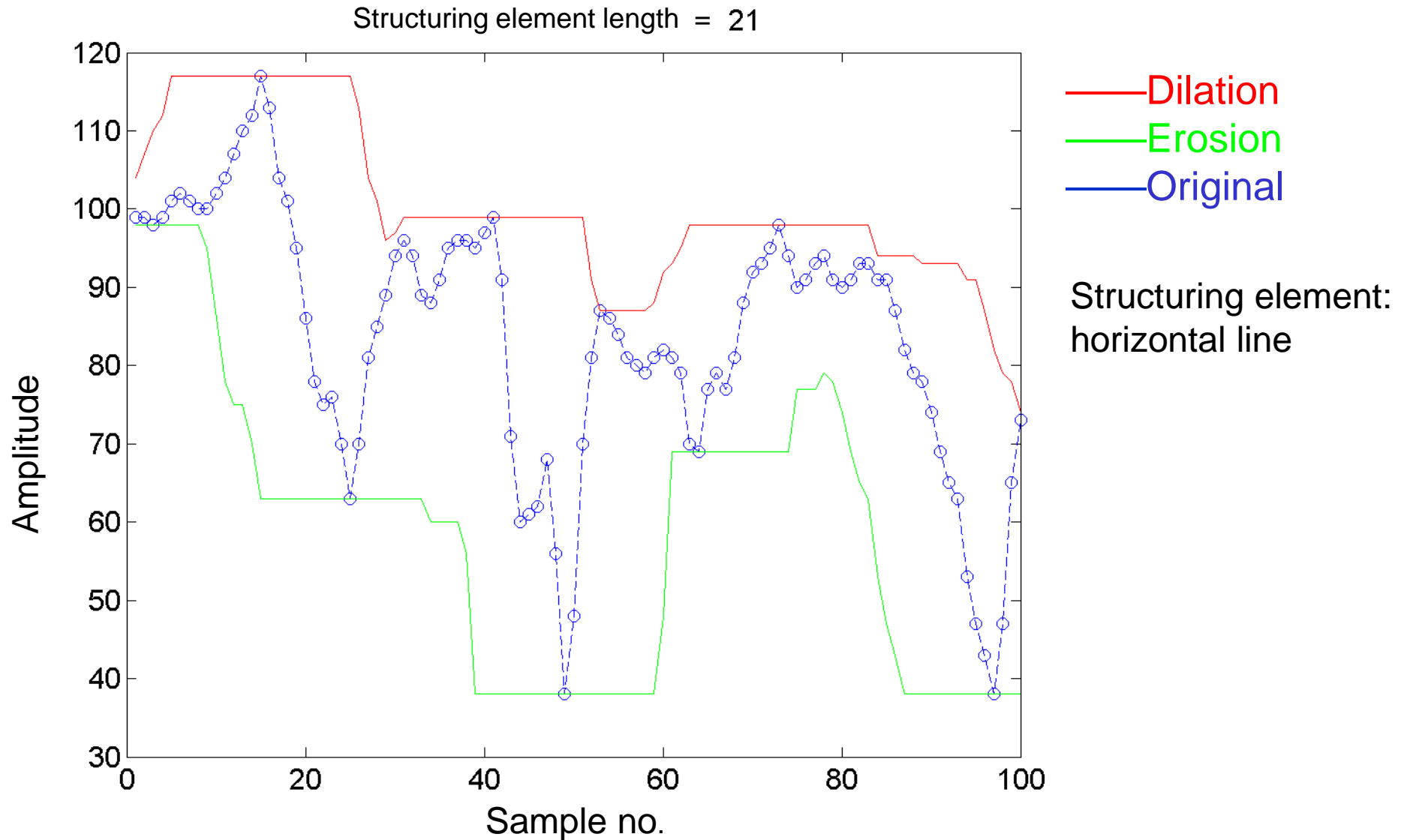
$$g[x, y] = \max \left\{ W^{(-)} \left\{ f[x, y] \right\} \right\} := \text{dilate}(f, W)$$

- Flat erosion operator: local minimum over window  $W$

$$g[x, y] = \min \left\{ W^{(+)} \left\{ f[x, y] \right\} \right\} := \text{erode}(f, W)$$

- Binary dilation/erosion operators contained as special case

# 1-d illustration of erosion and dilation



# Image example



Original



Dilation



Erosion



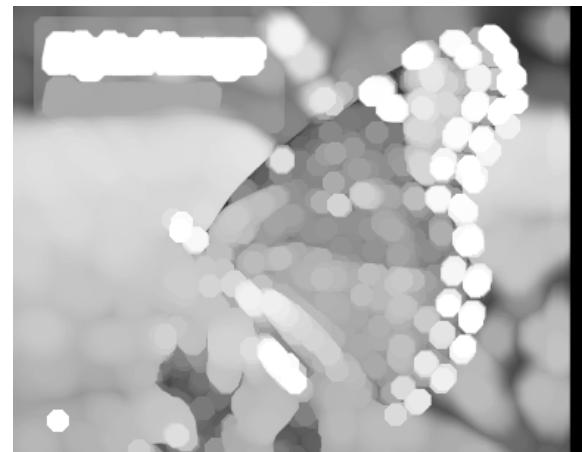
# Flat dilation with different structuring elements



Original



Diamond



Disk



20 degree line



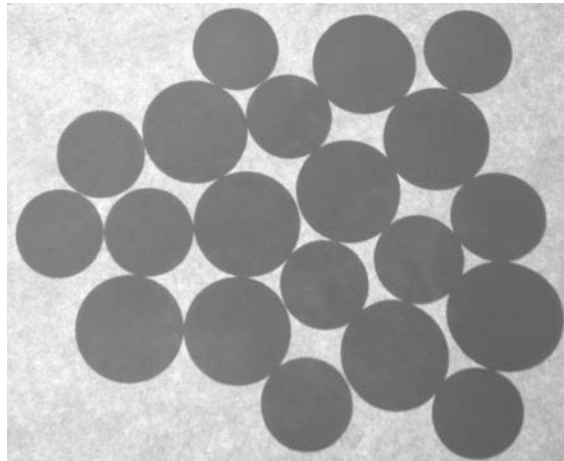
9 points



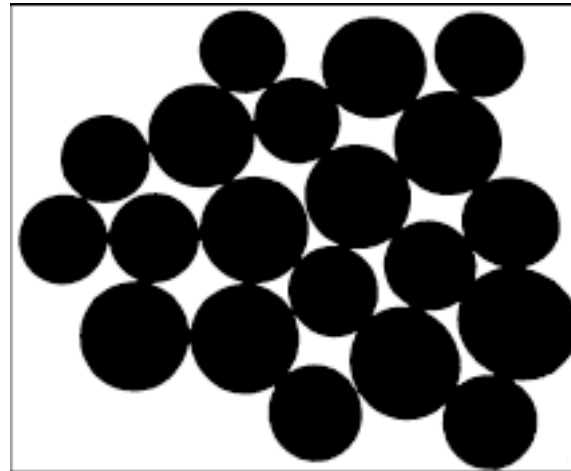
2 horizontal lines



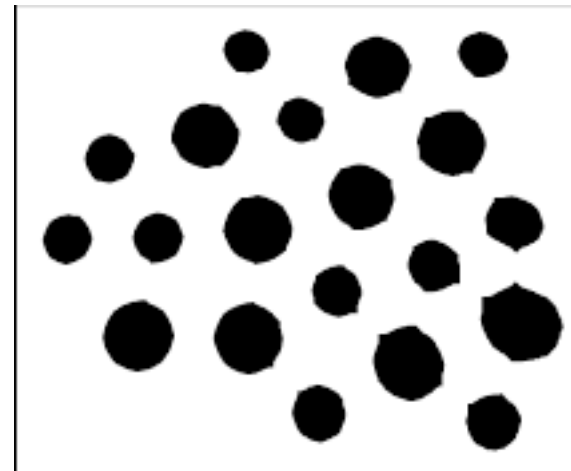
# Example: counting coins



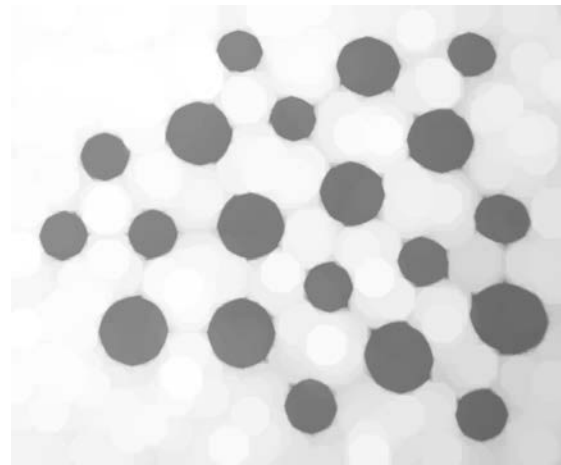
Original



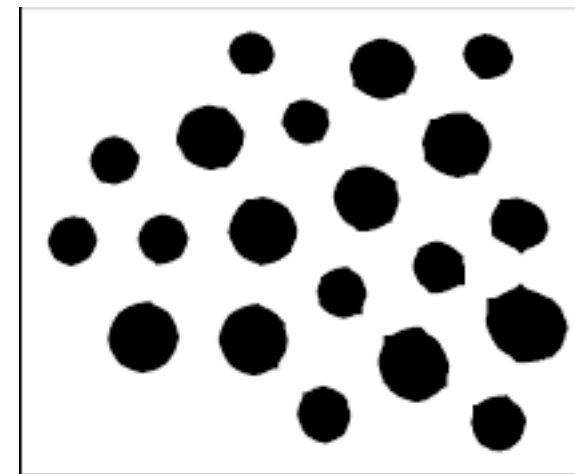
thresholded



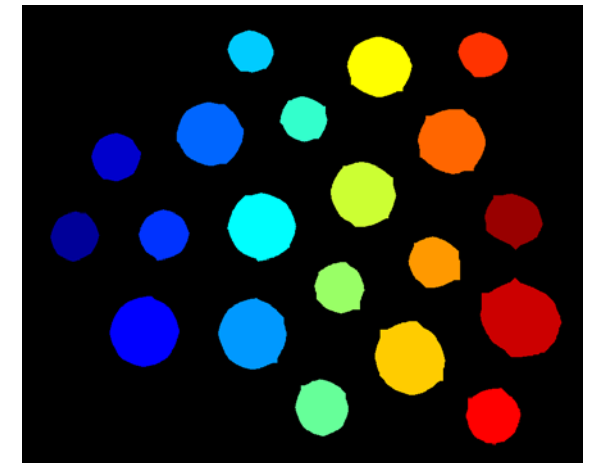
dilate after thresholding



dilation



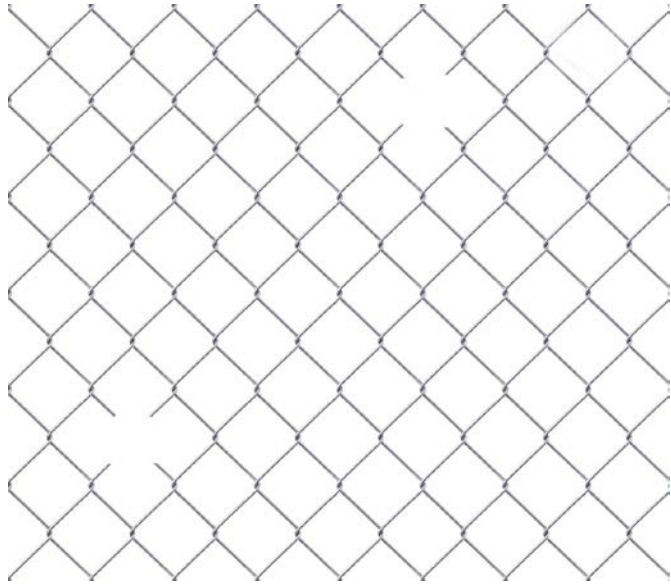
thresholded after dilation



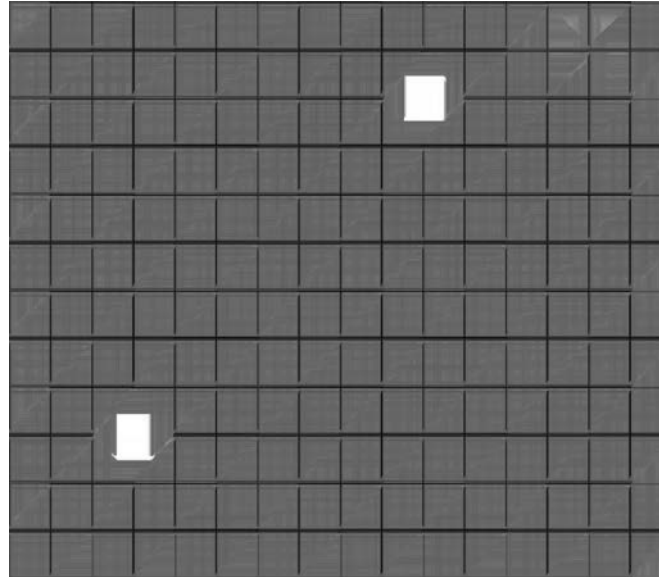
20 connected components



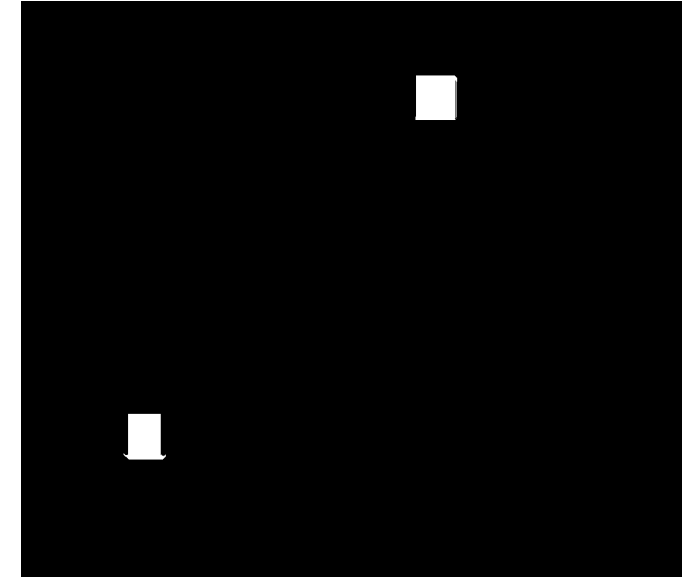
# Example: chain link fence hole detection



Original grayscale image  
*Fence (1023 x 1173)*



Flat erosion with 151x151  
"cross" structuring element



Binarized by Thresholding



# Morphological edge detector



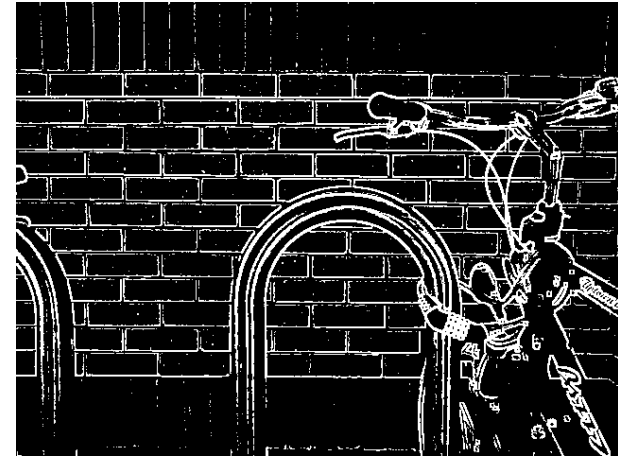
original  $f$



dilation  $g$



$g - f$



$g - f$  thresholded



# Beyond flat morphological operators

- General dilation operator

$$g[x, y] = \sup_{\alpha, \beta} \left\{ f[x - \alpha, y - \beta] + w[\alpha, \beta] \right\} = \sup_{\alpha, \beta} \left\{ w[x - \alpha, y - \beta] + f[\alpha, \beta] \right\}$$

- Like linear convolution, with sup replacing summation, addition replacing multiplication
- Dilation with “unit impulse”

$$d[\alpha, \beta] = \begin{cases} 0 & \alpha = \beta = 0 \\ -\infty & \text{else} \end{cases}$$

does not change input signal:

$$f[x, y] = \sup_{\alpha, \beta} \left\{ f[x - \alpha, y - \beta] + d[\alpha, \beta] \right\}$$

# Flat dilation as a special case

- Find  $w[\alpha, \beta]$  such that

$$f[x, y] = \sup_{\alpha, \beta} \left\{ f[x - \alpha, y - \beta] + w[\alpha, \beta] \right\} = \text{dilate}(f, W)$$

- Answer:

$$w[\alpha, \beta] = \begin{cases} 0 & [\alpha, \beta] \in \Pi_{xy} \\ -\infty & \text{else} \end{cases}$$

- Hence, write in general

$$\begin{aligned} g[x, y] &= \sup_{\alpha, \beta} \left\{ f[x - \alpha, y - \beta] + w[\alpha, \beta] \right\} \\ &= \text{dilate}(f, w) = \text{dilate}(w, f) \end{aligned}$$

# General erosion for gray-level images

- General erosion operator

$$g[x, y] = \inf_{\alpha, \beta} \left\{ f[x + \alpha, y + \beta] - w[\alpha, \beta] \right\} = \textit{erode}(f, w)$$

- Dual of dilation

$$g[x, y] = -\textit{dilate}(-f, w)$$

- Flat erosion contained as a special case

# Cascaded dilations



$f$



$dilate(f, w_1)$



$dilate(f, w_1, w_2)$



$dilate(f, w_1, w_2, w_3)$

$$dilate[dilate(f, w_1), w_2] = dilate(f, w)$$

$$\text{where } w = dilate(w_1, w_2)$$



# Cascaded erosions

- Cascaded erosions can be lumped into single erosion

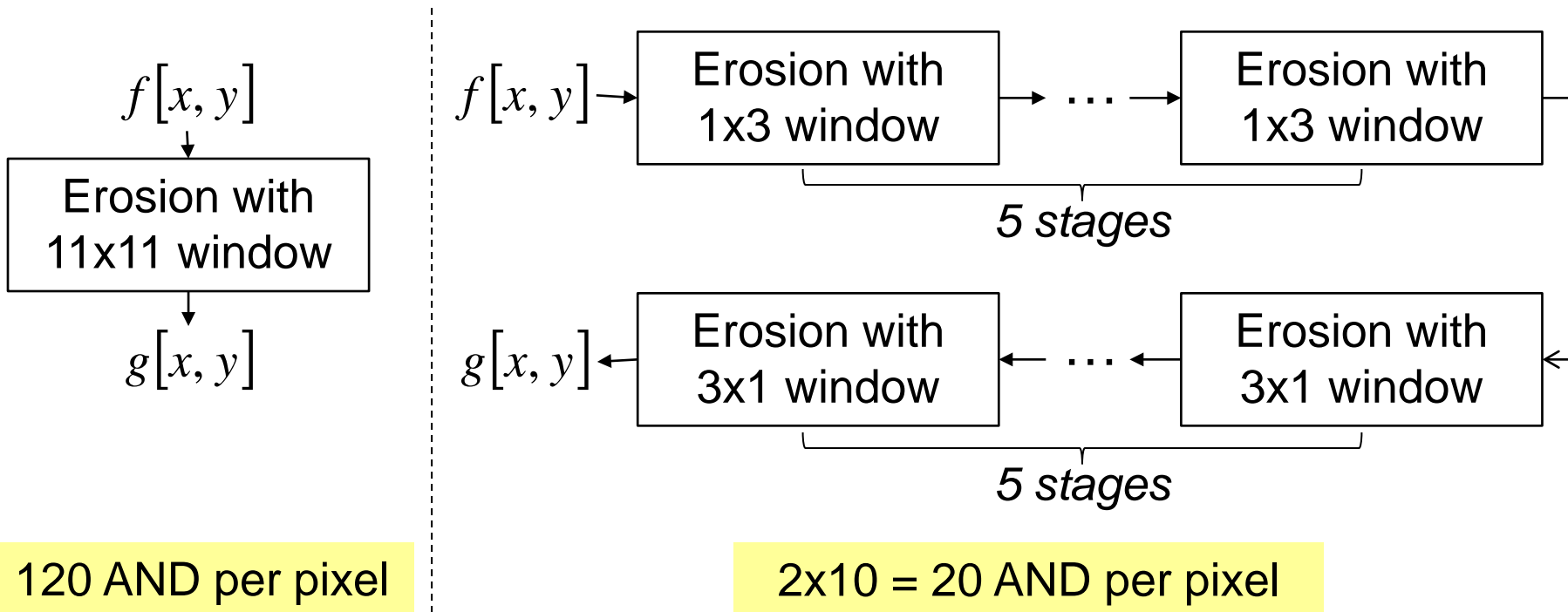
$$\begin{aligned} \text{erode}\left[\text{erode}(f, w_1), w_2\right] &= \text{erode}\left[-\text{dilate}(-f, w_1), w_2\right] \\ &= -\text{dilate}\left[\text{dilate}(-f, w_1), w_2\right] \\ &= -\text{dilate}(-f, w) \\ &= \text{erode}(f, w) \end{aligned}$$

$$\text{where } w = \text{dilate}(w_1, w_2)$$

- New structuring element (SE) is not the erosion of one SE by the other, but dilation.

# Fast dilation and erosion

- Idea: build larger dilation and erosion operators by cascading simple, small operators
- Example: binary erosion by 11x11 window



# Rank filters

- Generalisation of flat dilation/erosion: in lieu of min or max value in window, use the  $p$ -th ranked value
- Increases robustness against noise
- Best-known example: median filter for noise reduction
- Concept useful for both gray-level and binary images
- All rank filters are commutative with thresholding

# Median filter

- Gray-level median filter

$$g[x, y] = \mathit{median}\left[W \left\{ f[x, y] \right\}\right] := \mathit{median}(f, W)$$

- Binary images: majority filter

$$g[x, y] = \mathit{MAJ}\left[W \left\{ f[x, y] \right\}\right] := \mathit{majority}(f, W)$$

- Self-duality

$$\mathit{median}(f, W) = -\left[\mathit{median}(-f, W)\right]$$

$$\mathit{majority}(f, W) = \mathit{NOT}\left[\mathit{majority}(\mathit{NOT}[f], W)\right]$$

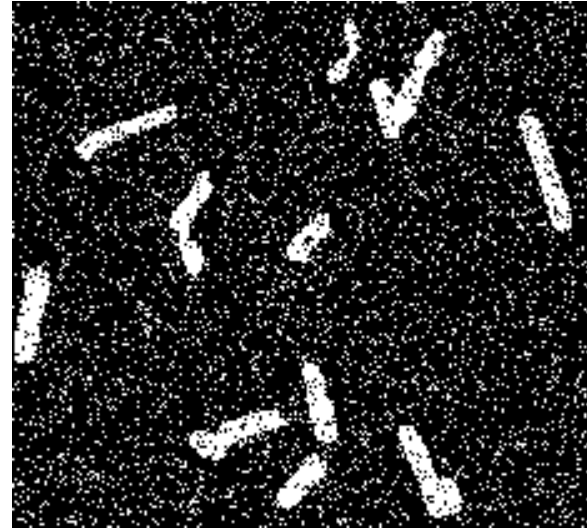
# Majority filter: example



Binary image with  
5% 'Salt&Pepper' noise



3x3 majority filter



20% 'Salt&Pepper' noise



3x3 majority filter



# Median filter: example



Original image



5% 'Salt&Pepper' noise



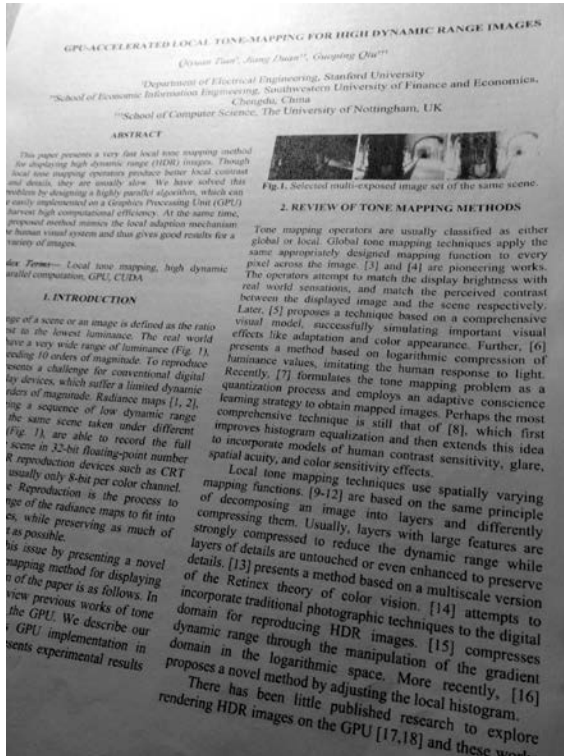
3x3 median filtering



7x7 median filtering



# Example: non-uniform lighting compensation



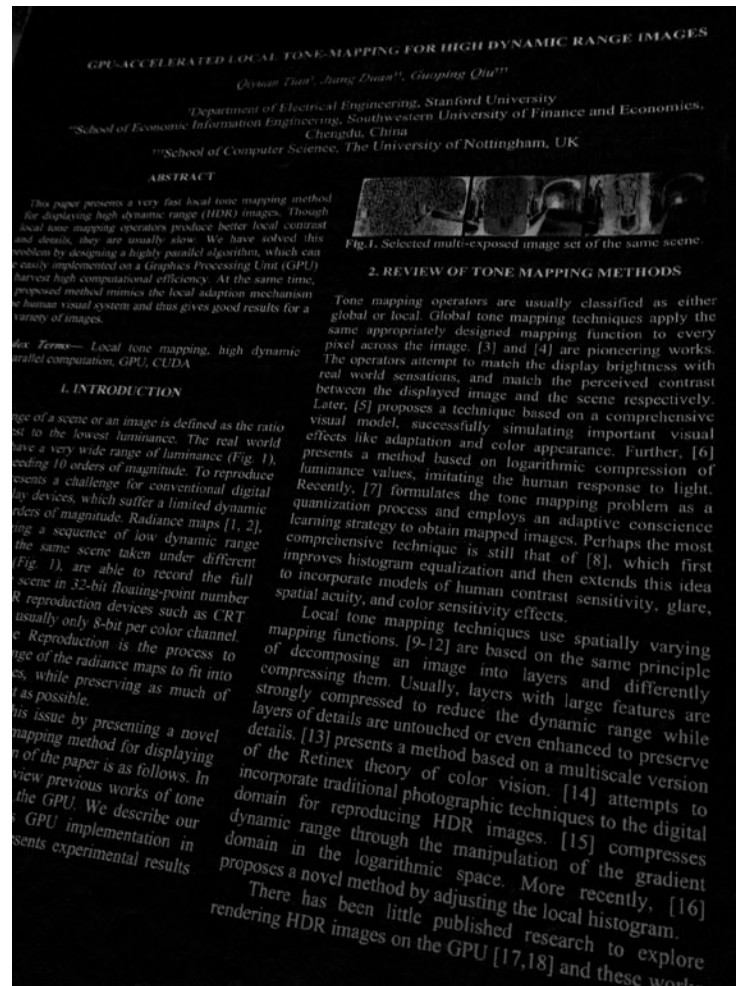
Original image  
1632x1216 pixels

Dilation (local max)  
61x61 structuring element

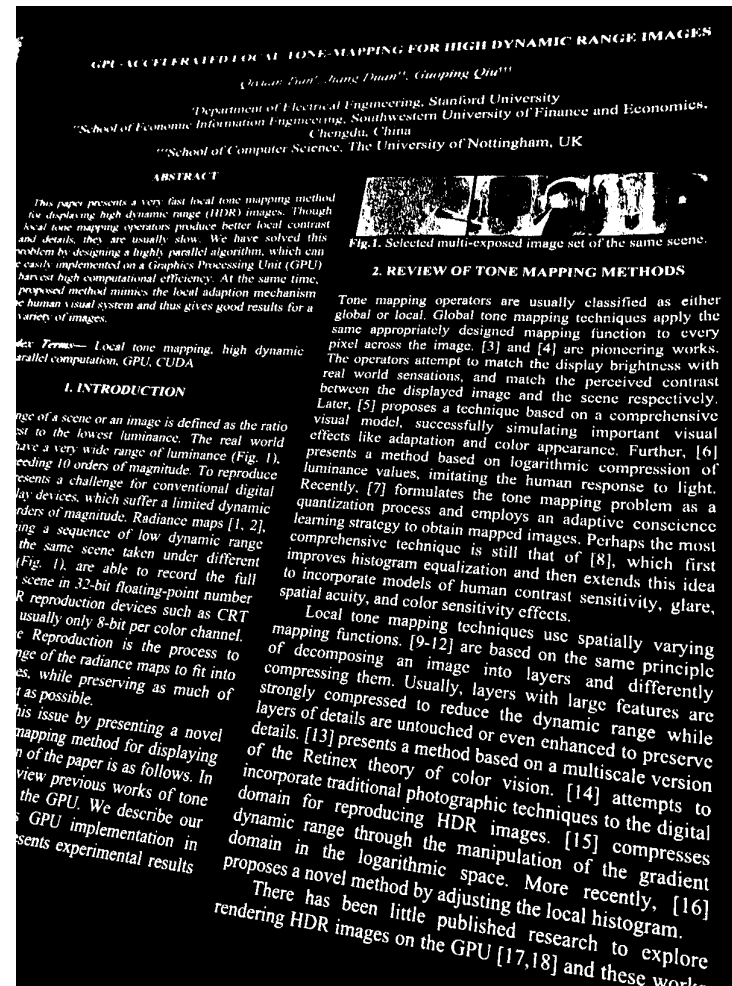
Rank filter  
10<sup>th</sup> brightest pixel  
61x61 structuring element



# Example: non-uniform lighting compensation



Background – original image



After global thresholding

