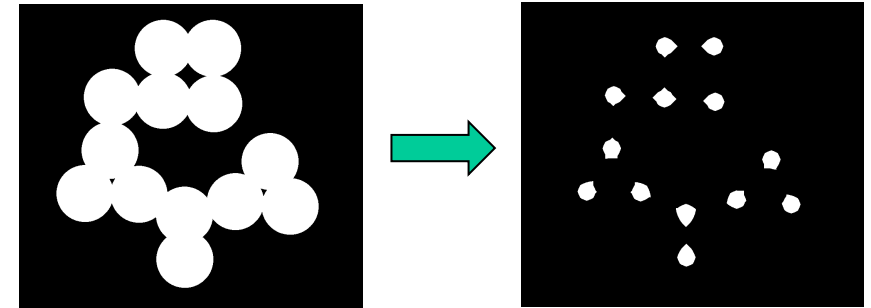


Morphological Image Processing

- Binary dilation and erosion
- Set-theoretic 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

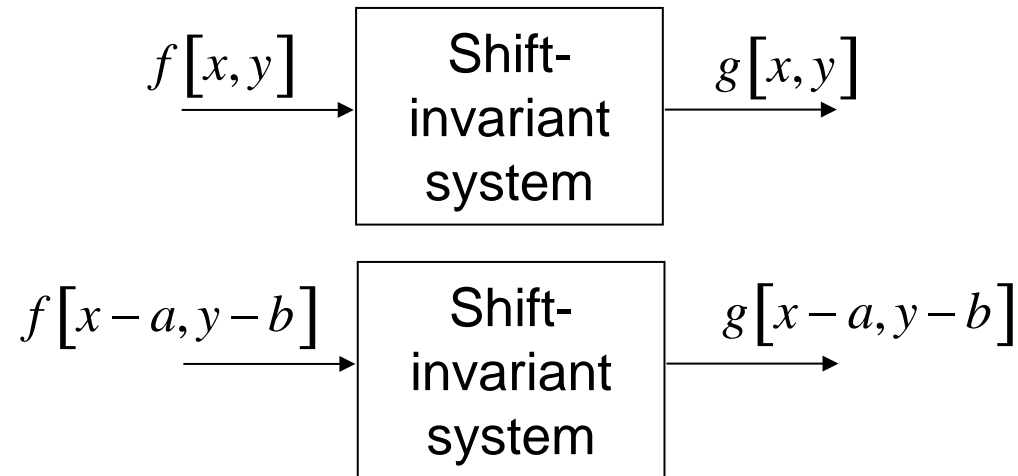
- Binary images are common
 - Intermediate abstraction in a gray-scale/color image analysis system
 - Thresholding/segmentation
 - Presence/absence of some image property
 - Text and line graphics, document image processing
- Representation of individual pixels as 0 or 1, convention:
 - foreground, object = 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

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

. . . then, for all integers a and b



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

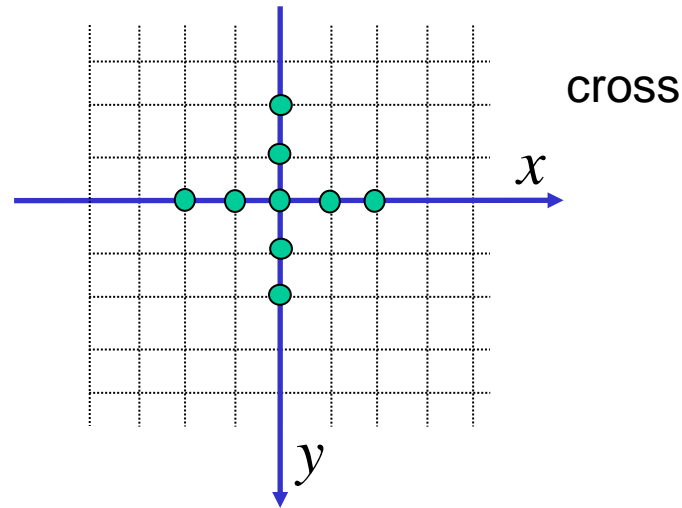
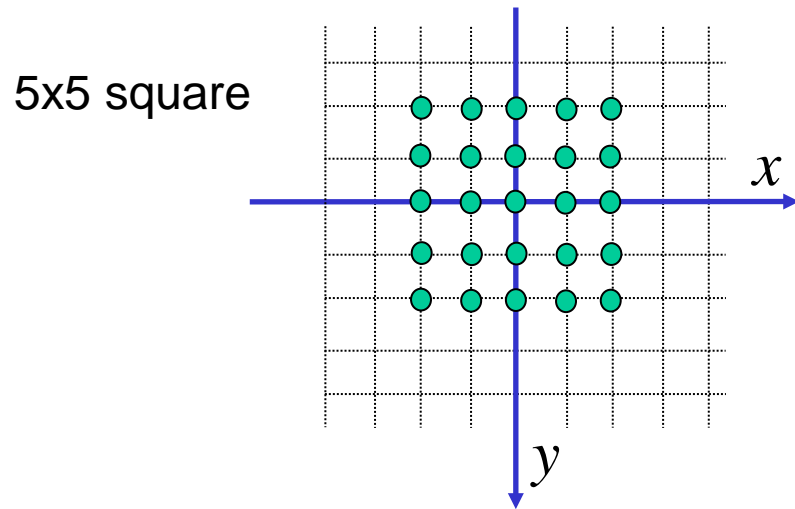
Structuring element

- Neighborhood “window” operator

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

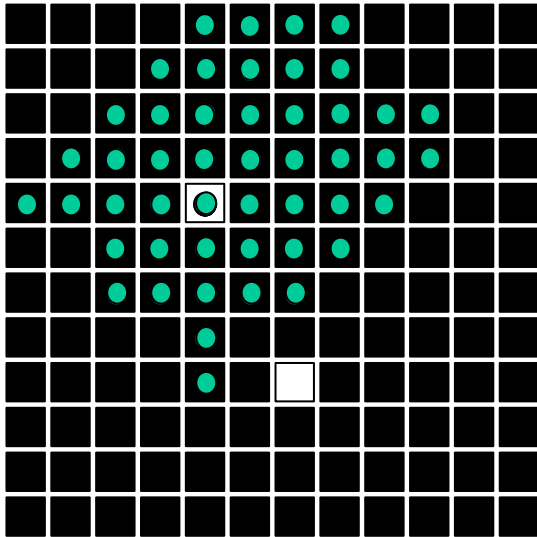
“structuring element”

- Example structuring elements Π_{xy} :

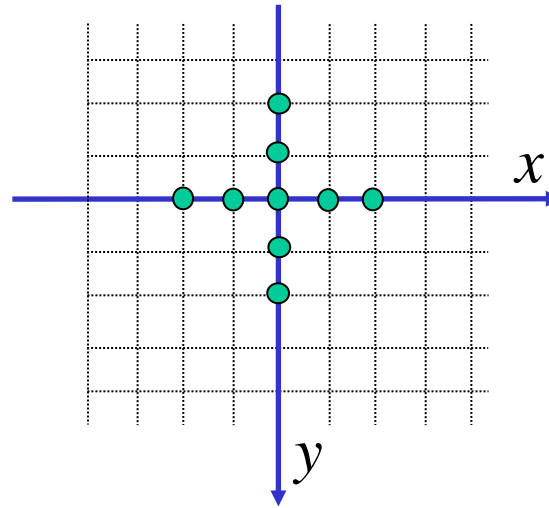


Binary dilation

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



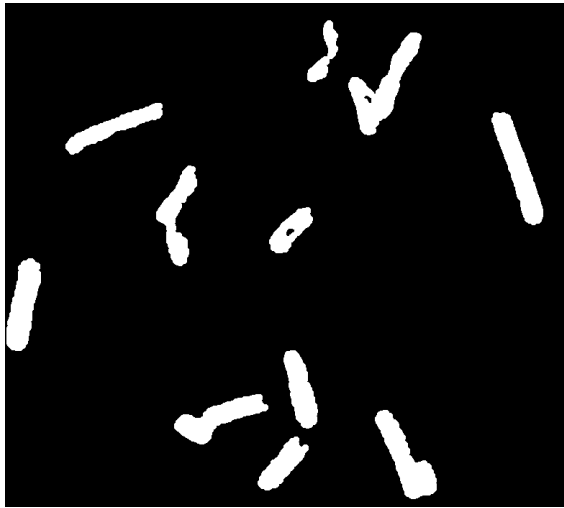
$f[x,y]$



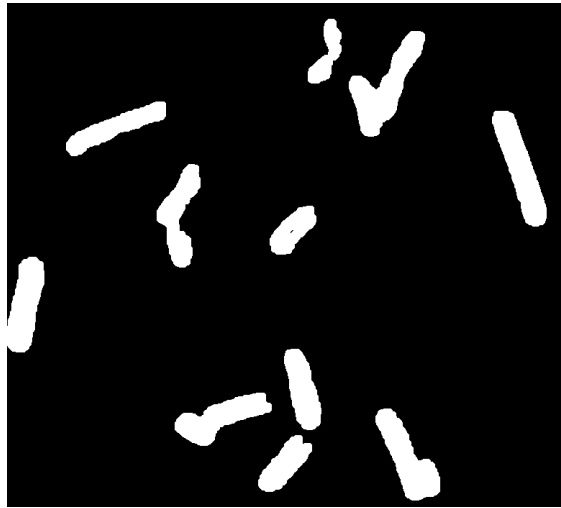
Π_{xy}

Binary dilation with square structuring element

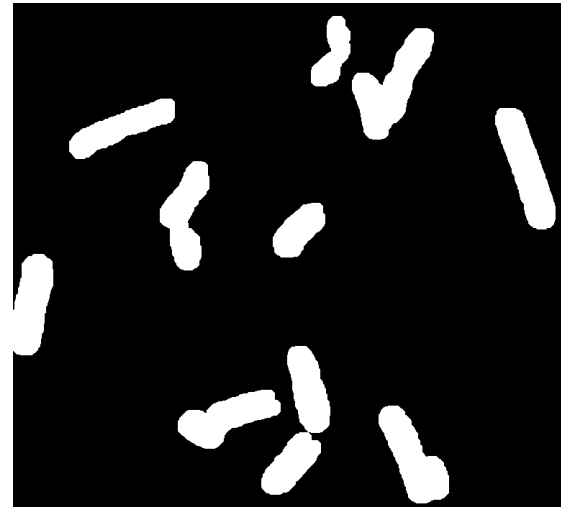
$$g[x,y] = OR[W \{f[x,y]\}] := dilate(f,W)$$



Original (701x781)



dilation with
3x3 structuring element



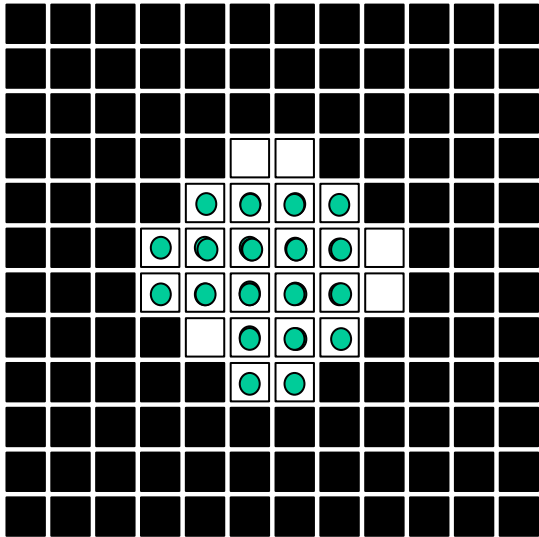
dilation with
7x7 structuring element

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

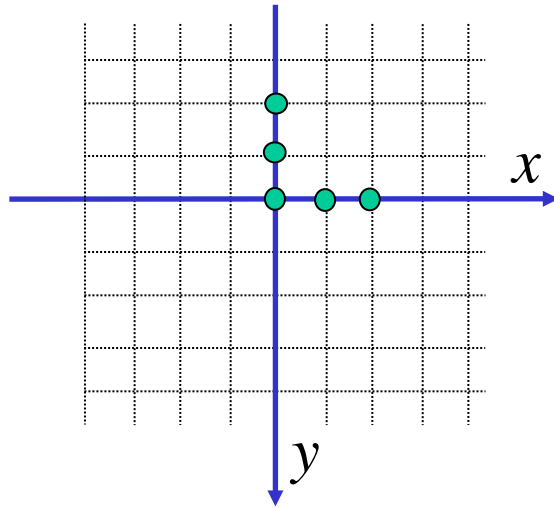


Binary erosion

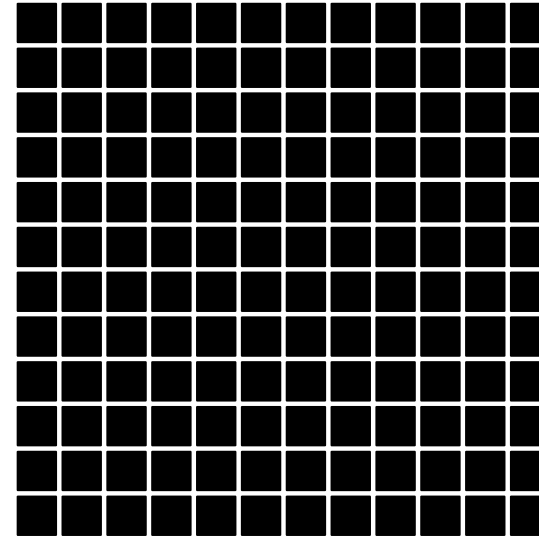
$$g[x, y] = \text{AND} \left[W \left\{ f[x, y] \right\} \right] := \text{erode}(f, W)$$



$f[x, y]$



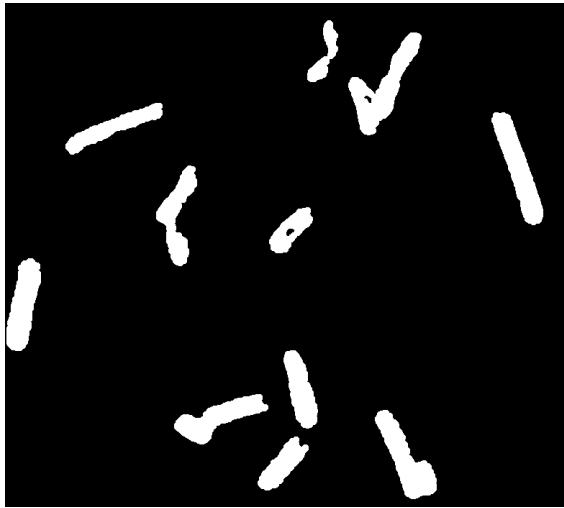
Π_{xy}



$g[x, y]$

Binary erosion with square structuring element

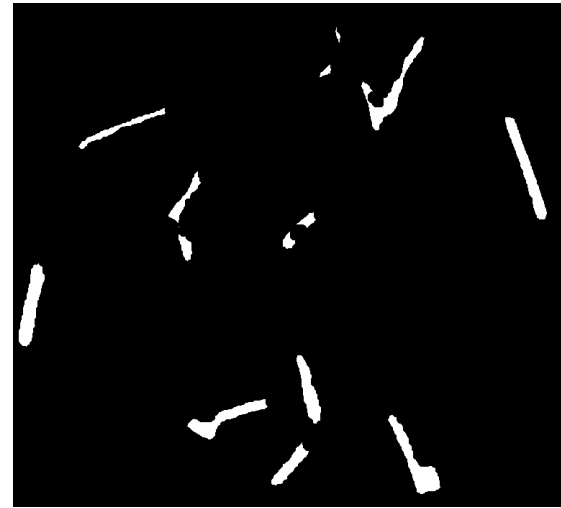
$$g[x, y] = \text{AND}\left[W \left\{ f[x, y] \right\}\right] := \text{erode}(f, W)$$



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

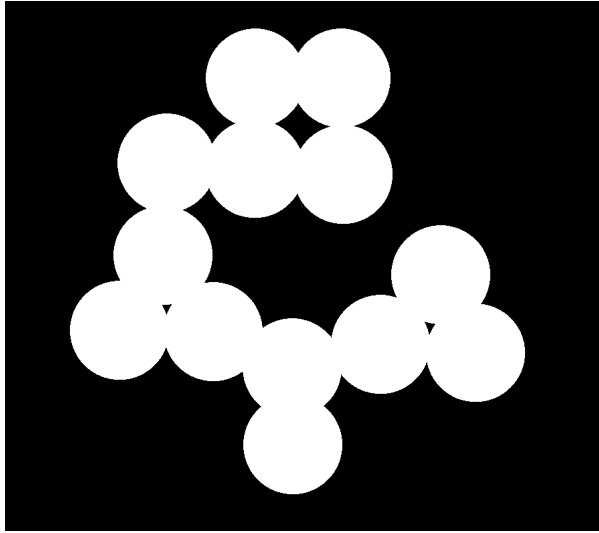
$$\text{dilate}(f, W) = \text{NOT}[\text{erode}(\text{NOT}[f], W)]$$

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

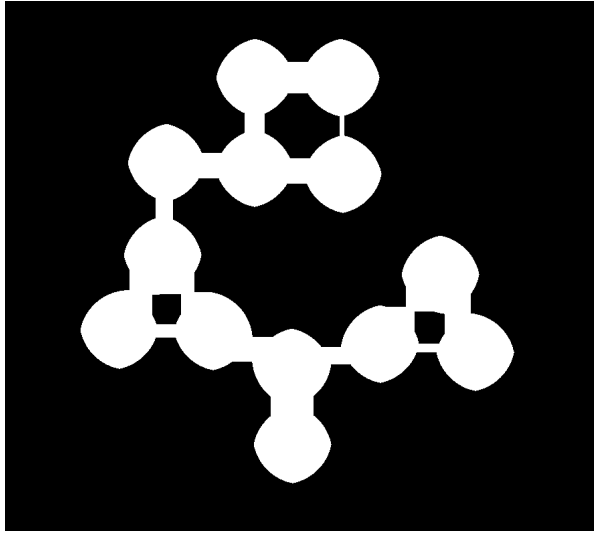
- But: erosion is not the inverse of dilation

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

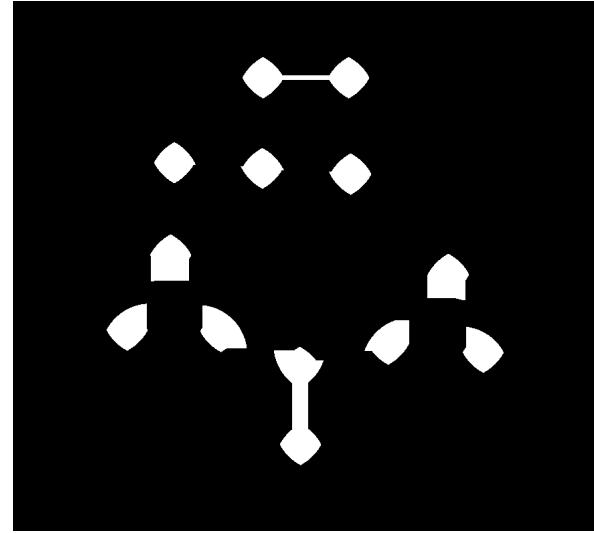
Example: blob separation/detection by erosion



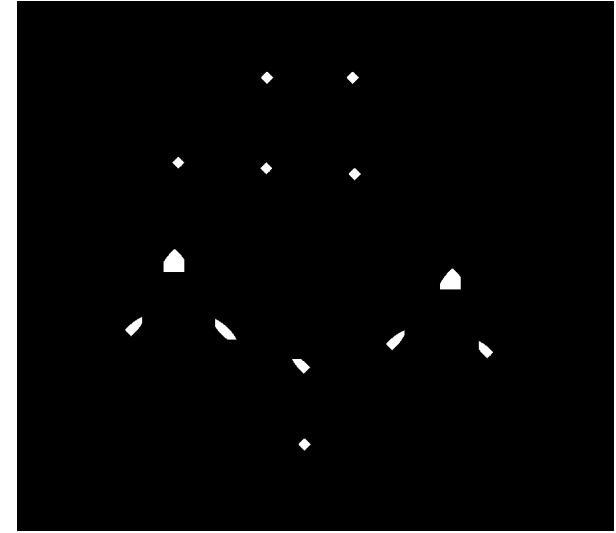
Original binary image
Circles (792x892)



Erosion by 30x30
structuring element



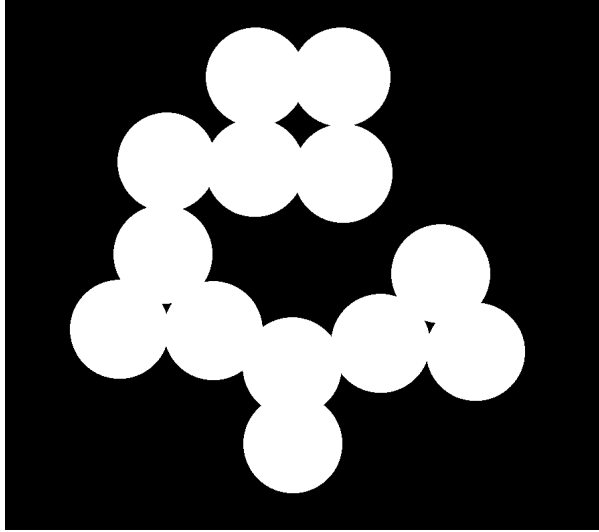
Erosion by 70x70
structuring element



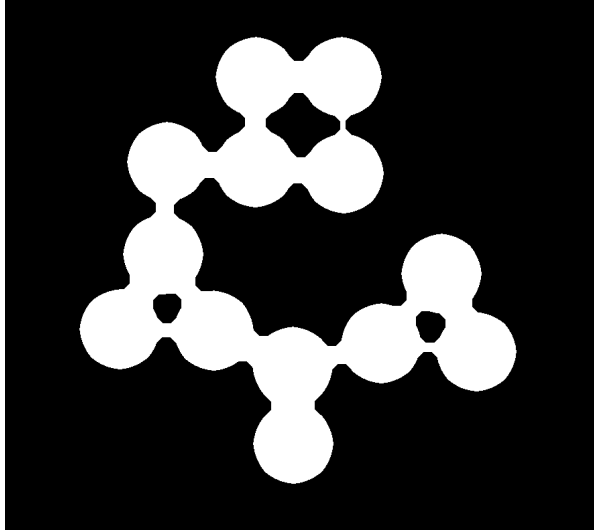
Erosion by 96x96
structuring element



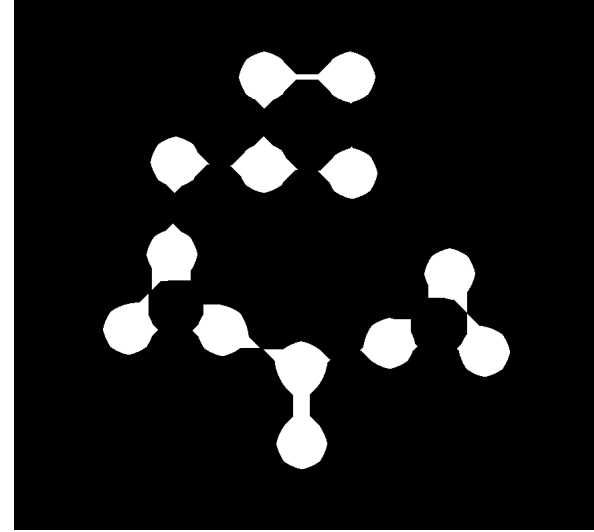
Example: blob separation/detection by erosion



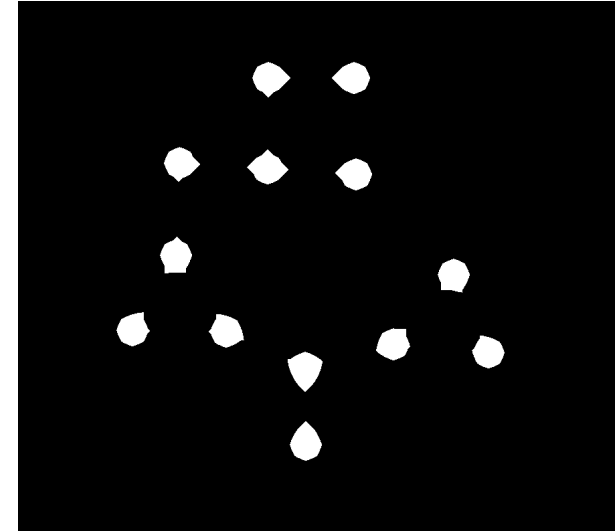
Original binary image
Circles (792x892)



Erosion by disk-shaped
structuring element
Diameter=15



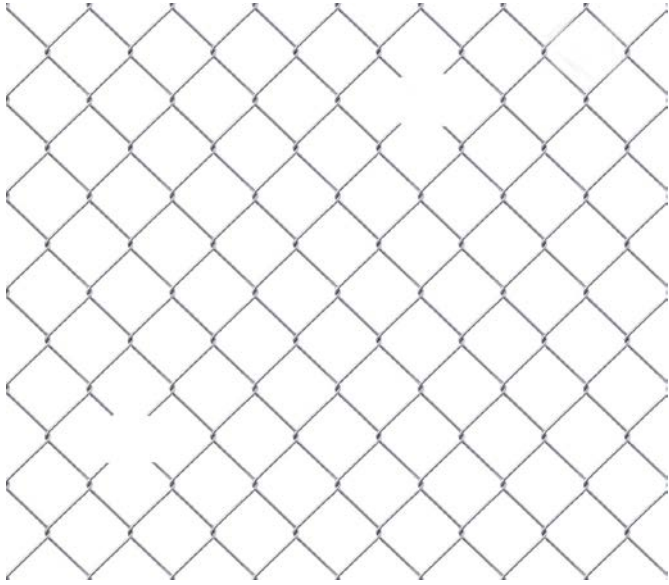
Erosion by disk-shaped
structuring element
Diameter=35



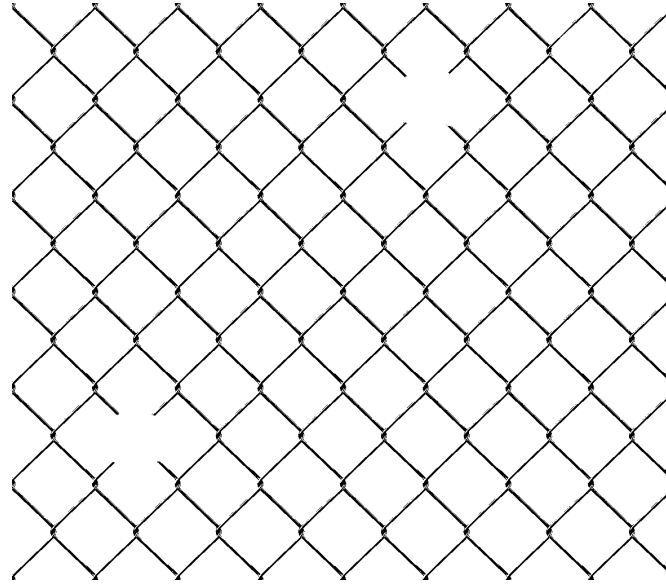
Erosion by disk-shaped
structuring element
Diameter=48



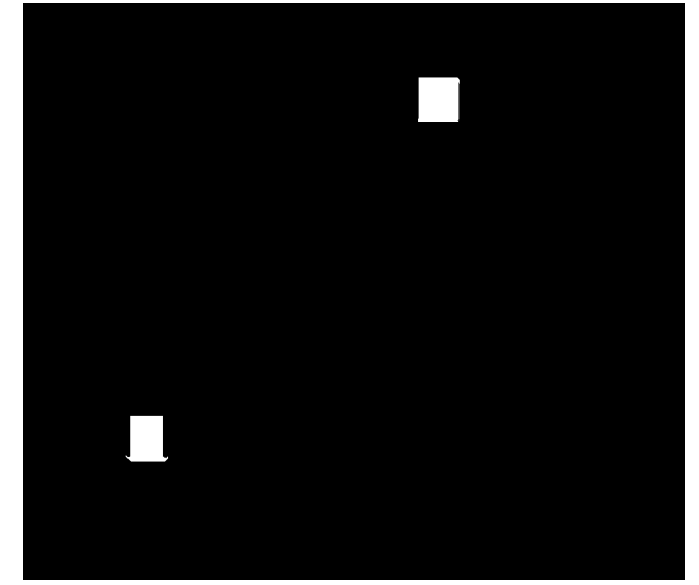
Example: chain link fence hole detection



Original grayscale image
Fence (1023 x 1173)



Fence thresholded
using Otsu's method



Erosion with 151x151
“cross” structuring element



Consider a binary image of infinite extent that is 0 everywhere, except for a single pixel $f[3,3]=1$. After performing dilation with an L-shaped structuring element (SE) of width 3 pixels, how does the resulting image look like?

- (a) All zero, the single pixel has been removed.
- (b) L-shaped SE shifted to position 3,3
- (c) L-shaped SE rotated by 180 degrees and shifted to position 3,3
- (d) L-shaped SE rotated by 180 degrees and shifted to position -3,-3
- (e) None of the above

Answer the same question, but now performing erosion in lieu of dilation.

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 Minkowski set addition

$$G = F \oplus \Pi_{xy}$$

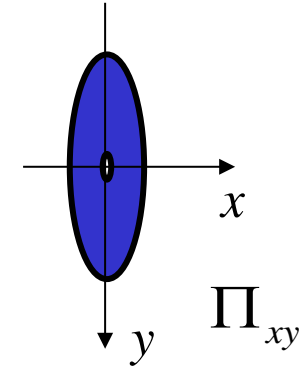
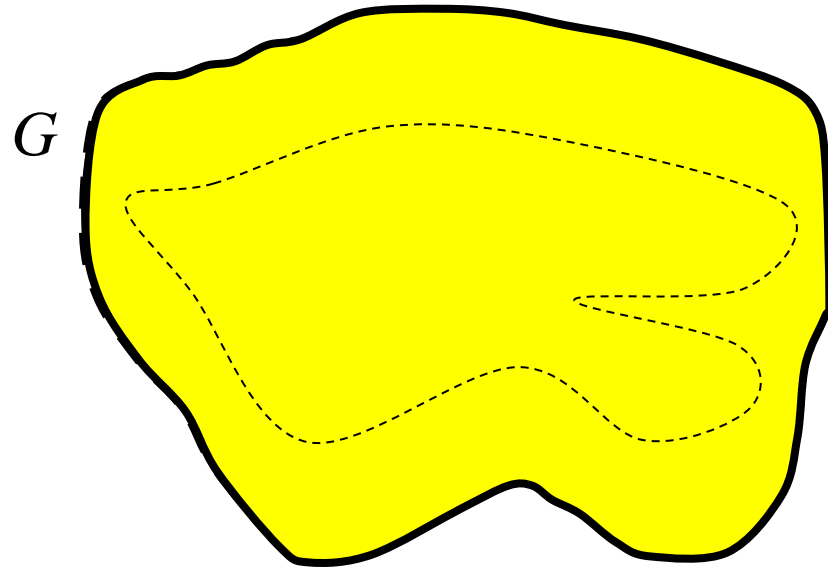
Commutative and associative!

$$= \left\{ (x + p_x, y + p_y) : (x, y) \in F, (p_x, p_y) \in \Pi_{xy} \right\}$$

$$= \bigcup_{(p_x, p_y) \in \Pi_{xy}} F_{+(p_x, p_y)}$$

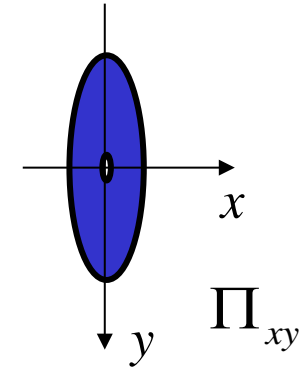
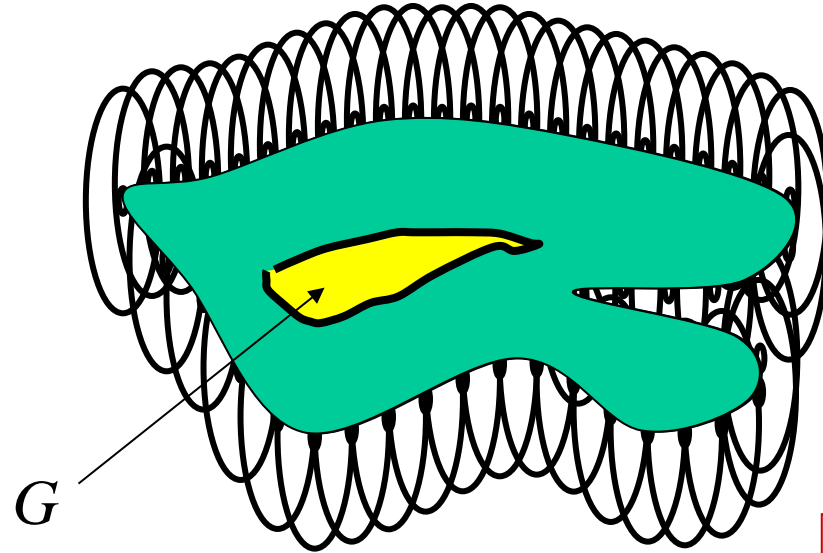
translation of F by vector (p_x, p_y)

Set-theoretic interpretation: dilation



$$\begin{aligned} G &= F \oplus \Pi_{xy} \\ &= \left\{ (x + p_x, y + p_y) : (x, y) \in F, (p_x, p_y) \in \Pi_{xy} \right\} \\ &= \bigcup_{(p_x, p_y) \in \Pi_{xy}} F_{+(p_x, p_y)} \end{aligned}$$

Set-theoretic interpretation: erosion



Not commutative!
Not associative!

Minkowski set subtraction

$$G = \bigcap_{(p_x, p_y) \in \Pi_{xy}} F_{+(p_x, p_y)} = F \ominus \Pi_{-xy}$$

Reversed structuring
element

Opening and closing

- Goal: smoothing without size change

- Open filter

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

- Close filter

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

- 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!

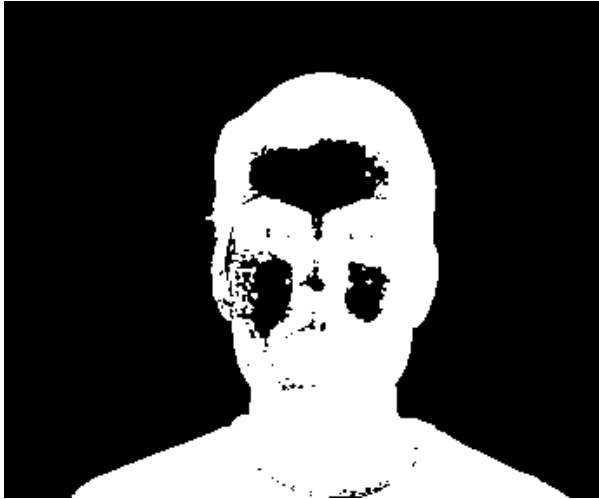
- Unbiased size-preserving smoothers

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

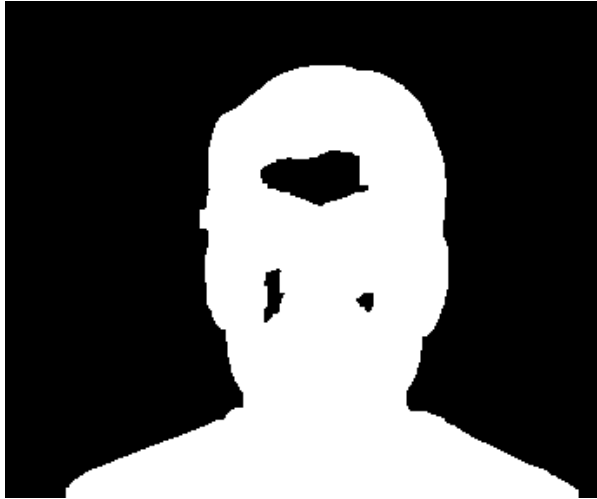
$$\textit{open} - \textit{close}(f, W) = \textit{open}(\textit{close}(f, W), W)$$

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

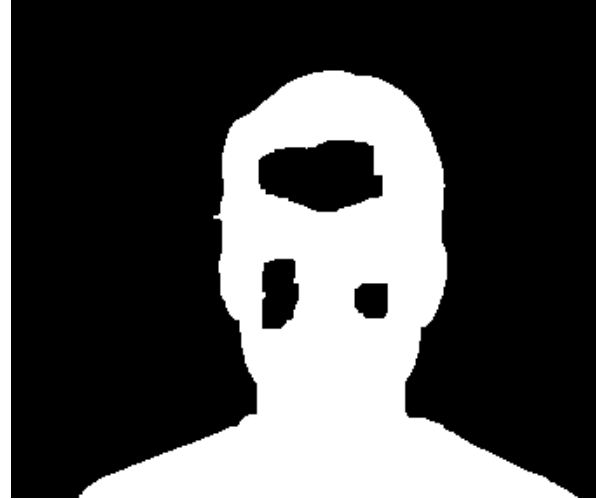
Small hole removal by closing



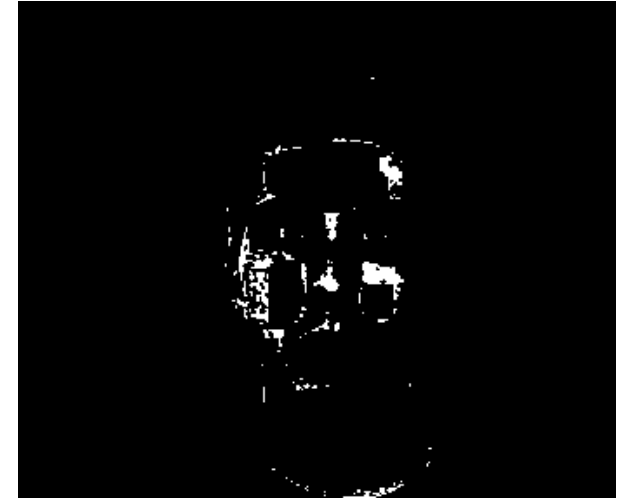
Original binary mask



Dilation
10x10



Closing 10x10



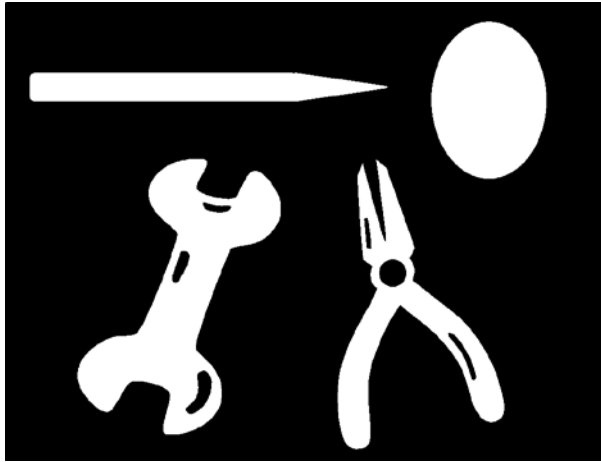
Difference to original mask



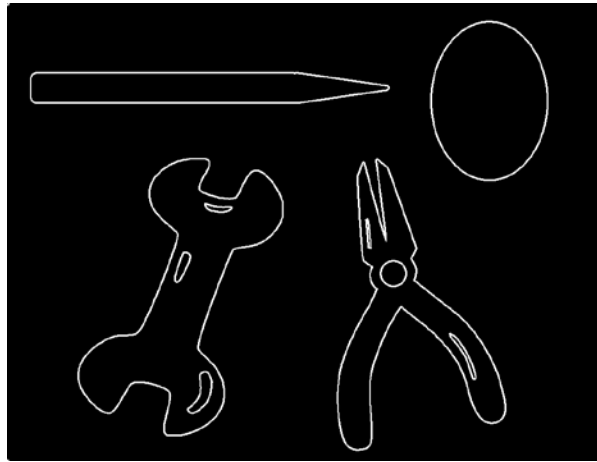
Which of the following holes in an object are filled by a morphological closing operation with a 10x10 square structuring element.

- (a) 5x5 square
- (b) 6x6 square
- (c) 9x9 square
- (d) 10x10 square
- (e) 10x10 square rotated by 45 degrees

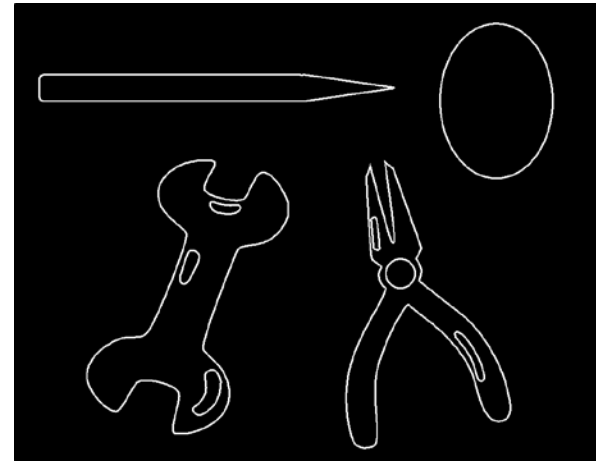
Morphological edge detectors



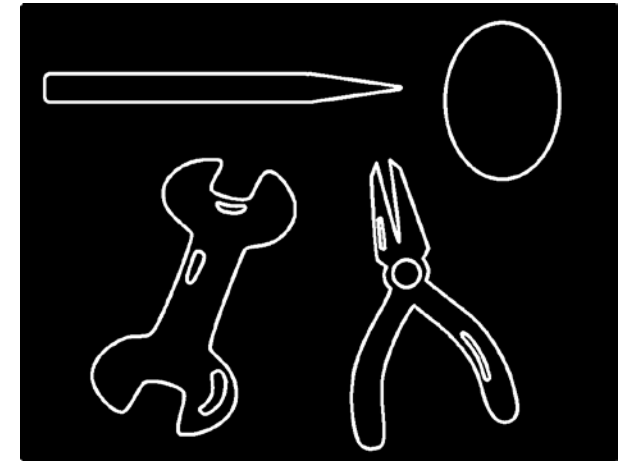
$f[x, y]$



$dilate(f, W) \neq f$



$erode(f, W) \neq f$



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



Recognition by erosion

Binary image f

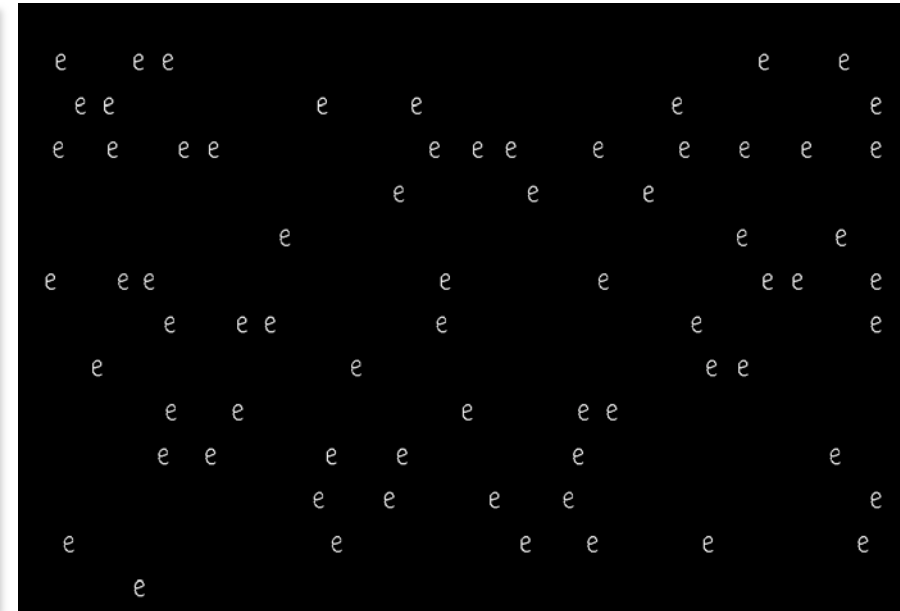
INTEREST-POINT DETECTION

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].

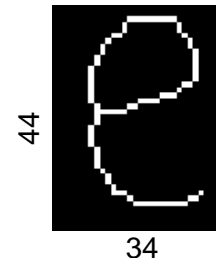
1400

2000

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



Structuring
element W



44

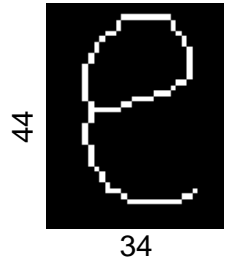
34

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



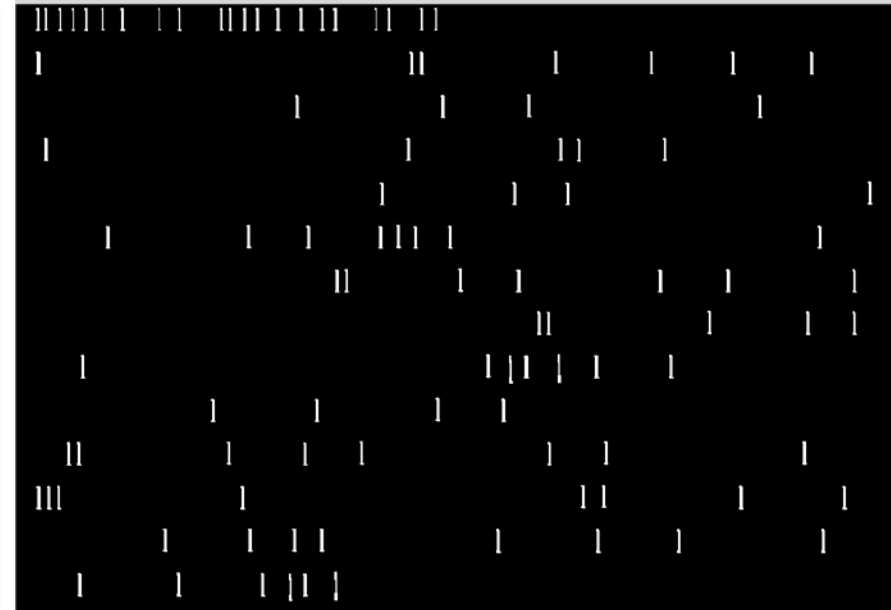
Recognition by erosion

Binary image f

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].

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



Structuring
element W



2000

1400

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



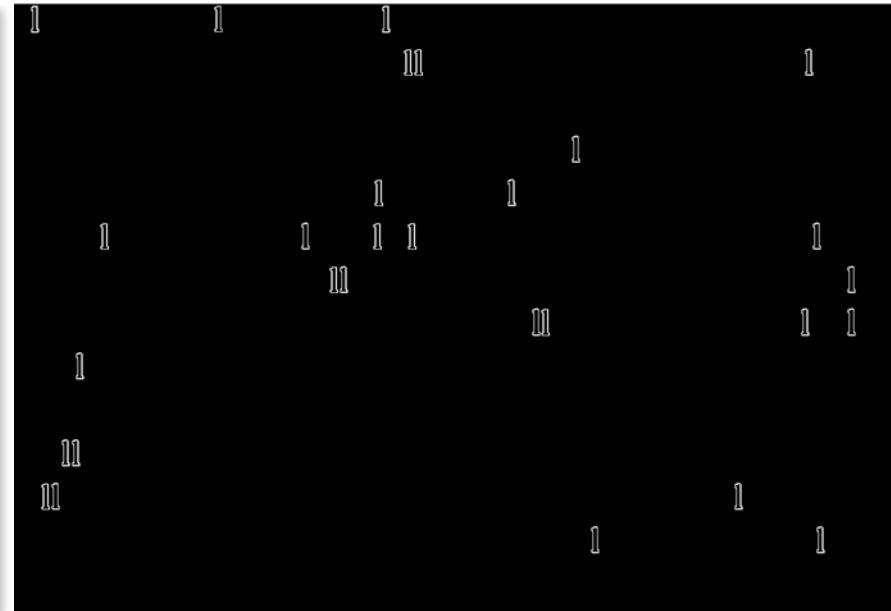
Hit-miss filter

Binary image f

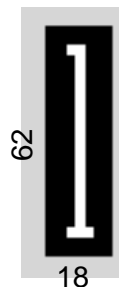
INTEREST-POINT DETECTION

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].

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



Structuring
element V



Structuring
element W



2000

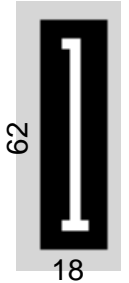
1400

Hit-miss filter

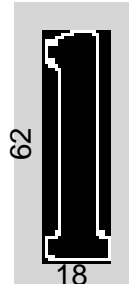
INTEREST-POINT DETECTION

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 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

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

- Flat erosion operator: local minimum over window W

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

- Binary dilation/erosion operators contained as special case

1-d illustration of erosion and dilation

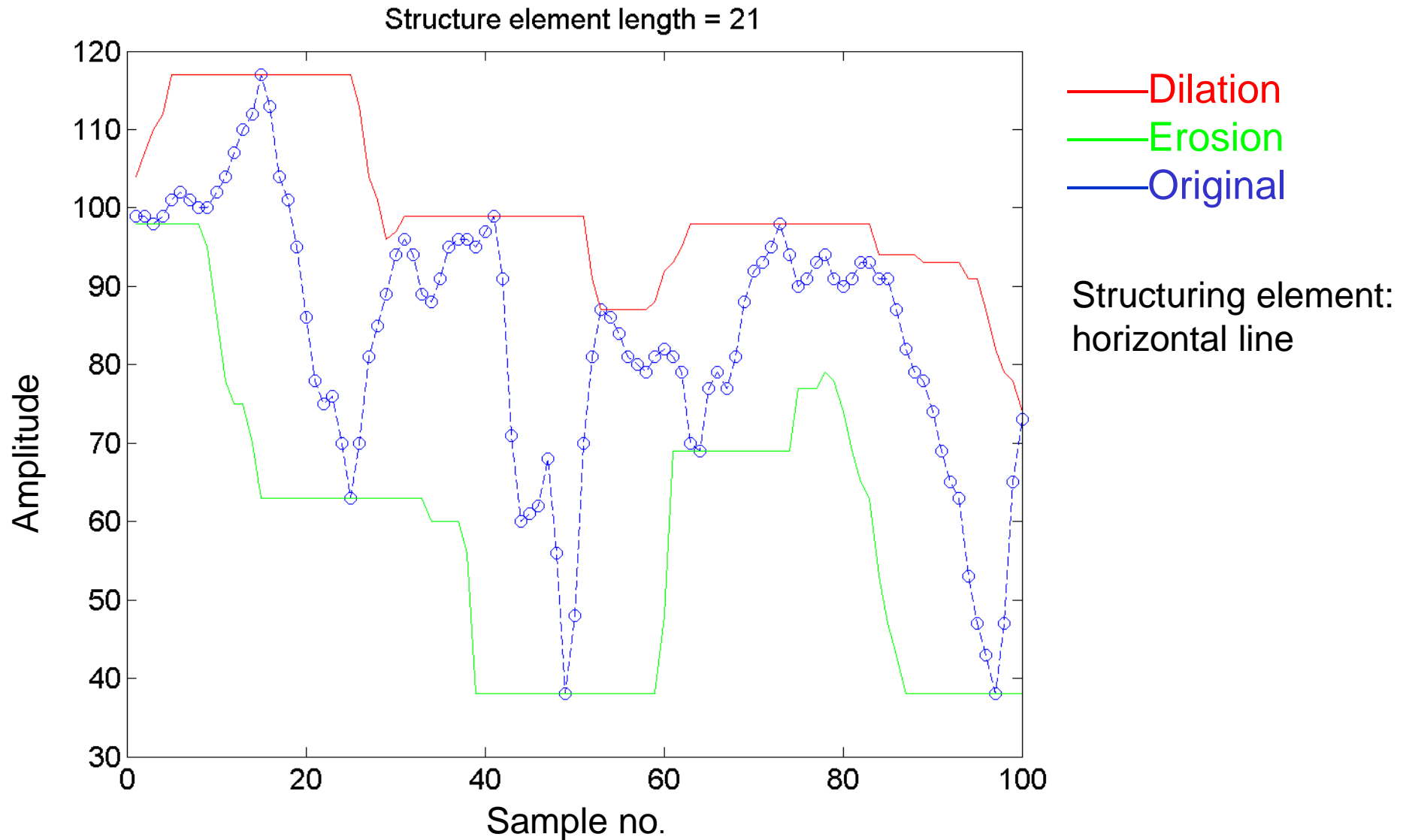
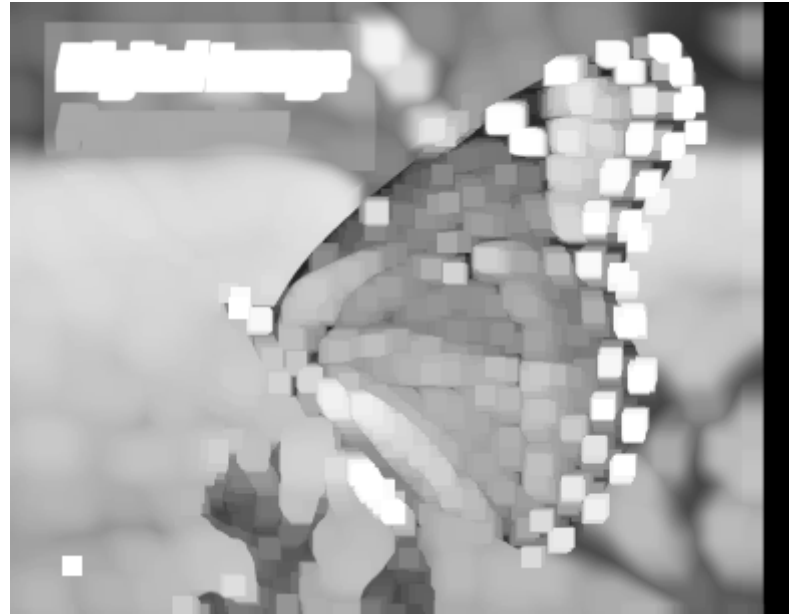


Image example



Original



Dilation



Erosion



Flat dilation with different structuring elements



Original



Diamond



Disk



20 degree line



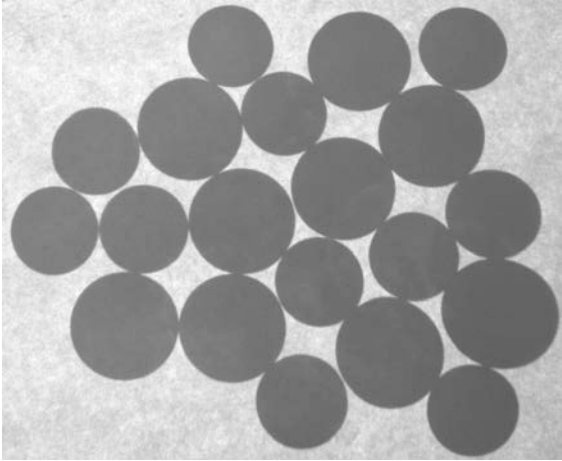
9 points



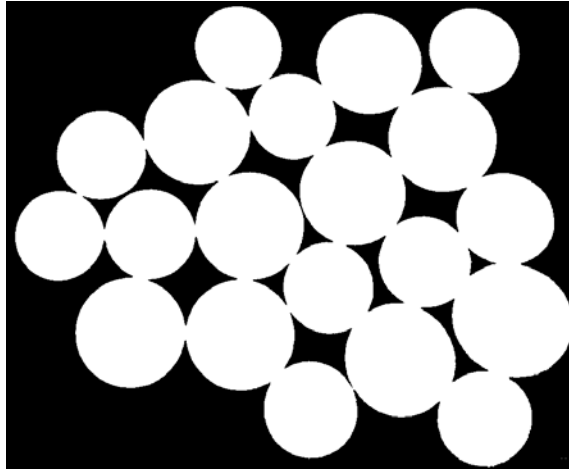
2 horizontal lines



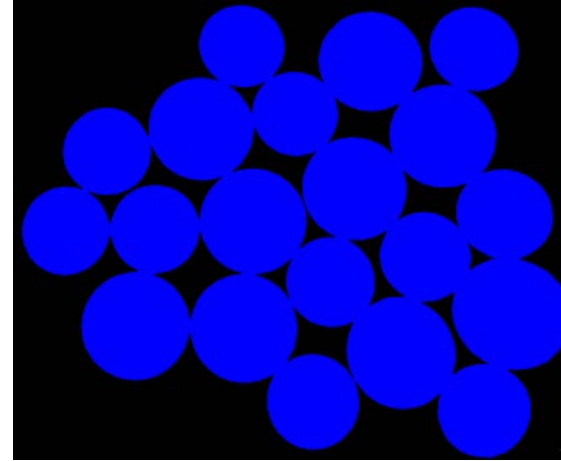
Example: counting coins



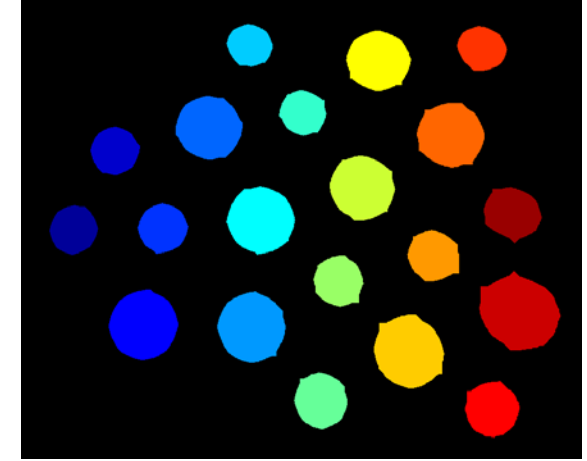
Original



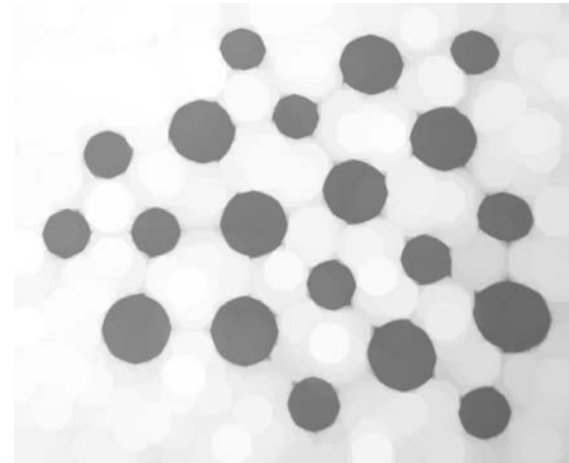
thresholded



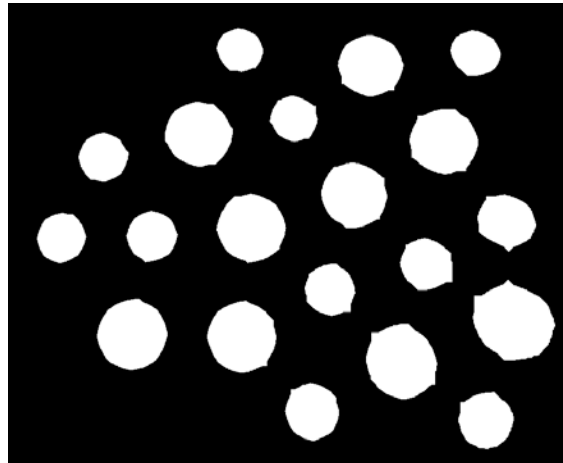
1 connected component



20 connected components



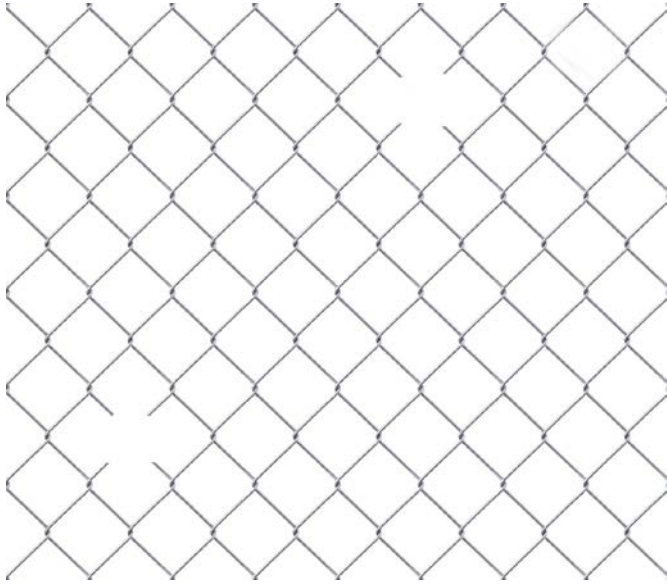
dilation



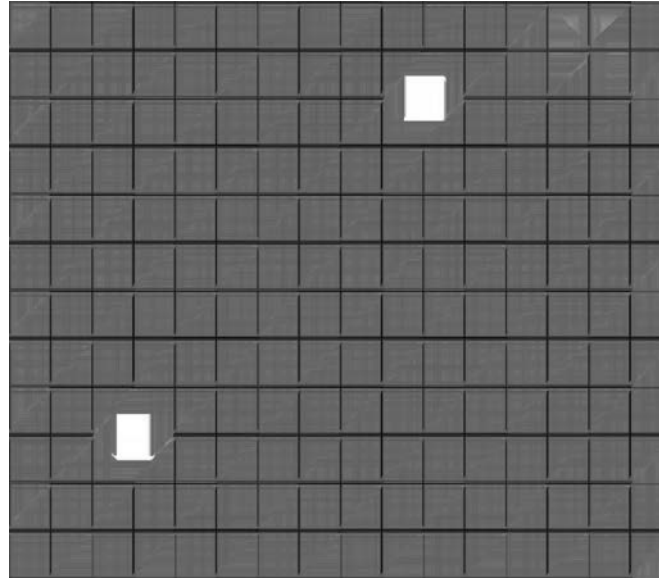
thresholded after dilation



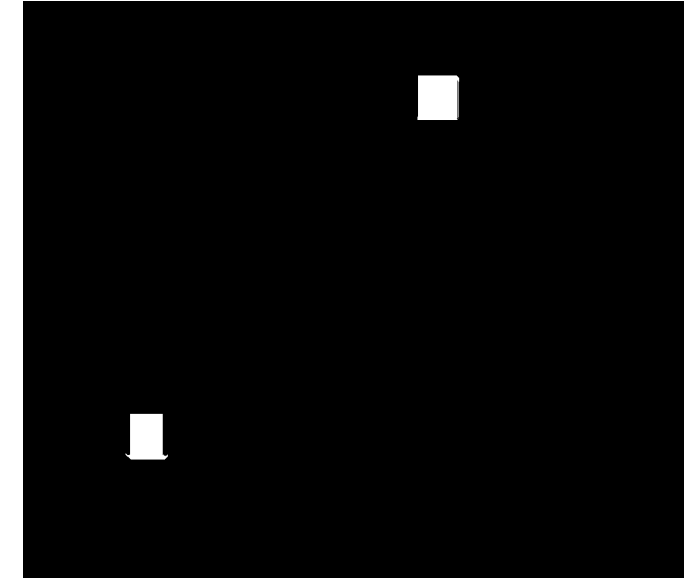
Example: chain link fence hole detection



Original grayscale image
Fence (1023 x 1173)



Flat erosion with 151x151
“cross” structuring element



Binarized by Thresholding



For the fence hole detection example, which method would you expect to give the better results in the presence of noise:

- (a) Flat erosion with 151×151 cross, followed by thresholding
- (b) Thresholding of the image, followed by erosion with a 151×151 cross
- (c) Neither has an advantage over the other

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

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

- Flat erosion contained as a special case

Cascaded dilations



f



$\text{dilate}(f, w_1)$



$\text{dilate}(f, w_1, w_2)$



$\text{dilate}(f, w_1, w_2, w_3)$

$$\text{dilate}[\text{dilate}(f, w_1), w_2] = \text{dilate}(f, w)$$

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



Cascaded erosions

- Cascaded erosions can be lumped into single erosion

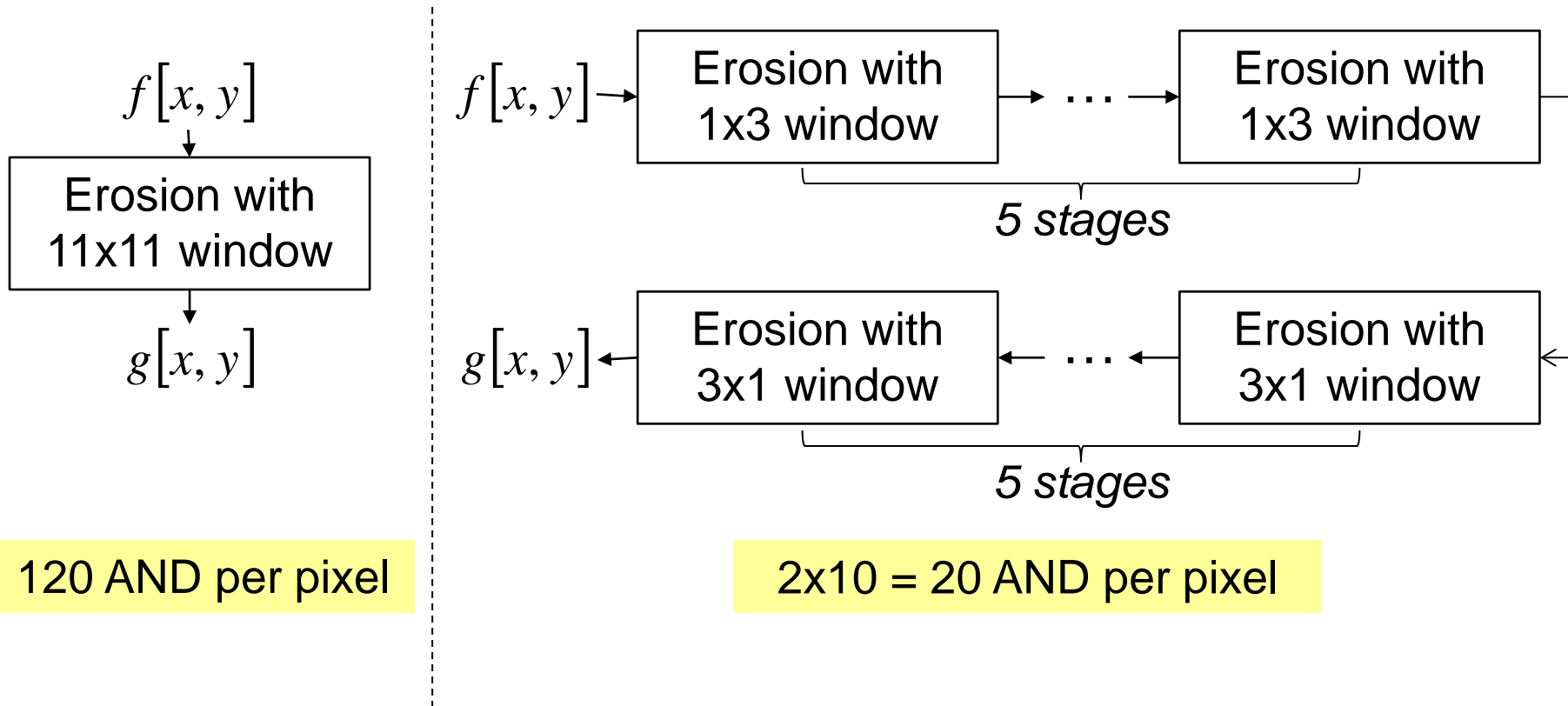
$$\begin{aligned} \text{erode}\left[\text{erode}(f, w_1), w_2\right] &= \text{erode}\left[-\text{dilate}(-f, w_1), w_2\right] \\ &= -\text{dilate}\left[\text{dilate}(-f, w_1), w_2\right] \\ &= -\text{dilate}(-f, w) \\ &= \text{erode}(f, w) \end{aligned}$$

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

- 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



Which of the following schemes performs dilation with a 5x5 square?

- (a) Two cascaded dilations with a 3x3 structuring element
- (b) Two cascaded dilations with a 1x3 structuring element, followed by two cascaded dilations with a 3x1 structuring element
- (c) NOT operation applied to invert the image, followed by two cascaded erosions with a 1x3 structuring element, followed by two cascaded erosions with a 3x1 structuring element, followed by another NOT operation

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

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

- Binary images: majority filter

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

- Self-duality

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

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

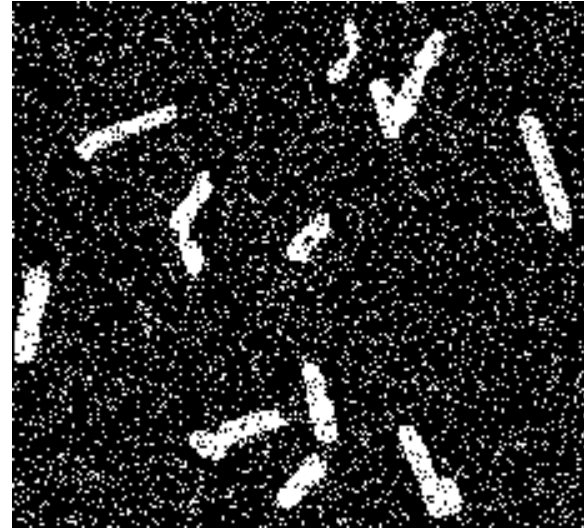
Majority filter: example



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



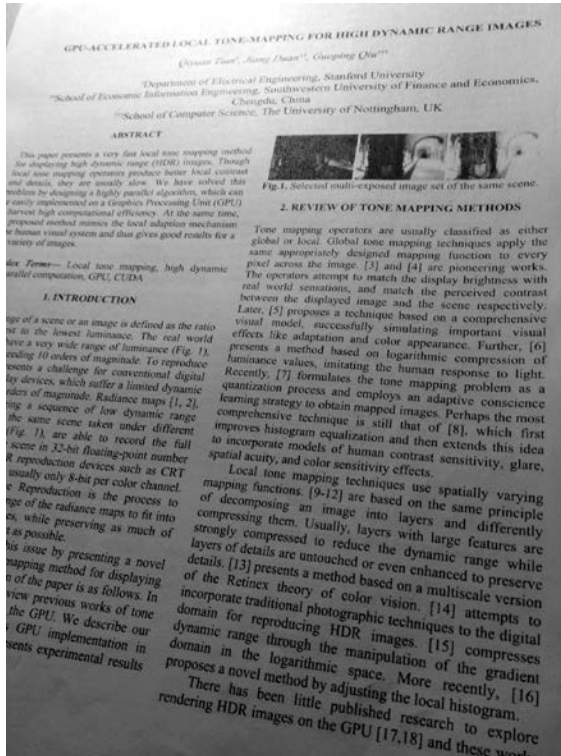
3x3 median filtering



7x7 median filtering



Example: non-uniform lighting compensation



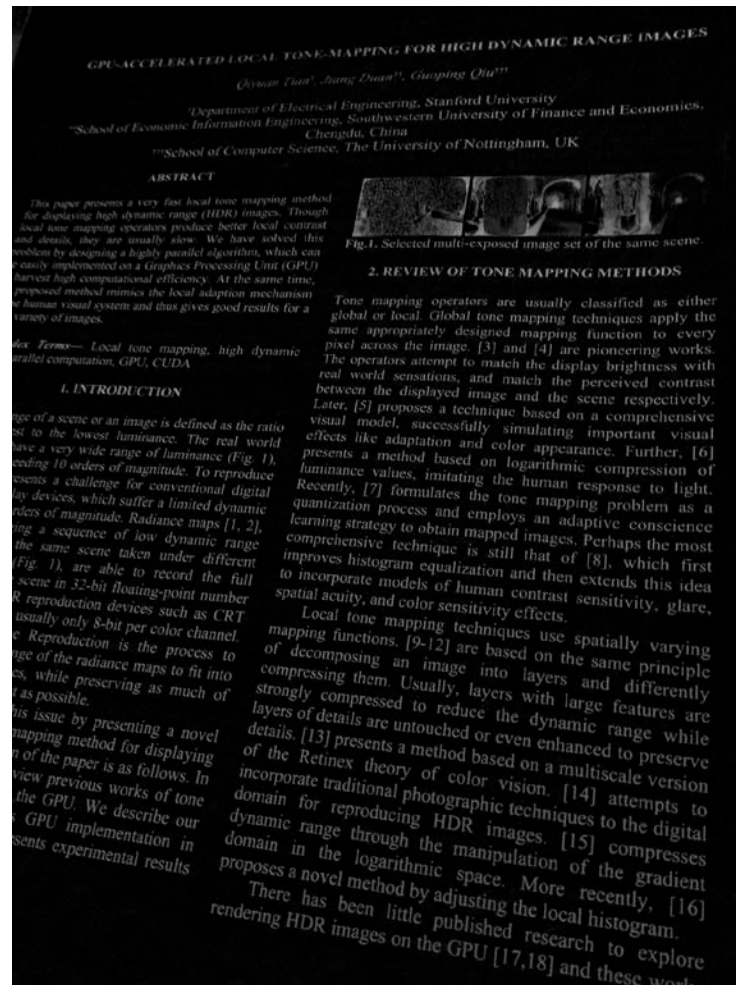
Original image
1632x1216 pixels

Dilation (local max)
61x61 structuring element

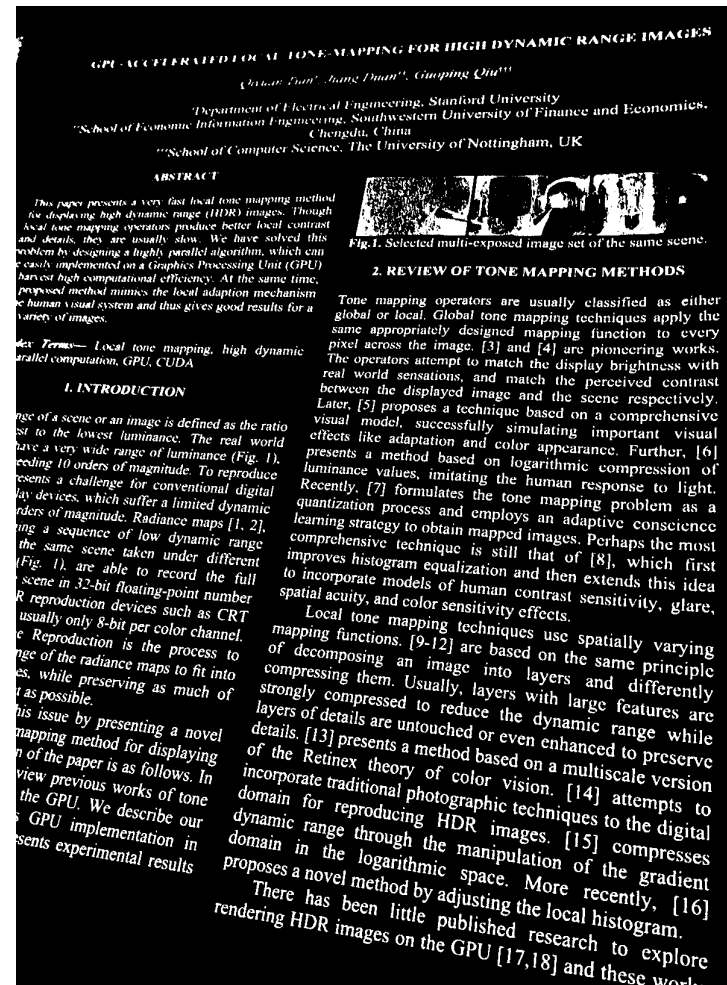
Rank filter
10th brightest pixel
61x61 structuring element



Example: non-uniform lighting compensation



Background – original image



After global thresholding

